Oracle® Fusion Cloud EPM

Administering and Working with Oracle Fusion Cloud Enterprise Data Management

E93969-73

Oracle Fusion Cloud EPM Administering and Working with Oracle Fusion Cloud Enterprise Data Management,

E93969-73

Primary Author: EPM Information Development Team

Contents

4 Working with Views

5 Working with Viewpoints

6 Validating a Viewpoint

7 Comparing Viewpoints

8 Subscribing to Viewpoints

9 About Requests

10 Making Changes Using Requests

16 Working with Reports

17 Working with Dashboards

Part II Administration

18 Getting Started with Administration

19 Working with Applications

21 Defining Alternate Views and Viewpoints

22 Working with Properties

25 Using Expressions to Define Custom Business Logic

26 Understanding Data Sources

27 Working with Matching and Deduplication

28 Working with Roles and Permissions

29 Configuring Policies

30 Integrating Data

33 Working with Financial Consolidation and Close Applications

34 Working with Enterprise Profitability and Cost Management Applications

35 Working with Tax Reporting Applications

36 Working with Oracle Financials Cloud General Ledger Applications

37 Working with E-Business Suite General Ledger Applications

38 Working with Universal Applications

Documentation Accessibility

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

Access to Oracle Support

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

Documentation Feedback

To provide feedback on this documentation, click the feedback button at the bottom of the page in any Oracle Help Center topic. You can also send email to epmdoc_ww@oracle.com.

1

Creating and Running an EPM Center of Excellence

A best practice for EPM is to create a CoE (Center of Excellence).

An **EPM CoE** is a unified effort to ensure adoption and best practices. It drives transformation in business processes related to performance management and the use of technology-enabled solutions.

Cloud adoption can empower your organization to improve business agility and promote innovative solutions. An EPM CoE oversees your cloud initiative, and it can help protect and maintain your investment and promote effective use.

The EPM CoE team:

- Ensures cloud adoption, helping your organization get the most out of your Oracle Fusion Cloud EPM investment
- Serves as a steering committee for best practices
- Leads EPM-related change management initiatives and drives transformation

All customers can benefit from an EPM CoE, including customers who have already implemented EPM.

How Do I Get Started?

Click to get best practices, guidance, and strategies for your own EPM CoE: Introduction to EPM Center of Excellence.

Learn More

- Watch the Cloud Customer Connect webinar: Creating and Running a Center of Excellence (CoE) for Cloud EPM
- Watch the videos: Overview: EPM Center of Excellence and Creating a Center of Excellence.
- See the business benefits and value proposition of an EPM CoE in *Creating and Running an EPM Center of Excellence*.

2

About Cloud EDM

Oracle Fusion Cloud Enterprise Data Management manages changes in your business by ensuring consistency across enterprise data, even across disparate applications.

As your company grows and evolves, you must manage enterprise data across information silos. For example, when you merge with or acquire companies, you must synchronize business entities across various domains. Likewise, new initiatives or legacy applications can make it difficult to compare business operations and consistently measure performance. Yet the success of your business depends on consistent, accurate, and high-quality data.

Cloud EDM provides solutions for managing changes in your business by ensuring consistency across enterprise data, even across disparate applications. Managers can create business-specific views to maintain consistent financial and management data across different lines of business. To simplify and reconcile financial and management reporting, you can define standard, alternate, and custom hierarchies. With Cloud EDM, you can quickly import, manage, and export consistent and accurate enterprise data to multiple external applications.

Videos

Your Goal	Watch This Video
Learn more about Cloud EDM features.	▶ Overview: Tour of Enterprise Data Management Cloud

About this Guide

This section is a description of the guide's content and organization.

This guide is organized into the following sections:

- **About Oracle Fusion Cloud Enterprise Data Management**—Overview information and a high-level process flow for using Cloud EDM.
- **Data Management**—Topics and procedures for users who manage enterprise data. In general, users work with views, viewpoints, requests and update nodes.
- **Administration**—Topics and procedures for administrators who set up users, applications, and enterprise data. This section also provides high-level procedures to achieve specific outcomes for certain business scenarios. Some of the tasks that Administrators perform are:
  - Registering external applications
  - Importing dimensions
  - Creating data chain objects
  - Provisioning users
  - Creating views
  - Exporting to external applications
- **Business Scenarios**—High-level procedures to achieve specific business outcomes.

- **Glossary**—Terms and definitions used in the product and documentation.

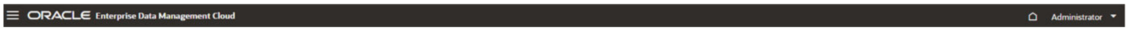
Overview of the Home Page

The Home page is the launch point for accessing your business process tasks.

The interface provides an intuitive user experience and an overview for quick access to commonly used functions. For example, users can view and manage application data using viewpoints and requests. Service Administrators can manage and customize the business process, create user groups and provision roles, migrate artifacts, and make announcements.

Global Header

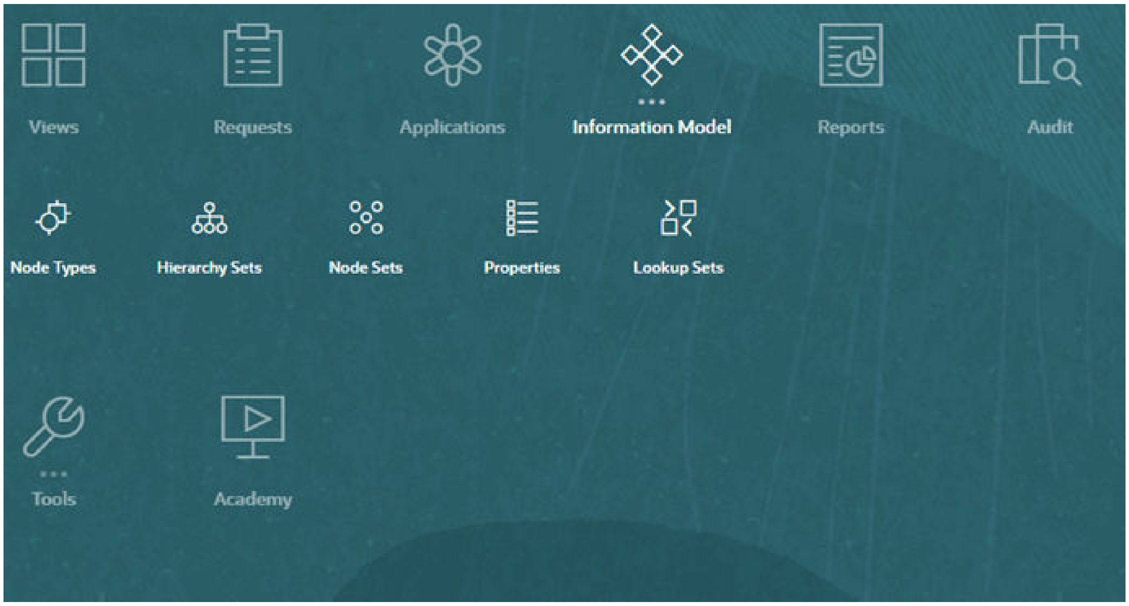
The global header is the area that stretches across the top of the user interface. It contains navigation icons as well as access to the Settings and Actions menu.



Parts of the Global Header from left to right:

Global Header Part	Description
☰	The Navigator icon opens the **Navigator** menu, which serves as a sitemap of the business process and provides links to all of the functions in Oracle Fusion Cloud Enterprise Data Management that you have access to.
ORACLE	Click the Oracle logo to return to the Home page while working elsewhere in the business process. **Note**: Service Administrators can customize the logo. See Customizing Your Users' Display.
Enterprise Data Management Cloud	The name of the current business process. **Note**: Service Administrators can hide the business process name. See Customizing Your Users' Display.
⌂	Click the Home button to return to the Home page while working elsewhere in the business process.
☆	Click the Favorites icon to display your Favorites list, which enables you to navigate to a viewpoint, viewpoint query, request activity filter, or audit transaction filter that you have added as a favorite. See Working with Favorites.
Administrator ▾	Click your user name to access the **Settings and Actions** menu.

Cards and Clusters



Terminology Note

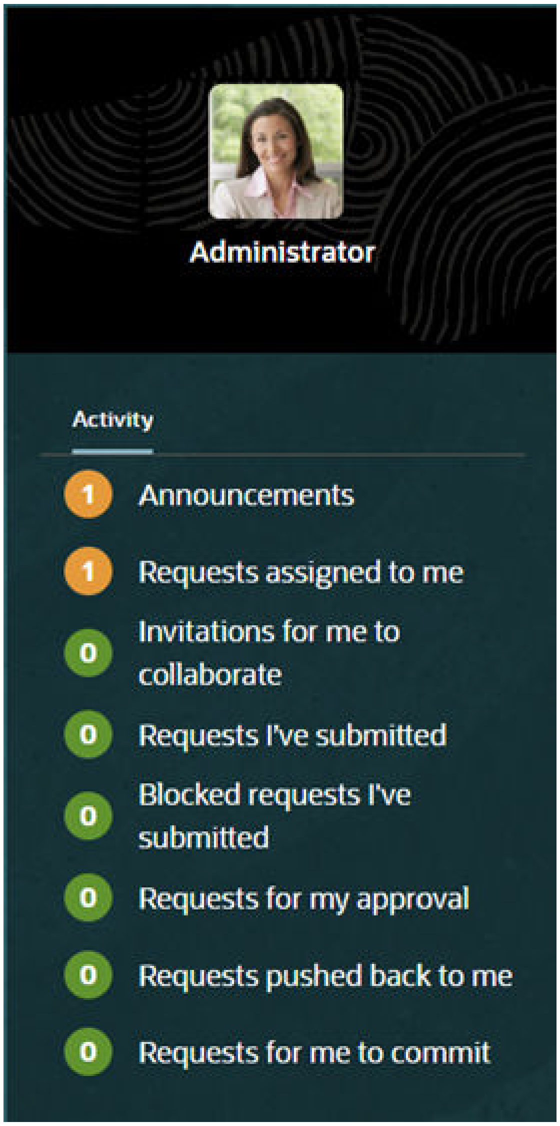
- A *card* is a user interface element that provides access to functionality in Cloud EDM (for example, **Requests** or **Views**).
- A *cluster* is an organizational element that contains a group of cards. Click the cluster to display the cards in that cluster (for example, **Tools**).

The cards and clusters area of the Home page displays icons, called cards, that link you to each functional area of the business process that you have access to. The three dots above a card label indicates a grouping of cards, called a cluster. In the above example, the Information Model cluster is selected, showing the cards that it contains.

> **Note:**
>
> Service Administrators can configure and assign different home page cards and clusters for different users based on their predefined role or user group. See Working with Home Page Layouts.

Welcome Panel

The Welcome Panel displays your user name and profile picture (if set), any system announcements entered by the Service Administrator, and helps you track your activity.



Administrator

Activity

1 Announcements

1 Requests assigned to me

0 Invitations for me to collaborate

0 Requests I've submitted

0 Blocked requests I've submitted

0 Requests for my approval

0 Requests pushed back to me

0 Requests for me to commit

- You can set your profile picture to display at the top of the Welcome panel. Your profile picture also indicates your current Out of Office status. See Setting Your Preferences.
- View the system announcements from your Service Administrator by clicking **Announcements**. The most recent announcement, sorted by start date, appears at the top.

  Click **Close Announcements** to return to the Activity panel.
- Use the Request Activity links to access request activity specific to you. See Accessing Requests from the Home Page.

Working with Favorites

The Favorites List enables you to navigate directly to one of your saved Favorites in these categories:

- Viewpoint (see Navigating Viewpoints)
- Viewpoint Query (see Working with Saved Viewpoint Queries)

- Request Activity Filter (see Working with Request Activity)
- Audit Transaction Filter (see Auditing Transaction History)
- Dashboard Filter (see Working with Dashboards)

To open your Favorites list, from the Home page, click **Favorites** ☆. The Favorites list is maintained for each user; you cannot share Favorites with other users.

From the Favorites list, select a Favorite to navigate directly to it, or click the **Action** ⋮ button next to a Favorite to delete or rename it.

> **Note:**
>
> If an item in your Favorites list is archived or deleted (for example, a saved Viewpoint Query is deleted or a viewpoint is archived), the corresponding Favorite is also deleted from your Favorites list.

Setting Your Preferences

You can set your preferences to upload a profile picture, change your time zone and display language, and mark yourself as out of the office.

To set your preferences:

1. Click your user name in the top- right corner of the screen, and then select **User Preferences**.
2. Click **Edit**.
3. Set your preferences for:
   - Your profile image. Click **Choose File** and then select a file of type `.jpg`, `.png`, or `.gif`. The file that you select must be less than 2 MB.
     Click **Clear** to remove the profile image.
   - Your time zone. Select your time zone from the drop down, or leave the default `Use Client Time Zone` to use the time zone of the browser's locale.

     > **Note:**
     >
     > The time zone that you select is reflected in date and time values throughout Oracle Fusion Cloud Enterprise Data Management, overriding the default time zone for the current locale.

   - Your language. If you don’t select a language, then your browser’s locale is used.

> **Note:**
>
> Log out and in to see the language change. The date format is displayed using the locale format in most places. For example:
>
> – Data objects such as views, applications, node types, hierarchy sets and node sets
> – Properties
> – Audit transaction history
> – Requests

- Your Out of Office status. Follow this procedure to mark yourself out of office:
  a. Select **I am currently out of the office** to take yourself out of the workflow, as follows:
     – Requests are no longer auto assigned to you in subscriptions. All assignments made before you marked yourself as out of the office remain. See Assigning Subscription Requests.
     – If your approval is required to break deadlocks on request escalations, the requests are automatically sent to the Data Manager until you mark yourself as back in the office.
  b. **Optional**: Use the **Delegate** drop down menu to select another user who will be invited to approve or commit requests on your behalf while you are out of the office. See Understanding Request Delegation.

     > **Note:**
     >
     > The person that you assign as your delegate should have the appropriate permissions to view content and approve or commit requests. The system does not automatically assign them the necessary permissions to view and take action on requests.

> **Note:**
>
> When you mark yourself as out of the office, your user profile is updated with an informational indicator to show your current status. Click the indicator to return to **User Preferences** and update your status back to in the office.

4. Click **Save**.

Act as This Role (Service Administrators Only)

Service Administrators can use the **Act As This Role** drop down menu to act as a User in the system. When you act as a User role, only the permissions that are granted to you as a user (and the groups that you belong to) are used to determine your data and metadata access. The permissions granted to you by the Service Administrator predefined role (with the exceptions listed below) are disabled until you update your status back to a Service Administrator.

This enables you to test approval policies and validations with the permissions of a User and not a Service Administrator without having to create test users.

> **Note:**
>
> When you are acting as a User, your user profile is updated with a warning indicator to show your current status as a User. Click the indicator to return to **User Preferences** to update your status back to a Service Administrator.

When you are acting as a User:

- You can see only the views and requests that your user has been assigned permission on.
- You can edit only the metadata that your user has been assigned permission on.
- You can continue to perform these Service Administrator tasks:
  - Register new applications
  - Perform all audit tasks, such as auditing transaction history
  - Make changes on these pages:
    - Access Control
    - Migration
    - Clone Environment
    - Daily Maintenance
    - Templates
- You cannot make changes on these pages:
  - Layouts
  - Connections
  - Data Sources
  - Appearance
  - Announcements
  - Settings

Process Flow

With Oracle Fusion Cloud Enterprise Data Management you can manage business entities for multiple applications in one place using this process flow.

> **Note:**
>
> Service Administrators can perform all tasks. Users perform tasks based on role and permission.

The following table lists the minimum required roles and permissions to perform common tasks. For more information about permissions and data objects, see Working with Permissions.

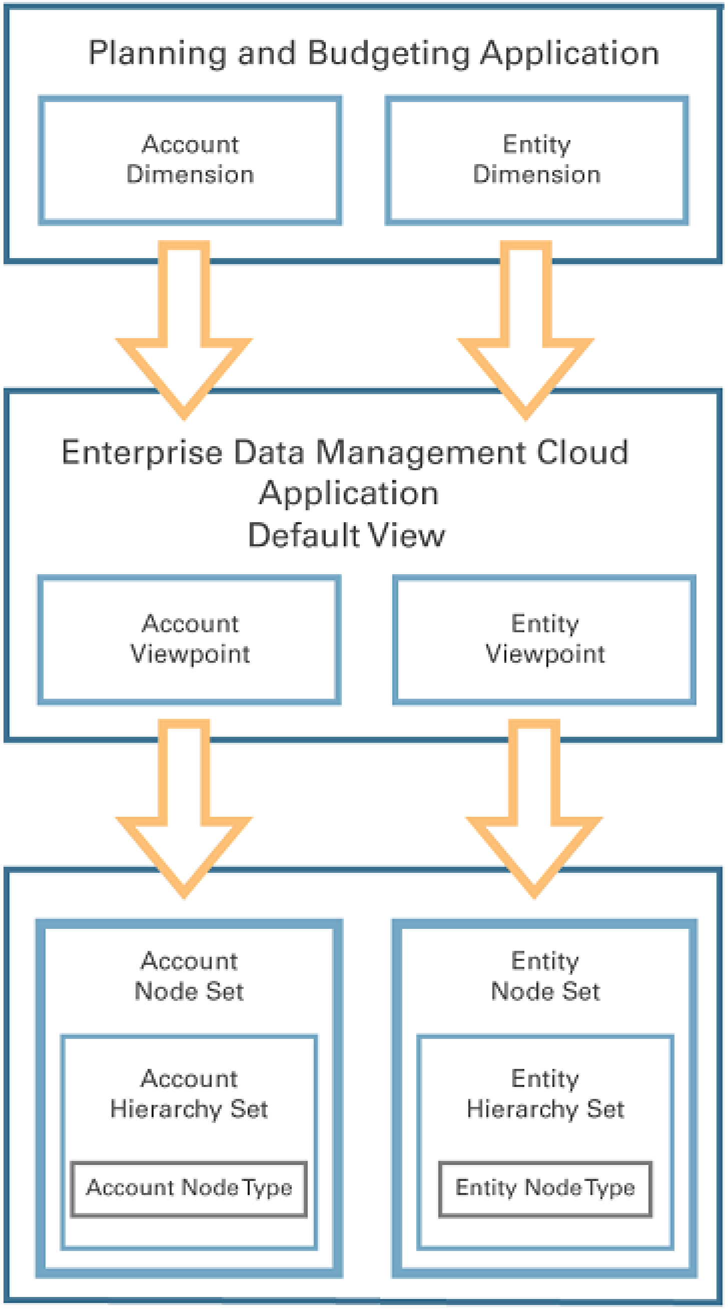
Task	Role	Permission	See
Register external applications (such as Planning) with Cloud EDM	Application - Create		Understanding Registering Applications
Create views	Views - Create		Creating a View
Create viewpoints		*Owner* on the view, and *Data Manager* or *Metadata Manager* on the dimension	Creating a Viewpoint
Browse views and viewpoints		*Participant (Read)* on a data chain object in a viewpoint to view data in that viewpoint	Working with Views Working with Viewpoints
Manage data		*Participant (Write)* on the data chain object to make changes using requests	Understanding Requests Making Changes Using Requests Governance Workflows and Approvals
Compare and rationalize data across applications		• To compare data in viewpoints: *Participant (Read)* on data objects in both viewpoints • To resolve issues by making changes to data: *Participant (Write)* on the data object	Comparing Viewpoints Correcting Node Differences From Compare Results
Validate data		*Participant (Write)* on the data chain object to make changes using requests	Validating a Viewpoint

Task	Role	Permission	See
Synchronize data		To create subscriptions: - *Participant (Read)* permission on the dimension in the source viewpoint - *Data Manager* or *Metadata Manager* permission on the dimension in the target viewpoint - *Owner* on the target view	Subscribing to Viewpoints
Import and export data from external applications		*Data Manager* or *Metadata Manager* on the dimension that you want to import to or export from	Importing DimensionsExporting Dimensions

Register and Import Data

To import data from an external application such as Planning, you'll use a wizard; in it, you select the application's dimensions, properties, and connections to manage in Oracle Fusion Cloud Enterprise Data Management. The registration process creates the components of the Cloud EDM data chain – node types, hierarchy sets, node sets, viewpoints and default view you'll need to work with the application's data in Cloud EDM. The import process populates the data chain with nodes.

This diagram shows the registration and import process specifically for Planning application dimensions Account and Entity.



Planning Account and Entity dimensions are created and imported into Cloud EDM. Each dimension is represented in a viewpoint, node set, hierarchy set, and node type in Cloud EDM. Users manage data in the Account and Entity viewpoints.

Manage Data

You'll access imported data using views in Oracle Fusion Cloud Enterprise Data Management and you'll make changes to the data using requests.

- Understanding Views
- Understanding Requests

Understanding Views

Views give you a collection of lists and/or hierarchies for a particular context or activity, such as entity maintenance. Views consist of one or more viewpoints where you can update data.

Viewpoints provide a subset of nodes for you to work with. For example, viewpoints may represent different cost centers. All changes to enterprise data are performed in viewpoints.

- Working with Views
- Understanding Viewpoints

Understanding Requests

A request is the mechanism to update data. Changes to data are organized into request items. A request item is a group of change actions for a specific node.

Requests enable you to visualize changes before you commit them.

Change actions that can be performed in requests are:

- Add, insert, move, remove, and delete nodes
- Copy and update properties

There are three ways to make changes to data in Oracle Fusion Cloud Enterprise Data Management:

- Interactively (manual changes to viewpoints). For example, in Cloud EDM, you add node Florida as a child node of Southeast-US and you move node Georgia from parent South-US to parent Southeast-US. With each change you make in the user interface, an item is added to the request.
- By subscribing a target viewpoint to a source viewpoint. For example, you subscribe an Entity viewpoint in your Planning application to an Entity viewpoint in your General Ledger application. When a cost center is added in the Entity viewpoint in the General Ledger application, the system generates a request to add that cost center in the Entity viewpoint in your Planning application as well. See Subscribing to Viewpoints.
- Loaded from a file. For example, you are creating a new United States hierarchy and need to add 50 new nodes (one to represent each state). Each change in the load file is a separate item in the request. In this scenario, there would be 50 items in the request, one for each state node being added.

Whether you make changes interactively, through a subscription, or by using a load file, all proposed changes are visualized in the viewpoints where you are making the changes or loading the file. Changes are visible in unique colors and icons so that you can see which parts of the hierarchy or list were changed and what areas may be affected by the change.

While making changes to viewpoints, the system performs validation checks to determine the validity of the changes that are being proposed. If items in the request have validation issues, you can see the reason for the failure and decide how to resolve the issue.

When a request is submitted, validation checks run again, the changes are committed, and the viewpoint is updated with the changes.

Compare and Rationalize Data

With Oracle Fusion Cloud Enterprise Data Management, you can compare and rationalize enterprise data across multiple external applications. You can align dimensions across applications, create and update nodes, and share data changes across applications.

- Aligning Dimensions Across Applications
- Creating New Dimensions
- Sharing Changes Across Applications

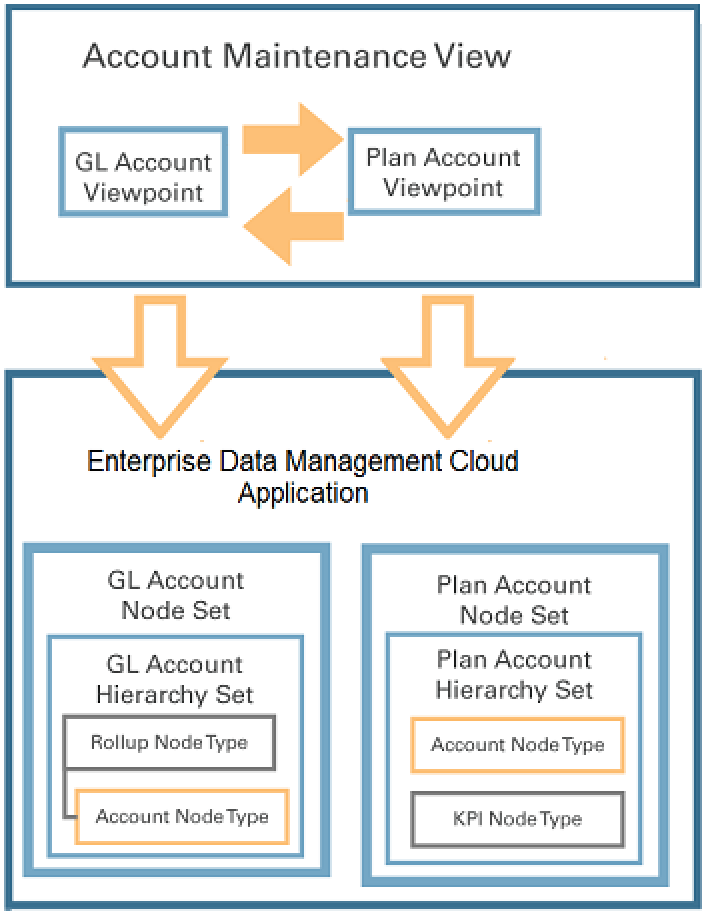
Aligning Dimensions Across Applications

An important benefit that Oracle Fusion Cloud Enterprise Data Management offers is enabling you to compare and align dimensions across multiple applications. If multiple applications that are supposed to have the same dimension information are managed independently of each other, the risk of misalignment between the applications is high.

Misalignment can result from:

- Certain nodes are in one application but not in another
- Nodes with the same name exist in both applications but are not meant for the same purpose. For example the descriptions may be different or the nodes roll up to different parents in the hierarchy.

Cloud EDM enables you to compare nodes in hierarchies or viewpoints and resolve differences to align the two applications.



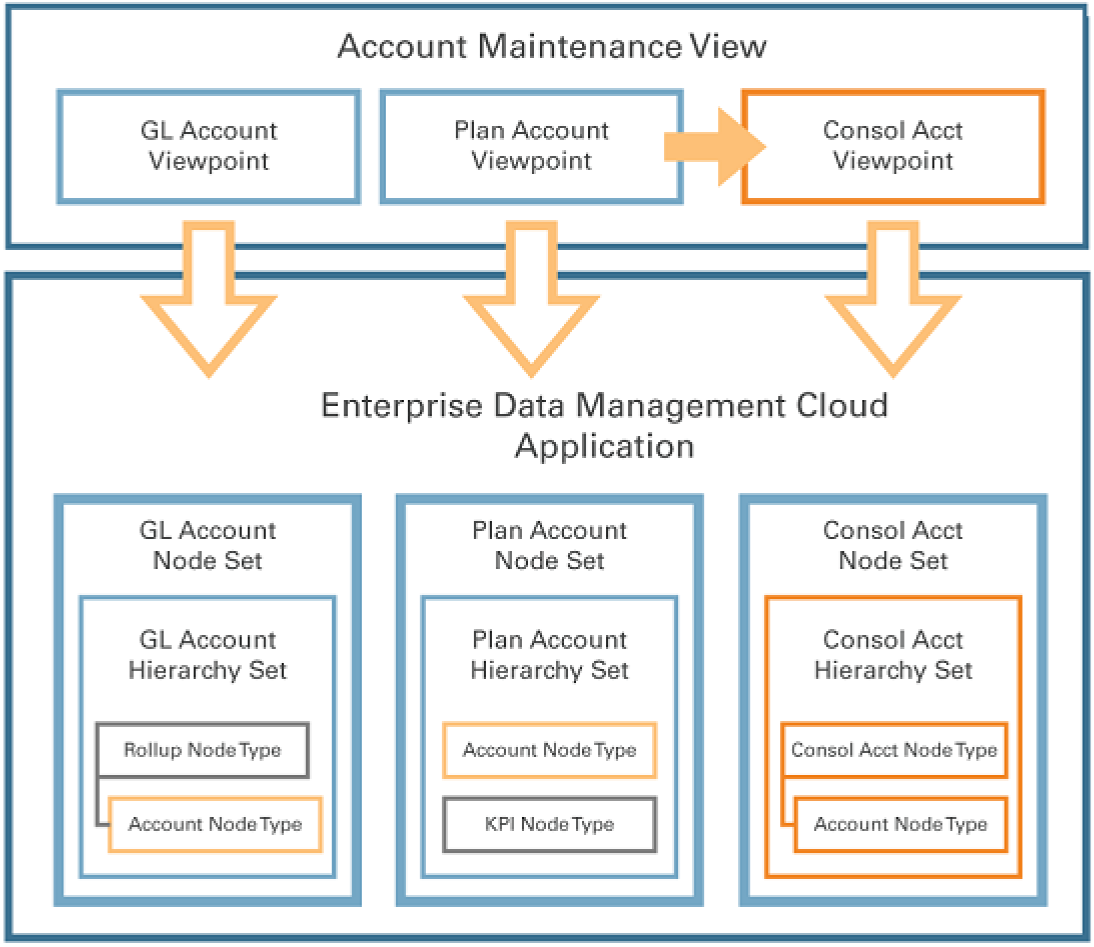
In this example, we are comparing two viewpoints – GL Account and Plan Account – in the Account Maintenance View. We are comparing the nodes in two viewpoints to determine if there are differences that need to be resolved.

Note:

Node type converters are needed to compare nodes of *different node types* across two viewpoints.

Creating New Dimensions

Applications that share transactional data often share dimension information as well. Dimensions for newly created applications can be modeled from existing dimensions from other applications managed by Oracle Fusion Cloud Enterprise Data Management.



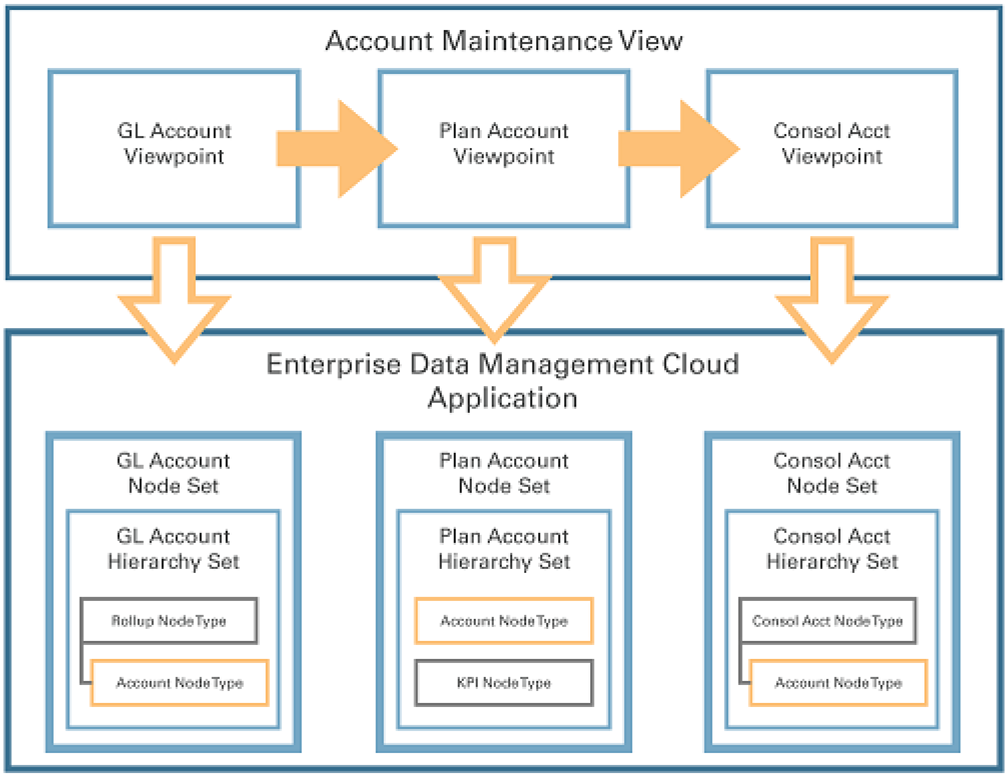
For example, let's say you have a General Ledger and a Planning application and manage the respective Accounts dimensions in Cloud EDM in an Account Maintenance view. Then you purchase Financial Consolidation and Close and need to create an Accounts dimension for your new Financial Consolidation application.

In Cloud EDM, you can use the existing Plan Accounts dimension as a baseline for the new Consolidation Account dimension in the Financial Consolidation application. You create a new dimension in the Account Maintenance view by creating a new viewpoint (Consol Accts). Because this is a new application, you need a new data chain containing the nodes (new node type), relationships (new hierarchy set), and properties for the new Consol Acct dimension.

Sharing Changes Across Applications

You can easily maintain and share dimension updates across applications, including updates to dimension content such as, nodes, properties, and relationships.

Your Goal	Watch This Video
Learn about how to share data across applications.	▶ Sharing Data Across Applications.



This example shows how a new account is synchronized across different viewpoints.

The Account Maintenance view has three viewpoints:

- GL Accounts
- Plan Accounts
- Consol Accounts

Each viewpoint has its own node set:

- GL - Account
- Plan - Account
- Cons - Account

Each node set has a node type (Account) that is common to and shared with all three viewpoints. Three node types (Rollup, KPI, and Consol Acct) are not common or shared.

You can directly make changes to the Rollup or KPI nodes types in the GL Accounts or Plan Accounts viewpoints respectively. Because the Account node type is common, if you add an account, then you must add the account to all three viewpoints so that it's synchronized across the viewpoints. You can add the new account to one viewpoint and then drag and drop it to the other two viewpoints. You can also place the new account in a unique position in the hierarchy of each viewpoint.

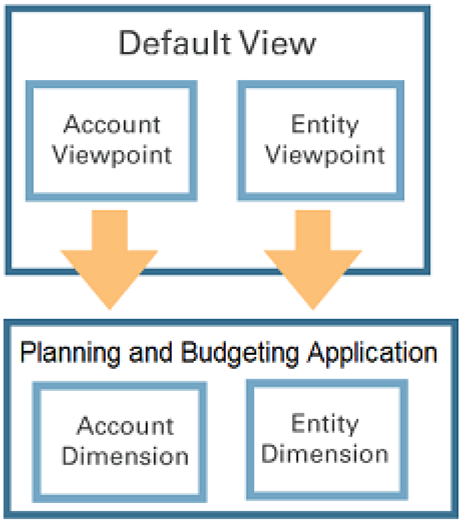
Exporting Data

After making data changes in Oracle Fusion Cloud Enterprise Data Management, you can export and synchronize the updated data to the external target applications.

- Exporting Changes to Applications
- Exporting Dimensions

Exporting Changes to Applications

You use viewpoints to export dimension changes to external applications. Application views mirror how data is represented in external target applications. This enables you to update data in each viewpoint and then load that data back into the respective dimensions of the target application.



Dimension data modified in Oracle Fusion Cloud Enterprise Data Management for Accounts and Entities is exported to the Account and Entity dimensions in the Planning application.

Part I

Data Management

In Oracle Fusion Cloud Enterprise Data Management there are two categories of tasks: data management and administration. This section of the guide describes data management tasks. Data management tasks include creating viewpoints and updating data such as nodes and properties.

See Getting Started with Data Management.

Related Topic: Getting Started with Administration.

3 Getting Started with Data Management

In Oracle Fusion Cloud Enterprise Data Management, you work in views and viewpoints to browse and make changes to enterprise data.

Viewpoints display nodes in a list or a hierarchy layout. You use requests to create and visualize proposed changes to a view. Changes can be made interactively in the view and by using a load file. After you have completed your changes, you submit the request to implement the changes in the view.

You can:

- Add, move, reorder, remove, and delete nodes
- Share nodes by dragging and dropping between viewpoints
- Copy and update properties

You can also compare viewpoints within and across applications to identify and resolve differences. You can validate viewpoints, resolve any issues, and then export updated dimensions to the external applications.

Videos

Your Goal	Watch This Video
Review helpful information on getting started with the user interface.	▶ Getting Started with Enterprise Data Management Cloud for Users

For more information see:

- About Cloud EDM
- Working with Views
- Working with Requests
- User Tasks
- About Nodes
- Understanding Lists and Hierarchies

User Tasks

The following table lists the tasks that can be performed by users who are browsing and managing data. The permission noted is the minimum required to perform the task.

For example, you need the *Participant(Read)* permission on a data chain object to browse that object, but you can also browse that object with *Participant(Write)* permission, because it includes the *Participant(Read)* permission. For more information on permissions, see Working with Permissions.

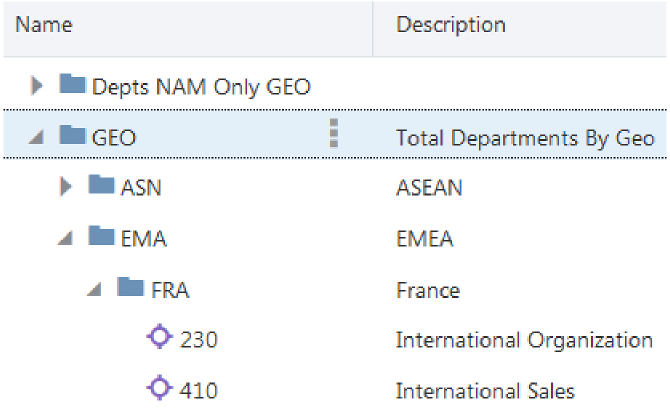
Task	Permission	See
Browse data	*Participant(Read)* on the data chain object	Opening a View Working with Views Understanding Viewpoints Working with Viewpoints
Manage data	*Participant(Write)* on the data chain object to make changes using requests	About Requests Making Changes Using Requests Governance Workflows and Approvals
Compare and rationalize data	• To compare data in viewpoints: *Participant(Read)* on data objects in both viewpoints • To resolve issues by making changes to data: *Participant(Write)* on the data object	Comparing Viewpoints Correcting Node Differences From Compare Results
Validate data	*Participant(Write)* on the data chain object to make changes using requests	Validating a Viewpoint
Synchronize data	To create subscriptions: • *Participant(Read)* permission on the dimension in the source viewpoint • *Data Manager* permission on the dimension in the target viewpoint • *Owner* on the target view	Subscribing to Viewpoints
Import dimensions from external applications	*Data Manager* on the dimension that you want to import to	Importing Dimensions
Export dimensions to external applications	*Data Manager* on the dimension that you want to export from	Exporting Dimensions

About Nodes

Master or reference data records used to describe, qualify, or summarize enterprise data are managed in Oracle Fusion Cloud Enterprise Data Management as nodes. Nodes can be grouped into lists or hierarchies. For example, within a hierarchy that represents an organizational structure, a node might represent a department or a cost center.

Every node is assigned to a node type. The node type defines the properties that are available for nodes. For more information, see Working with Node Types.

The following terms define the position and behavior of a node within a hierarchy. The examples refer to the hierarchy below.



Name	Description
Depts NAM Only GEO	
GEO	Total Departments By Geo
ASN	ASEAN
EMA	EMEA
FRA	France
230	International Organization
410	International Sales

Node	Definition and Example
Parent	The node that contains other nodes. `FRA` is the parent node of 230 and 410.
Child	A node that is contained by another node. 230 and 410 are child nodes of `FRA`.
Sibling	All nodes that have the same parent node in a hierarchy. `CAD` and `USA` are sibling nodes.
Leaf	A bottom-level node with no children. 230 and 410 are leaf nodes.
Limb	A node that has children. `EMA`, `FRA`, and `NAM` are limb nodes.
Top Node	The starting node for a hierarchy. A viewpoint can have multiple top nodes. `Depts NAM Only GEO` and `GEO` are top nodes.
Root Node	The highest node in a hierarchy set for a node type. Root nodes do not have parents. For example, the highest node in a hierarchy set is the root node.

Understanding Lists and Hierarchies

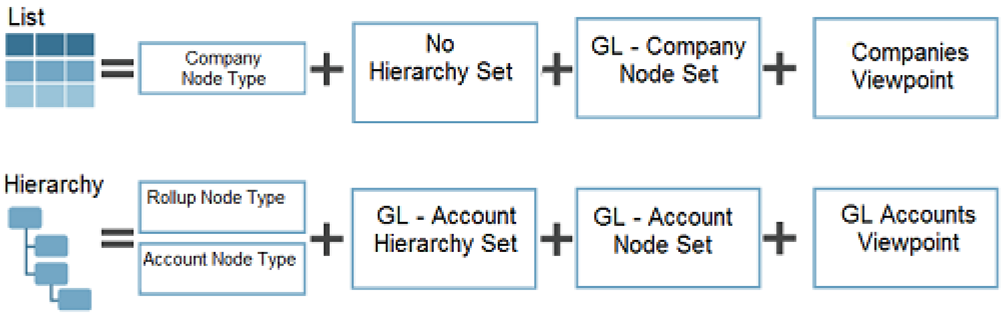
You can configure viewpoints to display nodes in either a list format or a hierarchy structure.

Lists are comprised of:

- One or more node types (for example, Account and Company)
- A node set—A set of nodes representing the members of the list
- A viewpoint—The interface applied on top of a node set to allow users to view and manage the nodes in the list format

Hierarchies are comprised of:

- One or more node types (for example, Account and Company)
- A hierarchy set—Relationships between the node types in the hierarchy set are defined here. For example you can define which nodes types can be children of another node type.
- A node set—A set of nodes representing the members of the hierarchy. The node set can contain a full hierarchy set or one or more branches (top nodes) of a particular hierarchy set.
- A viewpoint—The interface applied on top of a node set to allow users to view and manage nodes in the hierarchy structure



In this example, the Companies viewpoint uses a list format, while the GL Accounts viewpoint uses a hierarchy structure.

For the list, the Company dimension is managed by the Company node type. The set of nodes that represent the Company list are in the GL-Company node set. The GL-Company node set is made available and managed using the Companies viewpoint.

For the hierarchy, the Rollup and Account dimensions are managed by the Rollup and Account node types. The relationships between the Rollup and Account node types are defined in the GL-Account hierarchy set. For example, the Rollup node type can be configured to have child nodes of the Account node type and the Account node type can be set up to allow children. The set of nodes that represent the Rollup and Account node types and the GL-Account hierarchy set combine to create the GL-Account node set. The GL-Account node set is made available and managed using the GL Accounts viewpoint.

Understanding Shared Nodes

Shared nodes are nodes that exist under different parents within a hierarchy set or viewpoint. Shared nodes in different positions are still the same node containing the same node-level properties like Name, Description, and Node Type. Changes made to a node-level property on a shared node affect that shared node in all of its positions.

Videos

Your Goal	Watch This Video
Learn more about shared nodes.	▶ Overview: Shared Nodes in Enterprise Data Management Cloud

Relationship property values are unique to a specific parent-child relationship within a hierarchy set and differ across shared nodes based on their position.

Actions you can take with shared nodes:

- Insert an existing node to make it a shared node.
- Move shared nodes to change the parent.
- Update properties.
- Remove shared nodes. When you remove a shared node, you are removing it from a specific parent. Other relationships for the node in the hierarchy set are unaffected by a remove.
- Delete shared nodes. When you delete a shared node, you are deleting it from all parents in all hierarchies. The node is also deleted from the node type for the application.

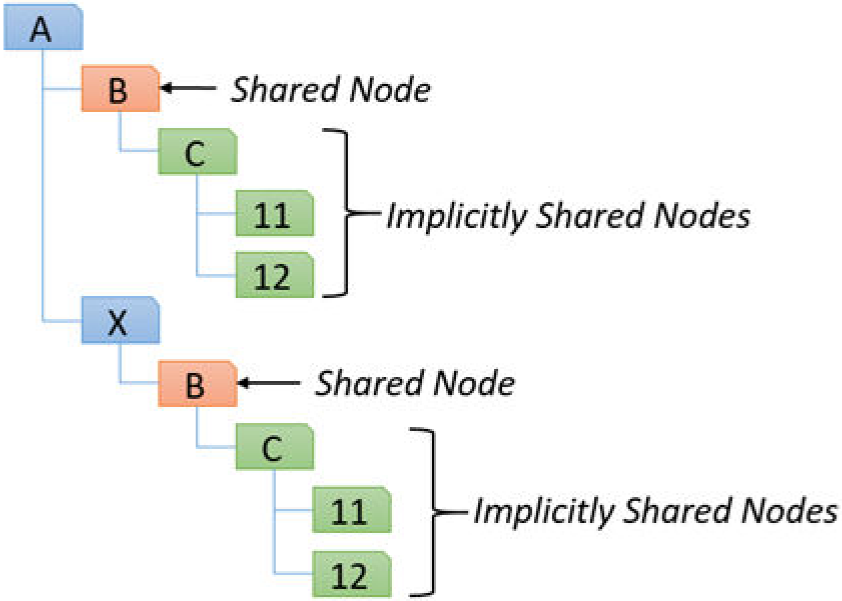
- Search for shared nodes. You can navigate between instances of a shared node when you search for nodes in a viewpoint.
- Enable and disable shared nodes. By default shared node support is enabled.

To configure shared nodes (enable and disable), see Creating Hierarchy Sets.

Implicitly Shared Nodes

Implicitly shared nodes are nodes that have the same parent in multiple locations in a hierarchy.

For example, in the diagram below, node `B` is a shared node because it exists in multiple locations in a hierarchy with different parents (`A` and `X`). Nodes `C`, `11`, and `12` are implicitly shared because they exist in multiple locations in the hierarchy with the same parent (`B`) but with different ancestors (`A` and `X`).



Because implicitly shared nodes also share relationships, if you make a change to the relationship in one part of the hierarchy (such as adding, inserting, moving, or removing a node), that change is made everywhere the implicitly shared nodes exist in the hierarchy. For example, in the diagram above, if you remove the child node `11` from parent `C` under ancestor `A`, that node is also removed from parent `C` under ancestor `X`.

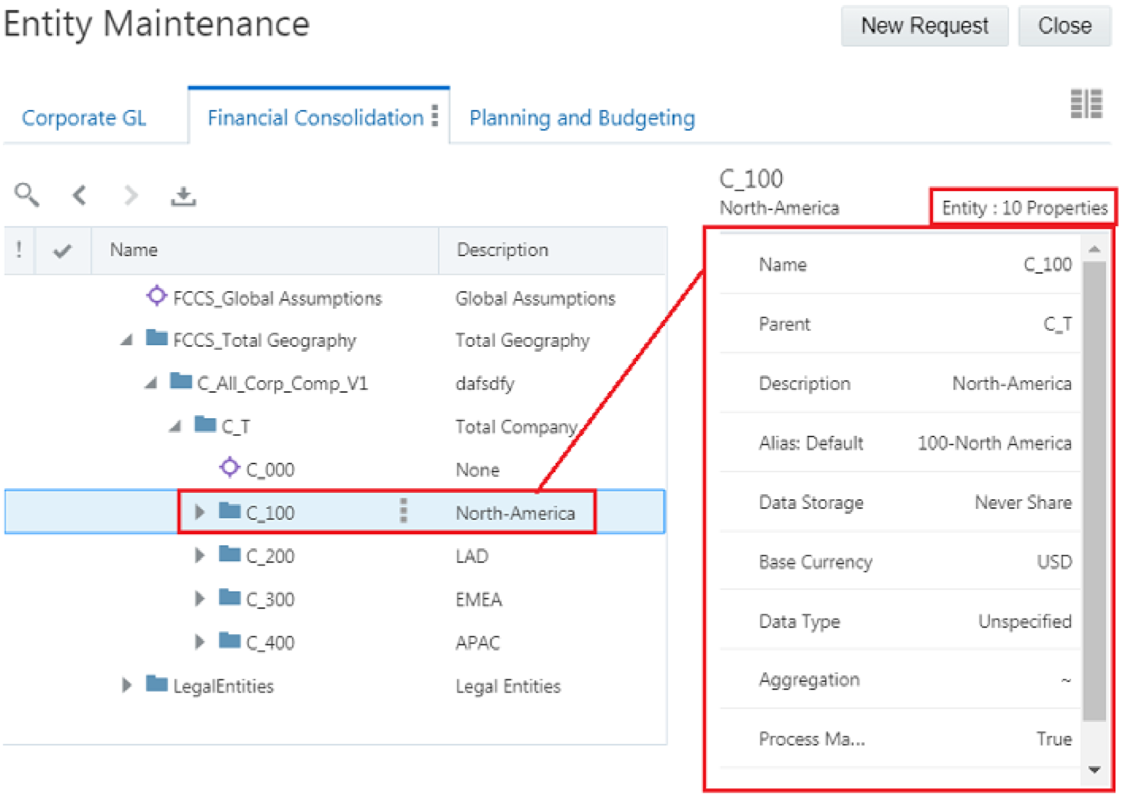
Since implicitly shared nodes have the same parent, defined relationship level property values for implicitly shared nodes are the same across all locations. If you change the value for a defined relationship level property in one location, the value is changed in all other locations as well.

Inherited or derived relationship level property values for implicitly shared nodes may differ across locations. See Property Inheritance and Shared Nodes and Derived Properties.

Working with Node Properties

Properties are characteristics about a node, such as name and description.

When you select a node in a list or hierarchy viewpoint, the properties for the node are displayed in the right side of the viewpoint window.



!	✓	Name	Description
		FCCS_Global Assumptions	Global Assumptions
		FCCS_Total Geography	Total Geography
		C_All_Corp_Comp_V1	dafsdfy
		C_T	Total Company
		C_000	None
		C_100	North-America
		C_200	LAD
		C_300	EMEA
		C_400	APAC
		LegalEntities	Legal Entities

Name	C_100
Parent	C_T
Description	North-America
Alias: Default	100-North America
Data Storage	Never Share
Base Currency	USD
Data Type	Unspecified
Aggregation	~
Process Ma...	True

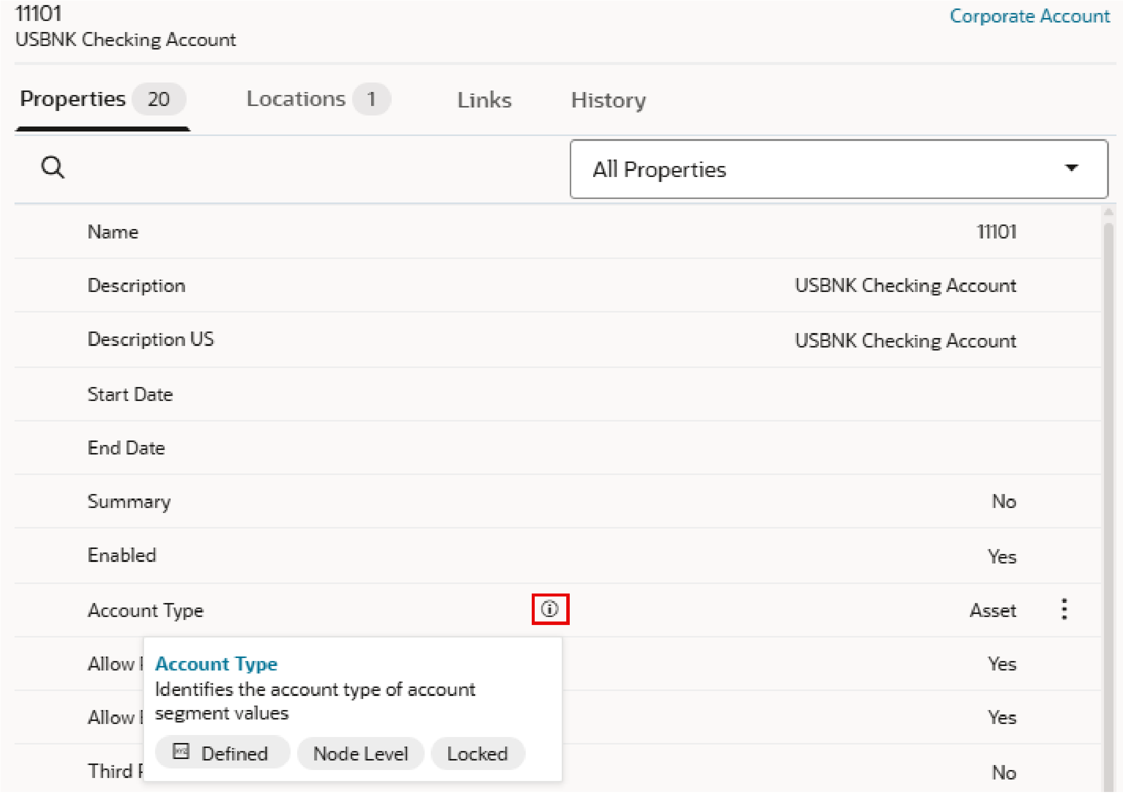
You can use requests to make updates to property values. For more information, see Updating Node Properties.

Property Details

In the viewpoint property pane, when you mouse over a property row an information icon is displayed next to the property label. Click the icon to display information about the property, including the property name, description, property origin, property level (Node or Relationship), whether the property is locked by the Lock on Commit option, and whether or not the property is read only to you (for example, if the property is not editable in the viewpoint).

From the property detail window, you can click the property name to open the property in the inspector. See Inspecting Properties.

You can select and copy text from the information window. For inherited properties, you can click the link to the ancestor providing the inherited property. Click anywhere outside the property detail window to close it. The property information is available whether you are viewing or editing properties.

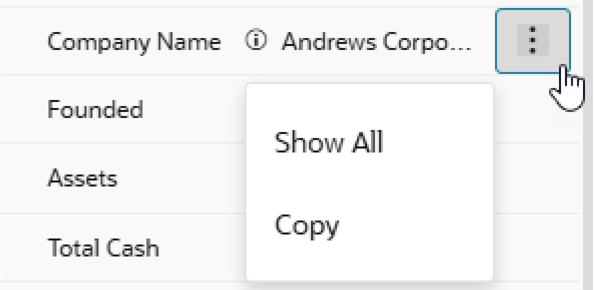


11101
USBNK Checking Account

Name	11101
Description	USBNK Checking Account
Description US	USBNK Checking Account
Start Date	
End Date	
Summary	No
Enabled	Yes
Account Type	Asset
Allow I	Yes
Allow I	Yes
Third F	No

Account Type
Identifies the account type of account segment values

When viewing a property in the property details pane, click **Actions** ⋮ and select **Show All** to display the full property value, or select **Copy** to copy the full property value to your clipboard. You do not need to be in the context of a request to display or copy the property value.



Company Name	Andrews Corpo...
Founded	
Assets	
Total Cash	

Property Groups

View owners create property groups and assign properties to them in order to filter the full property set to display a smaller subset of properties in a viewpoint. (see Configuring How a Viewpoint Displays Properties). When you select a property group from the drop down menu, only the properties in the selected group are displayed. Select `All Properties` to display all of the properties in the viewpoint again.

> **Note:**
>
> The property group drop down menu is not displayed in side by side viewpoint mode. The property groups that you have selected are retained when you display viewpoints side by side. To change the property group, exit side by side mode and select a different property group, and then enter side by side mode again. See Displaying Viewpoints Side By Side.

Property Origin

Property origin identifies how a property value originated for a node. Property values can have these origins:

- Default: Default value set up for the property during application registration or modification.
- Defined: Stored value entered or loaded from a file by user.
- Inherited: Value is inherited from an ancestor.
- Derived: Value is calculated. For example, number of children.
- Derived and Stored: Value is calculated during a request and then stored on the node when the request is completed.

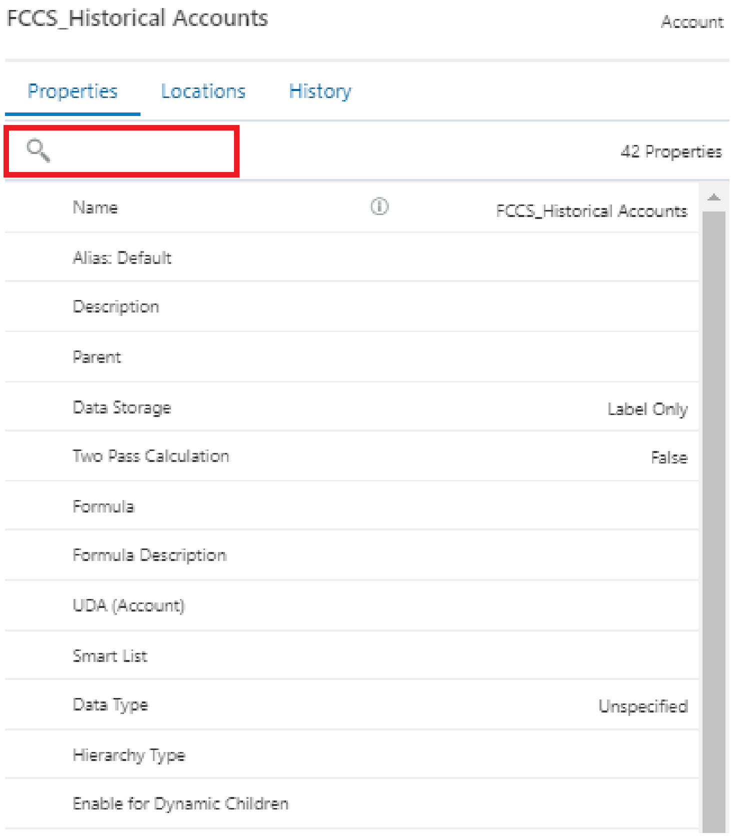
Using Property Inheritance

Property inheritance enables you to define a property value for a node which is automatically inherited by all of the node's descendants. Allowing properties to inherit their values reduces the manual effort of maintaining properties and ensures consistency of values for entire branches of nodes. Inheritance of a property is based on a nodes position in a hierarchy. If a node has an inheriting property, the system looks up the tree of ancestors until it comes to the first ancestor that has a defined value for the property. That defined value is the value that is inherited by the descendant nodes.

Properties used in Universal applications can be configured with inheritance when the applications are registered or modified. Some properties in Planning are enabled with inheritance by default, see Predefined Properties for Planning and FreeForm Applications.

Searching for Properties

In the Properties pane, you can search the properties of a selected node by the property label. This allows you to quickly find a particular property or multiple properties with a similar label. In cases where a node has many properties, being able to search for a property label makes it easy to find the property you're looking for.



FCCS_Historical Accounts

Account

42 Properties

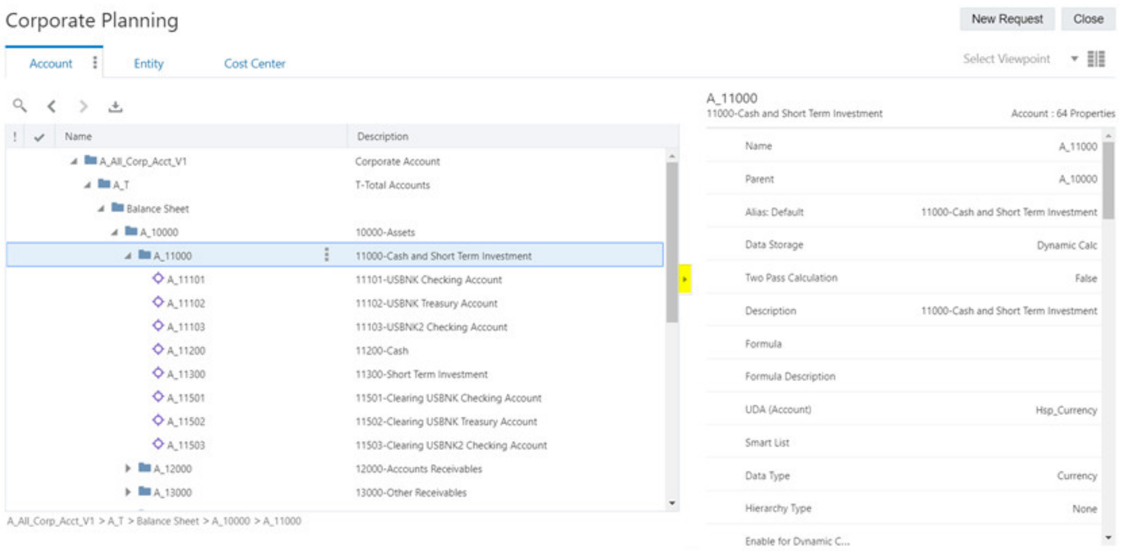
Name	FCCS_Historical Accounts
Alias: Default	
Description	
Parent	
Data Storage	Label Only
Two Pass Calculation	False
Formula	
Formula Description	
UDA (Account)	
Smart List	
Data Type	Unspecified
Hierarchy Type	
Enable for Dynamic Children	

Resizing and Hiding the Properties Pane

You can resize the properties pane within the viewpoint window by hovering over the border of the pane until a slider control is displayed and then dragging the control to the left or right.

You can also hide the properties pane to enable more room on the screen to view the node list or hierarchy.

In tabbed layout, the properties are displayed to the right of the viewpoint. Click the right arrow to hide the properties pane, and click the left arrow to show them.



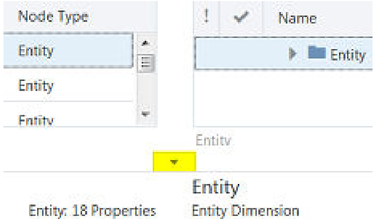
!	✓	Name	Description
		A_All_Corp_Acct_V1	Corporate Account
		A_T	T-Total Accounts
		Balance Sheet	
		A_10000	10000-Assets
		A_11000	11000-Cash and Short Term Investment
		A_11101	11101-USBNK Checking Account
		A_11102	11102-USBNK Treasury Account
		A_11103	11103-USBNK2 Checking Account
		A_11200	11200-Cash
		A_11300	11300-Short Term Investment
		A_11501	11501-Clearing USBNK Checking Account
		A_11502	11502-Clearing USBNK Treasury Account
		A_11503	11503-Clearing USBNK2 Checking Account
		A_12000	12000-Accounts Receivables
		A_13000	13000-Other Receivables

A_All_Corp_Acct_V1 > A_T > Balance Sheet > A_10000 > A_11000

A_11000
11000-Cash and Short Term Investment
Account : 64 Properties

Name	A_11000
Parent	A_10000
Alias: Default	11000-Cash and Short Term Investment
Data Storage	Dynamic Calc
Two Pass Calculation	False
Description	11000-Cash and Short Term Investment
Formula	
Formula Description	
UDA (Account)	Hsp_Currency
Smart List	
Data Type	Currency
Hierarchy Type	None
Enable for Dynamic C...	

In side by side layout, the properties are displayed beneath the viewpoints. Click the down arrow to hide the properties pane, and click the up arrow to show them.



For more information on tabbed and side by side layouts for viewpoints, see Navigating Viewpoints.

Viewing Node Locations

The Locations tab identifies the locations of a selected node in a hierarchy viewpoint. Multiple locations for the same node can be visualized together and you can navigate to any location where a node exists.

Locations are grouped by parent and can be viewed in a collapsed or expanded format. Property values for each location can be displayed to determine similarities or differences across locations.

Your Goal	Watch This Video
Learn about viewing and navigating to multiple locations of a node in a hierarchy.	Location of Nodes in Oracle Fusion Cloud Enterprise Data Management

> **Note:**
>
> The Locations tab displays up to a maximum of 250 locations for a node. If a node exists in more than 250 locations, the total number of locations is displayed.

For shared nodes that exist in multiple locations in a hierarchy viewpoint, the Locations tab provides an easy way to:

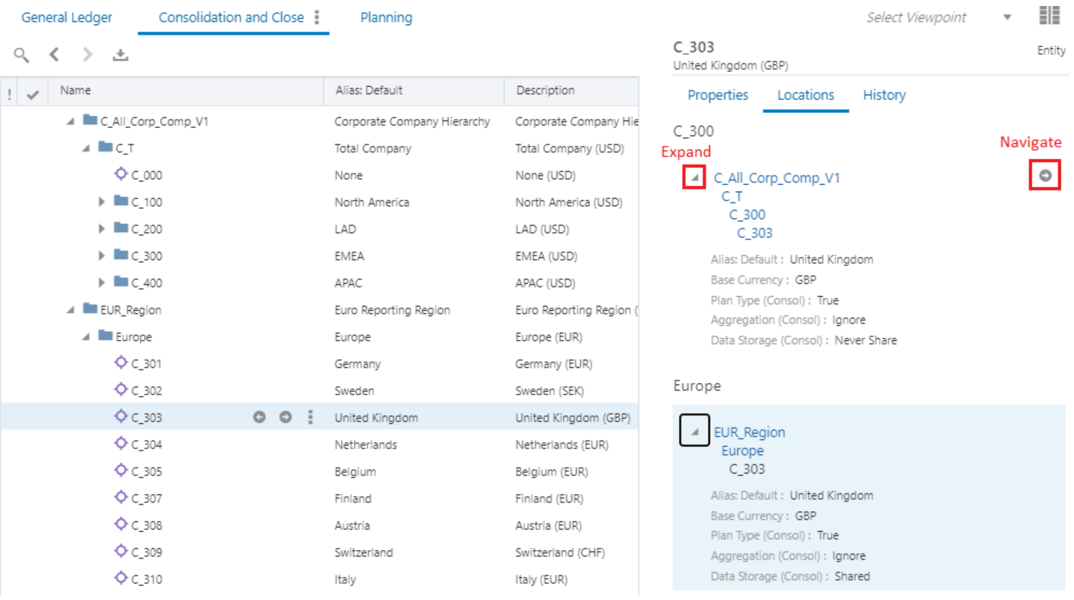
- Identify all of the locations - including the full ancestry of the node
- Navigate to any of the locations to browse or modify data
- Visually compare properties for the node in different locations

> **Note:**
>
> The properties that display in the Locations tab are configured in the viewpoint inspector. For more information, see Configuring Locations Tab Properties.

> **Tip:**
>
> Even if a node doesn't exist in multiple locations, the Locations tab provides a list of ancestors for the node. Expand the ancestry list to navigate to any of the node's ancestors.

To view node locations:

1. Open a view and select a hierarchy viewpoint.
2. Expand a hierarchy and select a node.
3. Click the **Locations** tab.



!	✓	Name	Alias: Default	Description
		C_All_Corp_Comp_V1	Corporate Company Hierarchy	Corporate Company Hie
		C_T	Total Company	Total Company (USD)
		C_000	None	None (USD)
		C_100	North America	North America (USD)
		C_200	LAD	LAD (USD)
		C_300	EMEA	EMEA (USD)
		C_400	APAC	APAC (USD)
		EUR_Region	Euro Reporting Region	Euro Reporting Region (
		Europe	Europe	Europe (EUR)
		C_301	Germany	Germany (EUR)
		C_302	Sweden	Sweden (SEK)
		C_303	United Kingdom	United Kingdom (GBP)
		C_304	Netherlands	Netherlands (EUR)
		C_305	Belgium	Belgium (EUR)
		C_307	Finland	Finland (EUR)
		C_308	Austria	Austria (EUR)
		C_309	Switzerland	Switzerland (CHF)
		C_310	Italy	Italy (EUR)

C_303
United Kingdom (GBP)

C_300

C_All_Corp_Comp_V1
C_T
C_300
C_303

Alias: Default : United Kingdom
Base Currency : GBP
Plan Type (Consol) : True
Aggregation (Consol) : Ignore
Data Storage (Consol) : Never Share

Europe

EUR_Region
Europe
C_303

Alias: Default : United Kingdom
Base Currency : GBP
Plan Type (Consol) : True
Aggregation (Consol) : Ignore
Data Storage (Consol) : Shared

4. Do any of the following to navigate in the Locations tab:
   - Click the expand button to see the node's ancestry.
   - In the expanded view, click any of the node's ancestors to navigate to that node.
   - Click the navigate button to go to a different location of the node in the viewpoint.

> **Note:**
>
> In the collapsed view, you can navigate to the node only. You cannot navigate to a node's ancestor from the collapsed view. In the expanded view, you can navigate to the node or you can navigate directly to an ancestor of the node.

About Node Links

Node links are relationships between nodes across node types. The node link diagram in a viewpoint visualizes these links between source nodes and target nodes.

Understanding Node Links and Data Sources

Node links are established when an existing target node is updated by an incoming source node that has a defined data source. The incoming node can be brought in using several different methods, depending on the type of data source:

- For **registered data sources** (that is, the data is coming from a registered application in Oracle Fusion Cloud Enterprise Data Management), the data can be brought in via:
  - An interactive request when you are inserting a node across viewpoints (see Making Changes Interactively).
  - Nodes that are synchronized from source to target viewpoints using subscriptions (see Subscribing to Viewpoints).
  - Request items are created from a viewpoint comparison (see Comparing Viewpoints).
  - Source nodes are matched and merged to target nodes (see Matching and Merging Request Items).
- For **unregistered data sources** (that is, the data is coming from an external system or file), the data can be brought in via:
  - A request load file (see Making Changes Using a Load File).
  - A viewpoint load (see Working with Viewpoint Loads) or a merge import (see Working with Merge Imports).
  - Source nodes are matched and merged to target nodes (see Matching and Merging Request Items)

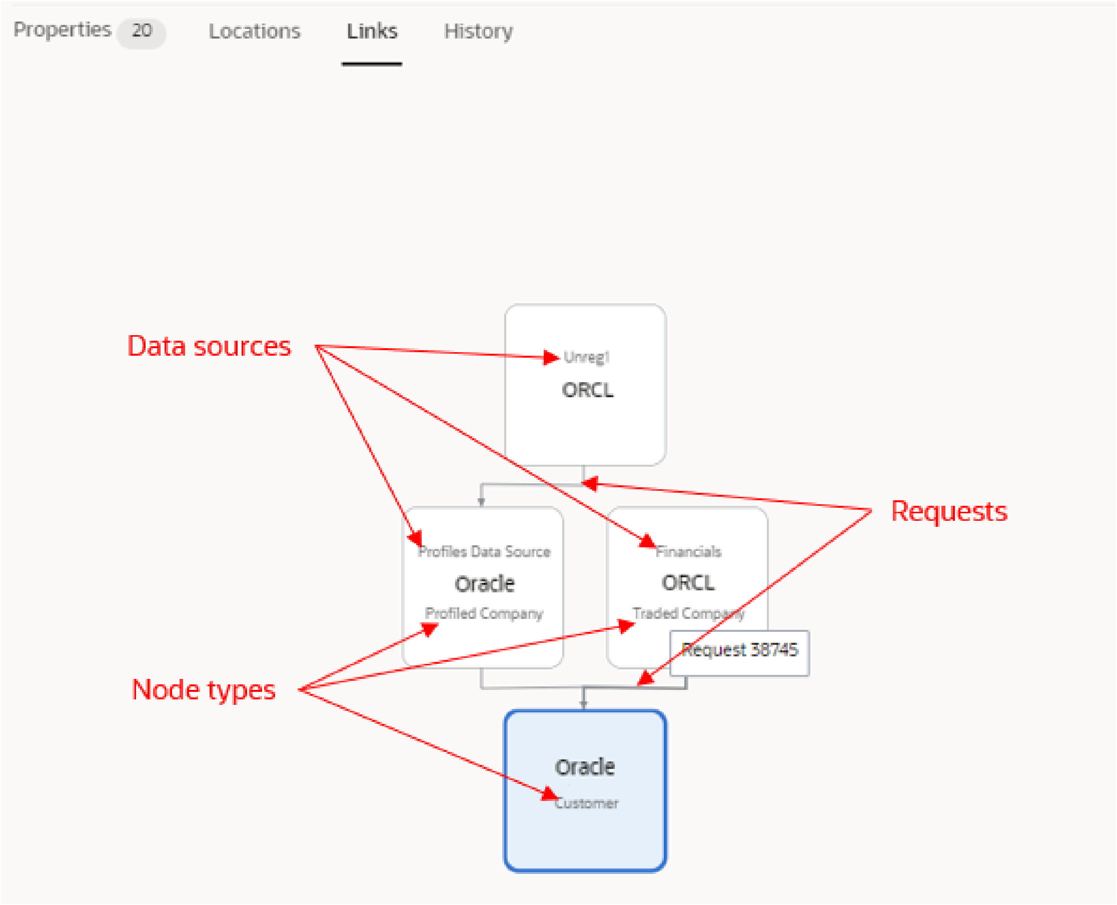
For any of the above methods, if the source node has a defined data source a node link is established between the source and target nodes for that data source. See Understanding Data Sources.

Viewing Node Links

Node links are available as a tab next to Locations in the current viewpoint tab. You need at least *Participant (Read)* permission on a node in order to be able to view its links.

To view links for a node:

1. Open a view and select a viewpoint.
2. Select the node that you want to view links for and then select the **Links** tab.



The link diagram displays the following information:

- The current node is displayed as the selected diagram node.
- The node names are displayed in bold, centered in the diagram boxes.
- The data source names used by the link are displayed above the node name.
- The node type of the nodes are displayed below the node names.
- The requests that link the nodes are displayed as linking arrows. Hover over the linking arrow to display the request name. Click the arrow to open the linking request in the inspector.

Viewing Node Transaction History

You can view the history of transactions for an individual node in a viewpoint. The node history is available as a tab next to properties in the current viewpoint tab. History for a node is available in the active view for the viewpoint and when viewing a draft request for the viewpoint.

To be able to view node history, you need at least *Participant(Read)* permission for the selected node.

Each node history transaction includes:

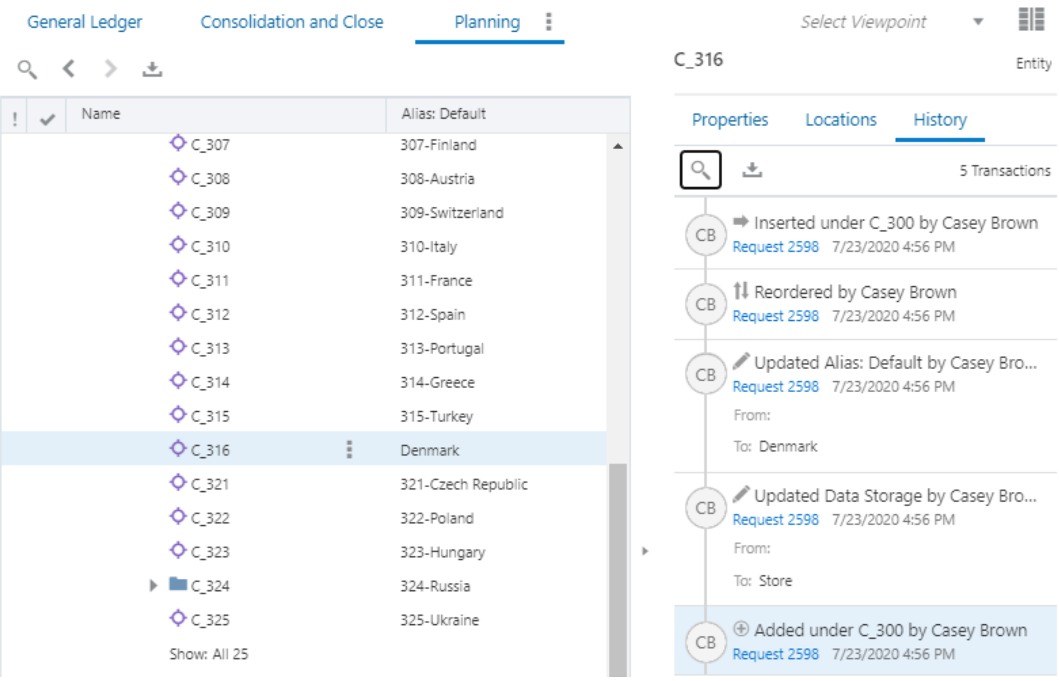
- User who made the change
- Action taken on the node (renamed, updated, moved, etc.)

- Time and date of the change
- Link to the request where the change was made

> **Note:**
>
> Transactions for properties that are hidden for the viewpoint do not display in the node history.

To view a node's history:

1. Open a view and select a viewpoint.
2. Select the node that you want to view the history for and then select the **History** tab.



!	✓	Name	Alias: Default
		C_307	307-Finland
		C_308	308-Austria
		C_309	309-Switzerland
		C_310	310-Italy
		C_311	311-France
		C_312	312-Spain
		C_313	313-Portugal
		C_314	314-Greece
		C_315	315-Turkey
		C_316	Denmark
		C_321	321-Czech Republic
		C_322	322-Poland
		C_323	323-Hungary
		C_324	324-Russia
		C_325	325-Ukraine

Show: All 25

CB Inserted under C_300 by Casey Brown
Request 2598 7/23/2020 4:56 PM

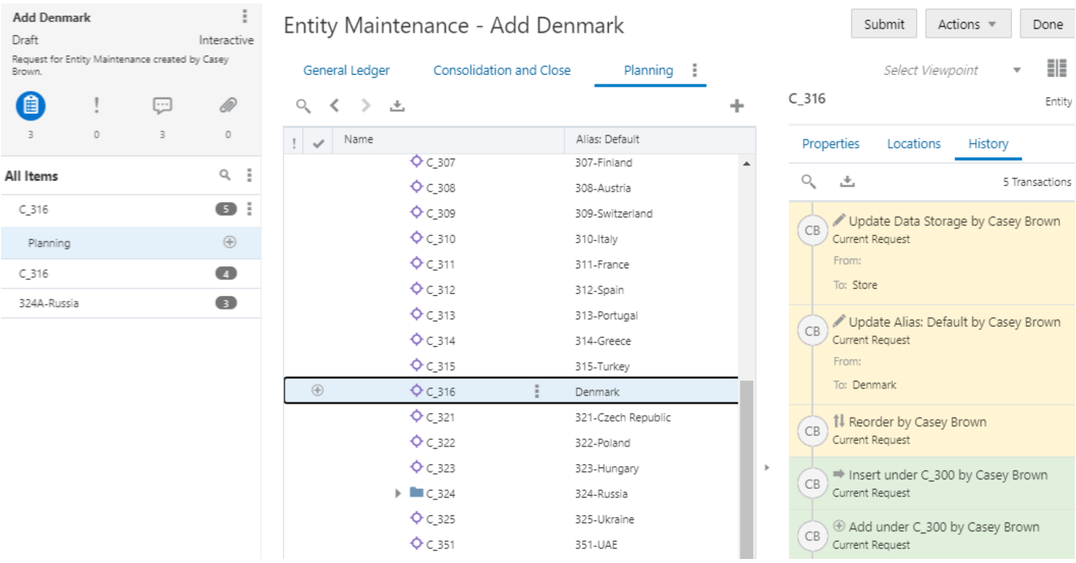
CB Reordered by Casey Brown
Request 2598 7/23/2020 4:56 PM

CB Updated Alias: Default by Casey Bro...
Request 2598 7/23/2020 4:56 PM
From:
To: Denmark

CB Updated Data Storage by Casey Bro...
Request 2598 7/23/2020 4:56 PM
From:
To: Store

CB Added under C_300 by Casey Brown
Request 2598 7/23/2020 4:56 PM

Node history is also available for draft requests.



Name	Alias: Default
C_307	307-Finland
C_308	308-Austria
C_309	309-Switzerland
C_310	310-Italy
C_311	311-France
C_312	312-Spain
C_313	313-Portugal
C_314	314-Greece
C_315	315-Turkey
C_316	Denmark
C_321	321-Czech Republic
C_322	322-Poland
C_323	323-Hungary
C_324	324-Russia
C_325	325-Ukraine
C_351	351-UAE

CB Update Data Storage by Casey Brown
Current Request
From:
To: Store

CB Update Alias: Default by Casey Brown
Current Request
From:
To: Denmark

CB Reorder by Casey Brown
Current Request

CB Insert under C_300 by Casey Brown
Current Request

CB Add under C_300 by Casey Brown
Current Request

3. From the node history list, you can:
   - Click the link to inspect the request in which the change was made.
   - Filter the list by searching for specific text.
     a. Click 🔍 and enter your search string.
     b. You can type a different search string to perform a another search or click ✕ to show all node history.
   - Download the node history list or the filtered list to a file.
     a. Click ⤓.
     b. Change the file name if needed and then click **Save**.

4 Working with Views

Views give you a collection of lists and/or hierarchies for a particular context or activity, such as entity maintenance. Views consist of one or more viewpoints where you can update data.

Videos

Your Goal	Watch This Video
Learn about creating views and viewpoints.	Creating Views and Viewpoints

There are two ways that views can be created:

- When you register an application, a default view is created to help you manage a related set of data chain objects for that application. For example, when you import from a Planning application, the view contains all of the elements that you registered for import.
- You can create views to manage data based on a specific business purpose such as synchronizing a chart of accounts across multiple ledgers.

For more information, see:

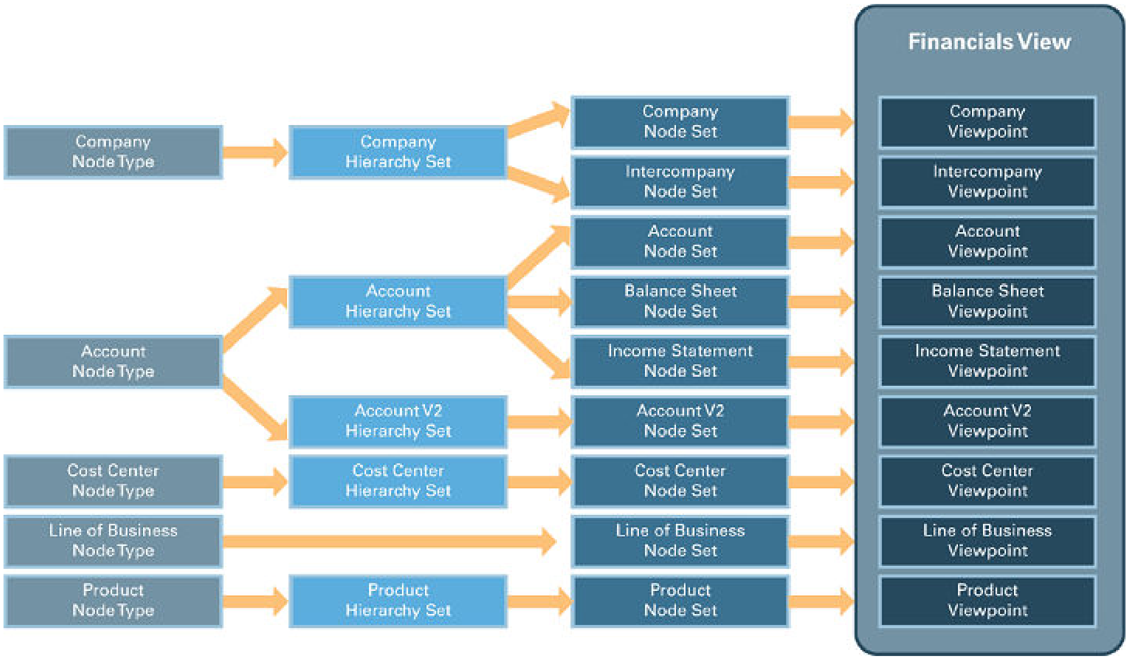
- About Viewpoints
- Creating a View
- Inspecting a View
- Searching for Views
- Opening a View
- Working with Viewpoints
- Inspecting a Viewpoint
- Comparing Viewpoints
- Subscribing to Viewpoints

About Viewpoints

Viewpoints provide a subset of nodes for you to work with. For example, viewpoints may represent different cost centers which require maintenance across applications such as financial applications and Planning and Budgeting Cloud applications.

All changes to data are performed within viewpoints. You can configure viewpoints to display nodes in a list or a hierarchy.

Each viewpoint uses a single node set as its source and may be constructed using one or more node types depending on the business requirements. Viewpoints add business purpose to a node set based on specific properties and validations.



In this example, multiple viewpoints represent a segment value set. Each viewpoint references a node set and a node type. For nodes that need to display in a hierarchy, a hierarchy set is also referenced.

- The Company and InterCompany viewpoints share the same hierarchy set (Company) ensuring that parent-child relationships are global across the Company and Intercompany node sets. This guarantees reporting reconciliation by making sure that a parent has the same descendants across both Company and InterCompany hierarchies.
- The Account, Balance Sheet, and Income Statement hierarchies share the same hierarchy set (Account). Changes you make to the Account hierarchy in one set can be applied across the alternate hierarchies.
- Creating a separate hierarchy set named Account V2 lets you separate the hierarchies for planning purposes and decouple changes you make in production to node sets that map to the Account hierarchy set. Alternate hierarchies allow you to relax the global relationship assumptions in the Account hierarchy set and develop a distinct structure for the next fiscal plan period.
- You can independently maintain the Line of Business value set as a list or value set without associating it with a hierarchy.

Searching for Views

When you perform a search in the Views list, the system searches for your text in view names and descriptions. The names and descriptions that contain the search text are displayed in the results window in bold-faced font. You can also include archived views in your search.

To search in the Views list:

1. From **Views**, click 🔍.
2. (Optional) If you want your search to include archived views, click and then select **Show Archived**.
3. Enter a text string to search for. The search is performed automatically after you stop typing.

To close the search, click ✕ .

Opening a View

You must open a view to be able to work data in viewpoints.

From **Views**, click the name of the view or click ⋮ in the **Actions** column, and then select **Open**.

After you open a view, the left side of the viewpoint window displays tabs to help you perform operations such as querying or validating viewpoints. The following tabs are available:

Icon	Tab	Description
☰	**Requests**	Provides access to open requests in this viewpoint. See Making Changes Using Requests.
○△ △○	**Match Request Items**	Perform matching operations to match and merge incoming request items with existing nodes in a viewpoint. See Matching and Merging Request Items.
⛛	**Query Viewpoint**	Search for nodes in a viewpoint whose properties match a specified criteria. See Querying a Viewpoint.
◫	**Compare**	Compare two viewpoints and locate node differences between them. See Comparing Viewpoints.
□✓ □□	**Validate Viewpoint**	Validate a viewpoint to determine if there are any data integrity issues. See Validating a Viewpoint.
📋	**Node Clipboard**	Construct a list of nodes that you can add or insert into one or more viewpoints. See Working with the Node Clipboard.

Archiving and Deleting Views

You can archive or delete views that you no longer need.

Archiving a View

Before you can archive a view, you must:

- Archive all active viewpoints, see Archiving and Unarchiving Data Chain Objects
- Delete any draft viewpoints, see Deleting a Draft Viewpoint

To archive a view:

1. From **Views**, inspect a view.

2. Select **General** and then click **Edit**.
3. From **Status**, select **Archived** and then click **Save**.

The view is no longer displayed in the **Views** list.

For more information, see Understanding the Lifecycle of Data Objects and the Data Chain.

Displaying Archived Views

You can display both active and archived views in the **Views** list. By default, archived views are not displayed.

From **Views**, click ▼, and then select **Show Archived**. You can select this option again to display active views only.

Restoring an Archived View

You can restore views to Active status that were temporarily archived or archived in error.

To restore an archived view:

1. From **Views**, inspect a view.
2. Select **General** and then click **Edit**.
3. From **Status**, select **Active** and then click **Save**.

Deleting a View

If a view is no longer needed and you have no reason to archive it, you can delete the view. Deleting the view is allowed only if the view contains no viewpoints. For example, you can archive and delete all active viewpoints for a view, and then you can then delete the view.

To delete a view:

1. From **Views**, click ⋮, and then select **Delete**.
2. Click **Yes** to confirm the deletion.

> **Note:**
>
> The view cannot contain any active or archived viewpoints in order to be deleted. You must delete all viewpoints before you can delete a view.

The view is removed from the Views list.

Inspecting a View

From the View inspector, you can view and edit information about the view.

Videos

Your Goal	Watch This Video
Learn about creating views	▶ Creating Views and Viewpoints.

To inspect a view:

1. From **Views**, find a view that you want to inspect.
2. In the **Actions** column for the view, click ⋮, and then select **Inspect**.
3. Select from the options available on these tabs:
    - **General**: Click **Edit** to change these view settings:
        - **Summary**: Change the name, description, or status for the view. See Understanding the Lifecycle of Data Objects and the Data Chain.
            - **Active**—When a view is created, its status is set to Active.
            - **Archived**—When you no longer need a view, you can set its status to archived. See Archiving and Deleting Views.
        - **Display**: Specify which users are able to see this view on the Views page. See Configuring How a View is Displayed in the Views List.
        - **Requests**:
            - For certain request types that are created for the view, you can specify whether the request description is required to be manually entered (for example, to capture a business justification for the request) or automatically generated:
                - **Defaulted**: Request descriptions are automatically generated for all request types but can be optionally overwritten by users.
                - **Required**: Request descriptions are automatically generated only for request types that can be auto-submitted (Subscription, Import, and Load). For Interactive and Consolidation requests, users must enter a description before these request types can be submitted. Use the **Placeholder** field to customize the instructional text that is displayed in the description field on a request (for example, to enter a short description of the information that users should include in the request description).
            - **Attach Request Files for Approvers, Committers, and Notified Users**: Enable this setting to automatically include a file attachment that contains all of the request items in a request to the notification emails that get sent to approvers, committers, and notified users.

                > **Caution:**
                >
                > The file attachment to the email will contain all of the request items in the request, regardless of the recipient's data permissions in Oracle Fusion Cloud Enterprise Data Management.

                > **Note:**
                >
                > This setting is visible only if a Service Administrator has enabled the system setting **Attach Request Files for Approvers, Committers, and Notified Users**. See Configuring System Settings.
        - **History**: Displays information about the view creator and the last person to modify it.

- **Viewpoints**: You can:
  - Create viewpoints, see Creating a Viewpoint
  - Reorder viewpoints, see Reordering Viewpoints
  - Delete a draft viewpoint, see Deleting a Draft Viewpoint
  - Inspect viewpoints, application, dimension, and node set.

  > **Note:**
  >
  > By default, archived viewpoints are not displayed. To display archived viewpoints, click ▼, and then select **Show Archived**. You can select this option again to display active viewpoints only.

- **Time Labels**: Create and edit time labels for the view. See Creating, Editing, and Deleting Time Labels. Users who have created private time labels can delete them from this tab. See Working with Private Time Labels.
- **Compares**: Create and edit compare profiles for the viewpoints in a view, see Creating, Editing, and Deleting Compare Profiles.
- **Permissions**: Set up user and group permissions for the view, see Working with Permissions.

Configuring How a View is Displayed in the Views List

By default, a view is displayed on the Views page to all users who have the ability to browse it. You can control which users can see a view on the Views page by creating a filter for that view in the View inspector.

For example, suppose you had a GL application that contained two different geographical regions, and you've created specific views for each region that have top nodes and labels tailored to that region. You can configure the views so that users see only the views specific to their region in the Views list.

> **Note:**
>
> While creating a filter to include or exclude users from a view determines which views are *visible* to a user on the Views page, it does not affect their actual *access* to a view. A user can still navigate to a viewpoint in a view from the request worklist, or invitations to approve or collaborate. The filter merely determines which views are displayed on the Views page for that user.

To configure how a view is displayed in the views list:

1. Inspect the view that you want to configure. See Inspecting a View.
2. On the General tab, click **Edit**.
3. In **Show on Views Page**, select an option:
   - **Include All Users with Data Access** (default): View is visible to all users who have at least browse access (that is, *Participant (Read)* access to all data chain objects in at least one viewpoint in the view) to the view.

- **Include Only These Users**: View is visible only to users who have browse access to the view that you specify.
- **Exclude These Users**: View is visible to all users who have browse access to the view except the users that you specify.

> **Note:**
>
> You cannot exclude View Owners or Service Administrators.

4. If you selected **Include Only These Users** or **Exclude These Users**, use the drop down list to select the users or groups to include or exclude. The drop down list displays the following users and groups:
   - Users with at least browse access to the view
   - All groups, regardless of permissions or data access
5. Click **Save**.

> **Note:**
>
> You specify users to include **or** to exclude. You cannot specify separate lists of users to include and users to exclude.

Assigning View Permissions

You must have the *Owner* permission on a specific view to be able to assign or remove View permissions for other groups or users.

View permissions are assigned to one view at a time. If you inadvertently remove view permissions for yourself, contact a user with the *Owner* permission or a Service Administrator to re-assign your permissions.

> **Note:**
>
> After you create a view you are given the *Owner* permission to the view.

See Adding, Removing, and Editing Permissions.

Reordering Viewpoints

You can change the order in which viewpoints display when you open a view.

To reorder viewpoints:

1. In **Views**, inspect a view.
2. Navigate to the **Viewpoints** tab, and then click **Edit**.
3. Click the **Action** menu ⋮ next to a viewpoint that you want to reorder, and then select from **First**, **Up**, **Down**, and **Last** to move the viewpoint.
4. Click **Save**.

Creating, Editing, and Deleting Time Labels

Time labels enable you to work with viewpoint data at a point in time in history and to future-date a request.

You can create two classes of time labels:

- **Historical** time labels enable you to create time labeled viewpoints that display data as it existed at a point in history.
- **Future** time labels enable you to set a future date for a request to be completed.

You must have the *Owner* permission on a view in order to create, edit, or delete a time label for it.

Historical Time Labels

Historical time labels enable you to work with viewpoint data at a point in time in history in order to assist in comparing, reporting, and analyzing data.

You can create two types of historical time labels:

- **Fixed** time labels are for a specific date and time in the past. For example, you can create a fixed time label from the initial creation of a Product node set in order to track changes made to it over time.
- **Rolling** time labels represent the end of a month, quarter, or year. For example, you can create a rolling quarterly time label for an Account node set in order to compare current data to last quarter's data.

After you create a historical time label for a view, you can create a time labeled version of any viewpoint in that view in order to work with data from that point in time. See Working with Time Labeled Viewpoints.

> **Note:**
>
> Time labeled viewpoints are read only.

Future Time Labels

Future time labels enable you to select a date in the future on which requests that use the time label will be completed. Requests that have been assigned a future time label will continue to go through all necessary workflow approvals, but they will only be completed and closed when the time in the time label is reached.

You can create future time labels for fixed time periods only. Unlike historical time labels, you cannot create rolling future time labels. However, after the time in a future time label has been reached and there are no longer any active requests assigned to that time label, you can modify the dates and reuse the time label. This lets you create, for example, an "End of Month" time label that you can reset each month after the previous month end has been reached and the associated requests have been closed.

After you create a future time label for a view, you can assign it to requests. When the time specified in the time label is reached, the request is completed and closed. See Working with Future Dated Requests.

Creating Time Labels

You create historical and future time labels at the view level.

To create a time label for a view:

1. Inspect the view that you want to create a time label for. See Inspecting a View.
2. On the **Time Labels** tab, click **Create**.
3. Enter a name, and optionally a description.
4. In **Class**, select Historical or Future.
5. Select a **Type** (historical time labels only):
   - **Fixed**: A specific date and time.
     a. Use the date and time selector to specify a date and time for the time label. You must select a date that is equal or prior to the current date.
     b. **Optional**: Adjust the time zone for the fixed date and time.
   - **Rolling**: A rolling period of time based on the calendar year.
6. For Fixed time periods: Specify a **Date and Time** for the time label.
   - For Future time labels, specify the date and time that requests will be completed and closed. The date must be greater than or equal to the current date and time.
   - For Historical fixed time labels, specify the date and time that you want to view data from. The date must be earlier than the current date and time.
7. For Rolling time periods:
   a. Select the **Period**:

      > **Note:**
      >
      > The Fiscal periods are based on the **Fiscal Year End Month** system setting. See Configuring System Settings.

      - End of Month: the last day of the month
      - End of Quarter: the last day of the quarter in the Gregorian calendar (March 31, June 30, September 30, and December 31)
      - End of Year: the last day of the Gregorian year
      - Fiscal Quarter End: the last day of the Fiscal quarter
      - Fiscal Year End: the last day of the Fiscal year

   b. In **Prior to Current**, select the number of periods prior to the current period to display data for.

> **Tip:**
>
> Examples of **Period** and **Prior to Current** settings:
>
> - End of Month, Prior to Current = 2 for 12/7/22 would be 10/31/22
> - End of Quarter, Prior to Current = 1 for 12/7/22 would be 9/30/22
> - End of Year, Prior to Current = 3 for 12/7/22 would be 12/31/19
> - When Fiscal Year End Month is set to May:
>   - Fiscal Quarter End, Prior to Current = 1 for 12/7/22 would be 11/30/22
>   - Fiscal Year End, Prior to Current = 2 for 12/7/22 would be 5/31/20

c. Enter the **Time of Day** for the time label.

8. Specify the **Time Zone**.
9. Click **Save**.
   The time label is displayed in **Time Labels**.

Editing and Deleting Time Labels

To edit a time label, click **Edit** and adjust any of the parameters above, and then click **Save**. You cannot edit a future time label that is in use by one or more active requests.

To delete a time label, click the **X** next to it, and then click **Yes** to confirm. You cannot delete historical time labels that are in use by viewpoints or future time labels that are in use by one or more active requests.

Creating, Editing, and Deleting Property Groups

Property groups enable you to filter the full property set to display a smaller subset of properties in a viewpoint.

For example, you could create property groups for each cube in a Planning application with properties specific to that cube (for example, Plan Type, Aggregation, Data Storage, and Formula). Then, when working in a Planning viewpoint you can use the property group to filter the displayed list of properties to just the cube you are working with.

> **Note:**
>
> While a property group is created at the view level, the properties that are in a group are specific to each viewpoint in that view. So a property group in the Corporate Planning view might contain different properties in the Account and Entity viewpoints.

Creating Property Groups

You must have *Owner* permission on the view in order to create a property group for that view.

There are two ways to create property groups:

- You can create property groups and assign properties to them directly in the viewpoint inspector. This enables you to create groups and assign properties to them in a single operation.

To create a property group in the viewpoint inspector, see Configuring How a Viewpoint Displays Properties.

- You can create property groups in the view inspector, and then navigate to the viewpoint inspector to assign properties to them. This enables you to see the property groups in a view and the viewpoints that use each group.
  To create a property group in the view inspector:
  1. In the view inspector, navigate to the Property Groups tab, and then click **Create**.
  2. Enter a name for the property group, and then click **Save**.

     > **Note:**
     >
     > The property group name is not case-sensitive, and it has a maximum of 20 characters.

  3. After you create property groups, navigate to the Properties tab in the viewpoint inspector in order to add properties to the group See Configuring How a Viewpoint Displays Properties.

Editing and Deleting Property Groups

You must have *Owner* permission on the view, as well as *Data Manager* or *Metadata Manager* permission on all of the dimensions for all of the viewpoints that use the property group in order to edit or delete a property group.

- To edit the name of a property group, select the group and click **Edit**, and then enter a new name and click **Save**.
- To delete a property group, click **X** next to the property group name, and then click **Yes** to confirm.

Creating, Editing, and Deleting Compare Profiles

For viewpoint comparisons that you run frequently, you can create compare profiles that store the comparison parameters in a reusable profile. These compare profiles are displayed in the viewpoint compare drop down menu.

Compare profiles also enable you to define a node expression in order to filter the nodes that are being compared.

Considerations

- Compare profiles can be used to compare viewpoints from the same view only. You cannot use a compare profile to compare viewpoints in different views.
- You must have the *Owner* permission on a view in order to be able to create a compare profile for that view. Users must have at least *Participant (Read)* permission on both of the viewpoints in a compare profile in order to run it.
- When you select a compare profile in a view, the viewpoints in the profile are displayed automatically and the other compare parameters are disabled. You cannot override the compare parameters from the viewpoint compare screen. Instead, edit the compare profile to modify the compare parameters. See Editing Compare Profiles.

Creating and Deleting Compare Profiles

To create a compare profile:

1. Inspect the view that you want to create the comparison for. See Inspecting a View.
2. On the **Compares** tab, click **Create**.
3. Enter a name, and (optionally) a description for the compare.
4. Select the **From** and **To** viewpoints for the compare from the drop down menus. You can select normal or time labeled viewpoints.

> **Note:**
>
> You can select the same viewpoint as both the From and To viewpoints and then select different top nodes in each in order to compare different hierarchies in that viewpoint. When you run the compare profile, a duplicate tab of the viewpoint is added to the view (if it does not already exist) and is displayed in the side by side layout.
> The duplicate tab remains in the view until you close the view. You can manually remove the duplicate tab or close and reopen the view if you do not want the duplicate tab to persist in the view after you run the compare.

5. Click **Create**.
   The compare profile is displayed in the inspector.

To delete a compare profile:

1. From the **Compare** tab of the View inspector, click **Edit**.
2. In the **Actions** column, click **Actions** ⋮, and then select **Delete**.

Editing Compare Profiles

1. On the **General** tab of the Compare inspector, click **Edit** to change the name or description of the compare profile. You cannot change the From or To viewpoint.
2. On the **Definition** tab of the Compare inspector, click **Edit** to modify the comparison parameters, as follows:
   - In **Compare Scope**, select the nodes that you want to compare (select one):
     - **All Nodes**: Compares all of the nodes in the viewpoint.
     - **Bottom Nodes**: Compares all of the bottom nodes in the viewpoint.
     - **Selected Node and Descendants**: Compares the node that you selected and all of the nodes underneath it. (Hierarchy viewpoints only)
     - **Bottom Nodes for Selected Node**: Compares the bottom nodes underneath the node that you selected. (Hierarchy viewpoints only)

     For both the **Selected Node and Descendants** and **Bottom Nodes for Selected Node** options, you must specify the Selected Node in the **From** and **To** viewpoints.

     Click ⌖ in each field to display the node selector and then drill down and select the nodes to compare.

   - In **Node Filter**, click **Edit Expression** *fx* to open the expression builder and build a node condition expression for the compare profile. If you specify an expression, only nodes in the **From** viewpoint that meet the conditions in the expression are used for the compare. Otherwise, all nodes in the **From** viewpoint are used. See Building Expressions.

- In **Compare Types**, select the type of comparison that you want to run (select one or more):
  - **Find Missing Nodes**
  - **Compare Relationships** (Hierarchy viewpoints only)
  - **Compare Property Values**

  > **Note:**
  >
  > By default, all properties are used for the comparison. Click **Specified** to select specific properties for the comparison. You can only select properties in the **From** viewpoint that are mapped to properties in the **To** viewpoint by a node type converter. See Working with Node Type Converters.

3. Click **Save**.

After you create a compare profile, you can use it to quickly compare two viewpoints using the parameters that you saved. See Comparing Viewpoints.

5
Working with Viewpoints

Viewpoints provide a subset of nodes for you to work with. For example, viewpoints may represent different cost centers which require maintenance across applications such as financial applications and Planning and Budgeting Cloud applications.

Videos

Your Goal	Watch This Video
Learn about creating viewpoints	▶ Creating Views and Viewpoints.

Each viewpoint uses a single node set as its source and may be constructed using one or more node types depending on the business requirements. Viewpoints add business purpose to a node set based on specific properties and validations.

There are two classes of viewpoints:

- Normal viewpoints enable you to access current data
- Time labeled viewpoints are read only viewpoints that enable you to access data from a historical point in time.

For more information on viewpoints, see:

- Understanding Viewpoints
- Navigating Viewpoints
- Validating a Viewpoint
- Inspecting a Viewpoint
- Changing the Order of Displayed Properties
- Comparing Viewpoints
- Copying a Viewpoint
- Downloading a Viewpoint
- Subscribing to Viewpoints

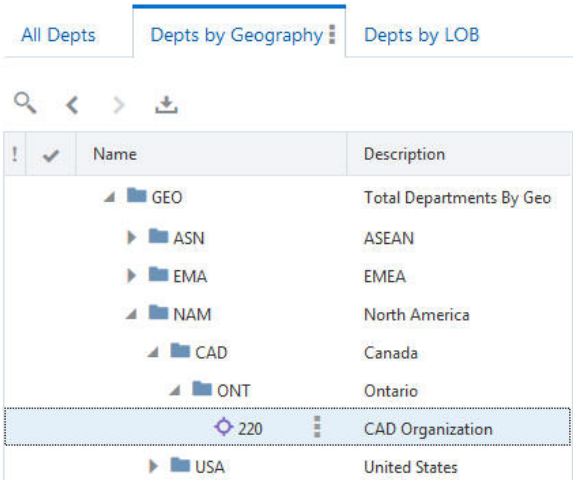
You make changes to data in viewpoints by creating requests. For more information, see About Requests.

Navigating Viewpoints

By default, viewpoints are displayed in a tabbed layout, but you can also display them side by side.

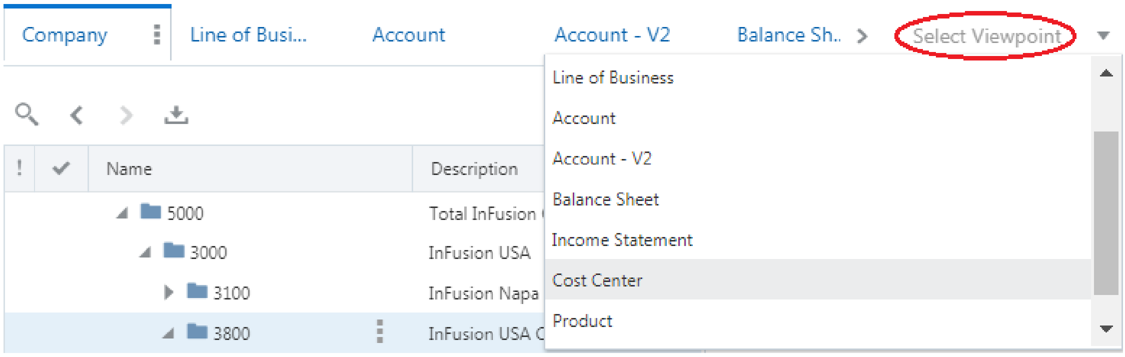
Tabbed Layout

The tabbed layout lets you work with one viewpoint at a time. For example, you need to work with only the Depts by Geography viewpoint.



Name	Description
GEO	Total Departments By Geo
ASN	ASEAN
EMA	EMEA
NAM	North America
CAD	Canada
ONT	Ontario
220	CAD Organization
USA	United States

You can also select a viewpoint to work with from the drop-down list at the far right of the tabbed layout. This list is helpful when you have a lot of viewpoints.



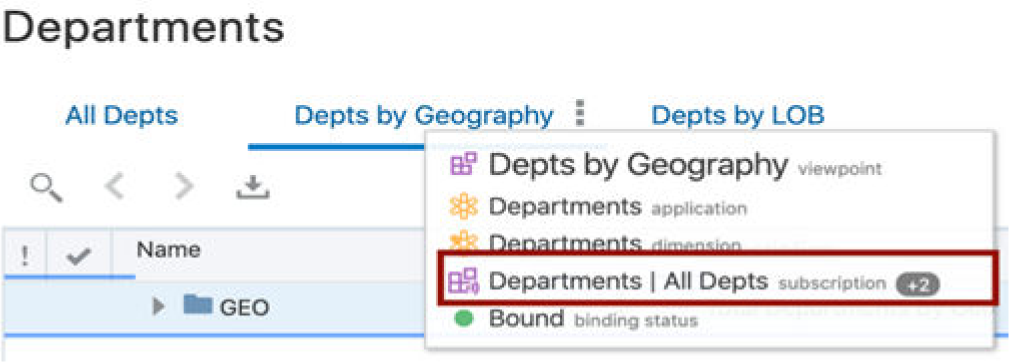
Name	Description
5000	Total InFusion (
3000	InFusion USA
3100	InFusion Napa
3800	InFusion USA C

In tabbed layout, you can hover over the viewpoint name to get more information about the application and dimension. You can also see if the viewpoint is a target of a subscription (see Subscribing to Viewpoints), the viewpoint binding status (see Binding Status), and the start and end dates of the blockout period if the viewpoint is currently blocked out (indicated by a yellow blockout icon) or has an upcoming blockout period (indicated by a green blockout icon). See Requests and Blockout Periods.

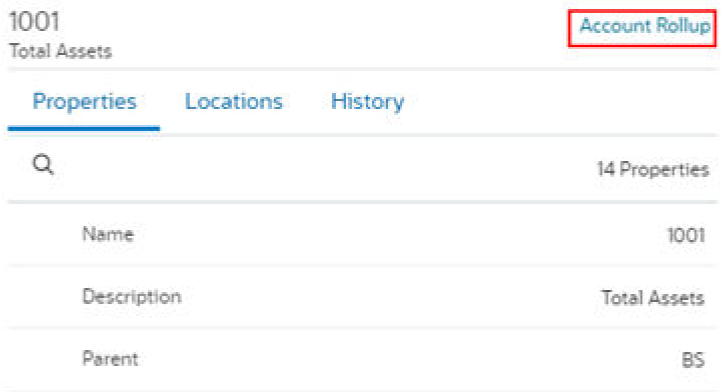
Time labeled viewpoints are indicated by a **Time Label** icon. Viewpoint with a private time label are indicated by a **Private Time Label** icon.

Note:

If the viewpoint is the target of multiple subscriptions, the name of the first one that was created is displayed and the additional number of subscribing viewpoints is indicated with a plus sign (for example, **+2**).



In the tabbed layout, the node type of a selected node in the viewpoint is displayed above the right panel. Click the node type to open it in the inspector. See Inspecting a Node Type.



Name	1001
Description	Total Assets
Parent	BS

Side by Side Layout

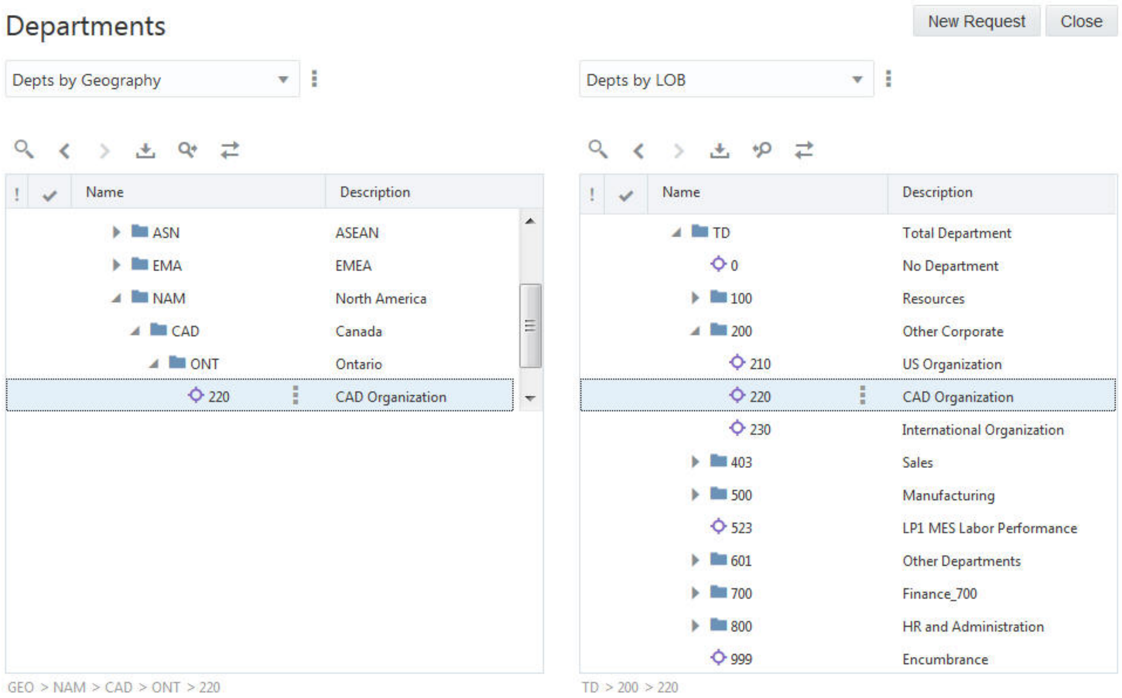
With a side by side layout, you can:

- Visually compare the nodes, properties, and relationships between two viewpoints
- Locate a node from one viewpoint in the other viewpoint
- Align nodes between two viewpoints. As you click on a node in one viewpoint, the node is highlighted in the other viewpoint.
- Perform request actions using action menus.
- Drag and drop nodes across viewpoints. To drag and drop nodes from a viewpoint, you must have at least *Participant(Read)* permission on the data object in the source viewpoint and at least *Participant(Write)* permission on the data object in the target viewpoint, and Add or Insert must be an allowed action.

> **Note:**
>
> Node type converters are needed to compare, locate, align, and drag and drop nodes of *different node types* across two viewpoints in side by side layout, see Working with Node Type Converters.

See Displaying Viewpoints Side By Side.

For example, you can use a side by side layout to locate node 220 in both the Depts by Geography viewpoint and the Depts by LOB viewpoint.



Departments

Depts by Geography

!	✓	Name	Description
		ASN	ASEAN
		EMA	EMEA
		NAM	North America
		CAD	Canada
		ONT	Ontario
		220	CAD Organization

GEO > NAM > CAD > ONT > 220

Depts by LOB

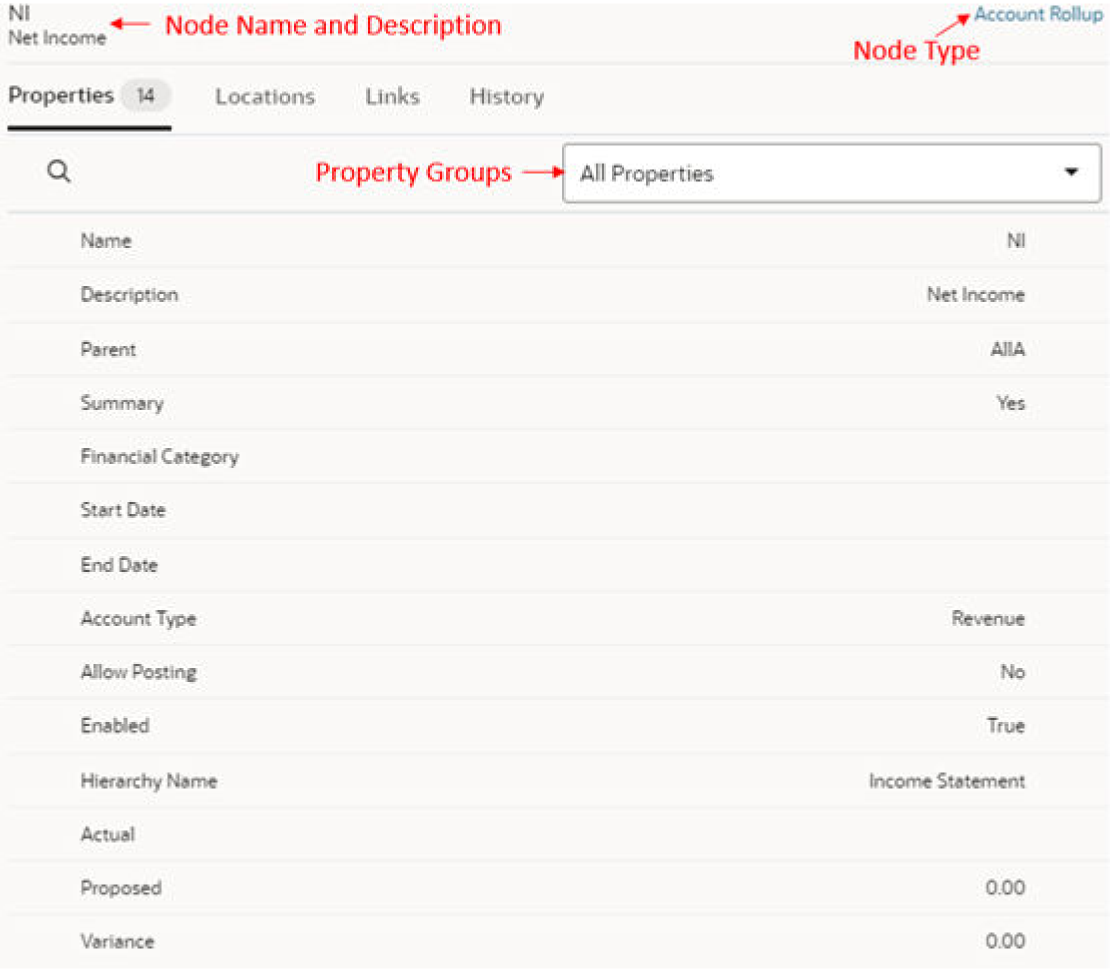
!	✓	Name	Description
		TD	Total Department
		0	No Department
		100	Resources
		200	Other Corporate
		210	US Organization
		220	CAD Organization
		230	International Organization
		403	Sales
		500	Manufacturing
		523	LP1 MES Labor Performance
		601	Other Departments
		700	Finance_700
		800	HR and Administration
		999	Encumbrance

TD > 200 > 220

You can navigate viewpoints in several ways:

- Search allows you to search for a node name by typing in a partial string, see Searching for Nodes.
- Use Back and Forward arrows to switch between the previously-selected node and the current node.
- Use Pager to page through nodes in list viewpoints.

Node Details Pane



Name	NI
Description	Net Income
Parent	AllA
Summary	Yes
Financial Category	
Start Date	
End Date	
Account Type	Revenue
Allow Posting	No
Enabled	True
Hierarchy Name	Income Statement
Actual	
Proposed	0.00
Variance	0.00

The node details pane displays information for a node in a viewpoint. View owners can configure the viewpoint to display specific properties (see Configuring How a Viewpoint Displays Properties) and change the order of the properties in the view (see Changing the Order of Displayed Properties). Users can also use the Property Groups drop down menu to filter the list of properties that are displayed in the viewpoint (see Creating, Editing, and Deleting Property Groups).

The node details pane is displayed to the right of the main tab in the tabbed layout, and on the bottom of each viewpoint in the side by side layout.

Use the tabs in the node details pane to access the following information:

- **Properties**: Displays the properties for a selected node. See Working with Node Properties
- **Locations**: Identifies all of the locations of a selected node in a hierarchy viewpoint grouped by parent. See Viewing Node Locations.
- **Links**: For matched nodes, displays link information about a selected node including the source of the node and downstream linked nodes. See About Node Links.
- **History**: Displays the history of transactions on a selected node. See Viewing Node Transaction History.

Viewpoint Menu

In both the tabbed and side by side layouts, click **Viewpoint Menu** ⋮ next to a viewpoint name to access these operations:

- **Inspect**: Opens the viewpoint inspector, see Inspecting a Viewpoint.
- **Query**: Opens the viewpoint query panel, see Querying a Viewpoint.
- **Compare**: Opens the Compare pane so that you can determine the node differences between two viewpoints, see Comparing Viewpoints
- **Validate**: Validates the nodes in the viewpoint, see Validating a Viewpoint.
- **Add Favorite**: Adds the viewpoint to your Favorites list. Viewpoints in your Favorites list are indicated by a ★ icon. You can remove a viewpoint from your Favorites list by selecting **Undo Favorite**. See Working with Favorites.
- **Copy**: Makes a copy of the viewpoint, see Copying a Viewpoint.
- **Download to file**: Downloads the viewpoint data to an .XLSX file, see Downloading a Viewpoint.
- **Load**: Opens the viewpoint load screen with the current viewpoint selected to be loaded into, see Working with Viewpoint Loads.

> **Note:**
>
> If the viewpoint is currently blocked out due to application blockout settings, the **Load** option is not available.

- **Open in new tab**: Opens the viewpoint in a duplicate tab. See Using Duplicate Viewpoint Tabs.

Viewpoint Toolbar

When working in a viewpoint, the viewpoint toolbar provides access to the following functions:

Icon	Function
Search 🔍	Searches for a node in the viewpoint. See Searching for Nodes.
Back and **Forward** ‹ ›	Navigate backwards and forwards to previously selected nodes.
Download to File ⤓ (hierarchy viewpoints only)	Downloads the selected node and its descendants to a file. See Downloading a Viewpoint.
Expand/Collapse All Below ▾ (hierarchy viewpoints only)	Click and select **Expand All Below** or **Collapse All Below** to expand or collapse all levels of descendants below the selected node.
Locate Node (side by side layout only)	In side by side layout, locates the selected node from one viewpoint in the other viewpoint. See Locating Nodes.
Align Node ⇄ (side by side layout only)	In side by side layout, aligns nodes and properties. See Aligning Nodes.

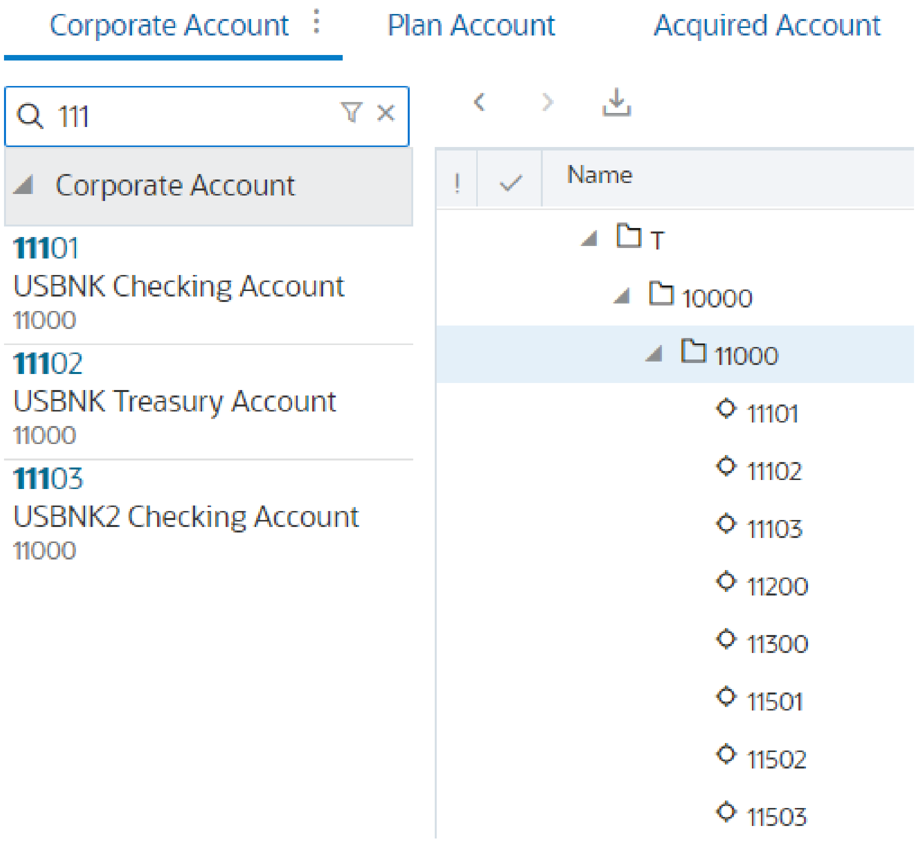
Searching for Nodes

You can search for nodes by name and description in a viewpoint and in the node selector. Enter text to search for in the Search box and nodes that have a name or description that contains a word that starts with the string that you entered are displayed. For hierarchy viewpoints only, you can search for a node below a specific node in the hierarchy.

If you enter multiple search strings, the search results will display nodes whose name or description contains a word that begins with any of the search strings. For example, if you search for `store 002`, the search results will display nodes whose name or description contain a word that begins with `store` and nodes whose name or description contain a word that begins with `002`.

You can search for nodes in node types that have defined a default or alternate qualifier (see Working with Node Type Qualifiers). For example, if the node type has a default qualifier of "A_" and an alternate qualifier of "C_", then searching for 100 will return 100, A_100, and C_100.

Nodes are grouped by node type in the search results list. You can collapse a node type section to be able to navigate to nodes only in the node type that you need to work with. For hierarchy viewpoints, the parent name of each node is also displayed in the search results. In the example below, the search results display the parents underneath each node that meet the search criteria of "111".



Corporate Account | Plan Account | Acquired Account

111

Corporate Account

11101
USBNK Checking Account
11000

11102
USBNK Treasury Account
11000

11103
USBNK2 Checking Account
11000

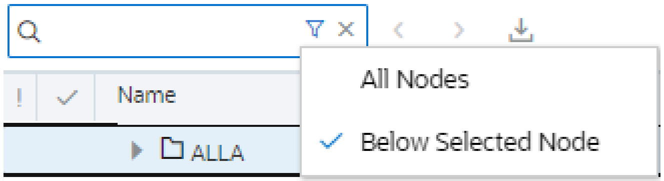
Name

T
10000
11000
11101
11102
11103
11200
11300
11501
11502
11503

In a Viewpoint

To search for a node in a viewpoint:

1. From an open viewpoint, click **Search** .
2. (Optional): To search for a node below a specific node in a hierarchy viewpoint:
   a. Navigate to the node in the viewpoint that you want to search below.
   b. In the search box, click **Filter** , and select **Below Selected Node**.



3. Enter the text to search for and click **Search** or press **Enter**.
4. Click the node in the results to locate it in the list or hierarchy.

> **Note:**
>
> If the node is a shared node, then the number of instances of the node in the hierarchy appears next to the node in the results box. If you selected the Search Below Node option, only the number of instances of the node below the selected node in the hierarchy are displayed.
> You can move to the next or previous instance of the shared node by clicking the arrows next to the node in the hierarchy.

In the Node Selector

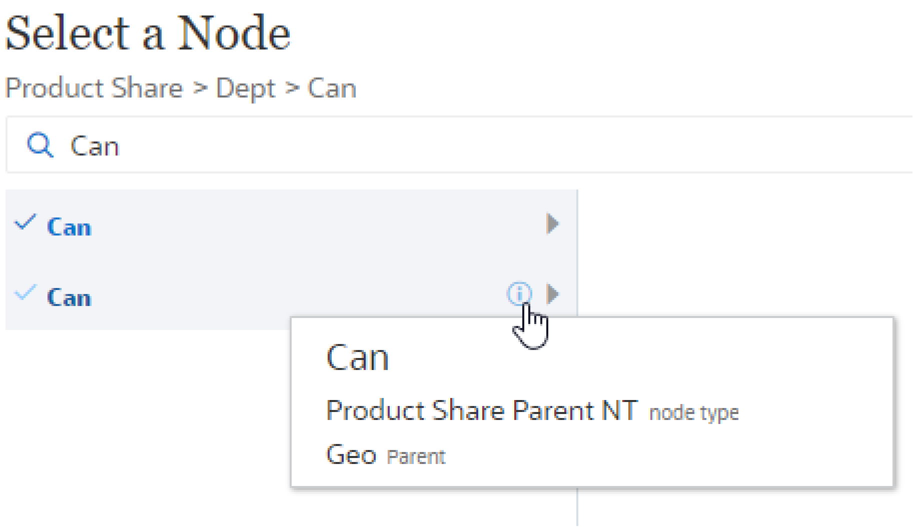
In the node selector, descriptions are displayed next to each node by default. Searches in the node selector locate the search text in node names and descriptions. You can turn node descriptions off so that they don't display in the node selector. If you turn descriptions off, a search in the node selector is performed on node names only.

To search for a node in the node selector:

1. From the node selector, enter the text to search for and click **Search** or press **Enter**.

> **Note:**
>
> By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.
>
> Selecting a different property changes the display only. You cannot search for properties other than Name and Description.

Multiple search results are displayed if the node text that you are searching for appears under more than one parent in the hierarchy. Click the **Information** ⓘ icon to display the node type and parent information for each node returned in the search results.



Select a Node

Product Share > Dept > Can

Can

✓ Can

✓ Can

Can

Product Share Parent NT node type

Geo Parent

> **Note:**
>
> Nodes that are implicitly shared (that is, nodes that exist under the same parent in multiple locations in a hierarchy) are displayed only once in search results. See Understanding Shared Nodes

2. Click on a specific node instance in the results to select it.

Querying a Viewpoint

Viewpoint queries enable you to search for nodes in a viewpoint whose properties match your specified criteria.

While viewpoint *searches* enable you to search for nodes in a viewpoint by name or description only (see Searching for Nodes), viewpoint *queries* expand that capability in several ways:

- You can query on properties beyond just name and description.
- You can add multiple query filters with different property values to further refine your query.
- You can use different operators for different properties (such as finding nodes that were added between two dates).

> **Note:**
>
> You cannot query time labeled viewpoints.

Considerations

- To add a property to a query filter, the following must be true:
  - For List viewpoints, only node level properties can be selected. For Hierarchy viewpoints, node and relationship level properties can be selected.
  - The user creating the query must be able to see the property in the viewpoint. The property cannot be hidden as a result of the user's data access or viewpoint security.
- Properties in viewpoints are either indexed (indicated by a icon) or not indexed. Indexed properties are properties that have stored defined values (including values that were derived and stored) and whose default type is not specified, derived, or inherited.
  - For multiple query filters joined by AND, at least one property must be an indexed property.
  - For multiple query filters joined by OR, all properties must be indexed properties.

> **Note:**
>
> Properties with the Derived and Stored default type are able to be queried only after a value has been either defined or derived and then stored for that property on a node. This is true even if you join the Derived and Stored query filter with an indexed property query filter using an AND statement.

- When querying on the `Core.Name` property, you can query for nodes in node types that have defined a default or alternate qualifier (see Working with Node Type Qualifiers). For example, if a node type in the query has a default qualifier of "A_" and an alternate qualifier of "C_", then querying for a node name of 100 will return nodes 100, A_100, and C_100.
- The following `Core` and `CoreStats` properties can be used in viewpoint queries:
  - `Core.Name`
  - `Core.Alternate Name`
  - `Core.Description`
  - `CoreStats.Created By`
  - `CoreStats.Created Date`
  - `CoreStats.Last Modified By`
  - `CoreStats.Last Modified Date`
- You can perform a query in the context of a request. The viewpoint query is run on the "After" state of the request.

Query Visibility

Private Queries

After you create a viewpoint query, you can save it so that you don't need to recreate it the next time you want to run it. When you save a viewpoint query, it is saved as a private query, which can only be viewed and run by the user who created it.

Public Queries

If you have at least *Data Manager* or *Metadata Manager* permission on the viewpoint, you can promote a private query to a public query, which can be viewed and run by all users who have access to the viewpoint in the query. See Promoting Viewpoint Queries.

Creating, Running, and Saving Viewpoint Queries

You create and run viewpoint queries from the query pane in a viewpoint. You can save a query as a private query, which can only be viewed and run by the user who created it. Users with *Data Manager* or *Metadata Manager* permission on the viewpoint can promote a private query to a public query, which can be viewed and run by all users who have access to the viewpoint in the query.

Creating and Running Viewpoint Queries

To create and run a query in a viewpoint:

1. From an open viewpoint, click **Actions** ⋮ next to the viewpoint tab name and then select **Query**, or else click the **Query Viewpoint** tab on the left side of the page.
2. In the Query Viewpoint pane, select a viewpoint to query using the top drop down selector. You do not have to query the viewpoint that you are currently viewing. If you select a different viewpoint from the drop down menu, that view is displayed.
3. Select the nodes to search:
   - Hierarchy viewpoints:
     - **All Nodes**: Search the entire viewpoint.
     - **Below Selected Node**: Limit the query to the nodes below the currently selected node.

       When searching below a selected node, hover over the ⓘ icon to display the selected node.
     - **Orphan Nodes**: Search for nodes that are in the node types used by the hierarchy set but that aren't a top node or a descendant of a top node in the node set.
   - List viewpoints: **All Nodes**: Search the entire viewpoint.
4. If the viewpoint uses multiple node types, select **Any Node Type** to search across all node types, or select a specific node type to query on in the drop down menu. The list of properties available for use by a query filter is limited to properties for the selected node types only.
5. Click **Add a Filter**.

> **Note:**
>
> A filter is required when querying **All Nodes** or **Below Selected Node**, above. You do not need to add a filter if you have selected **Orphan Nodes**.

6. Select a property to add a filter for. The property must meet all of the conditions described in the **Considerations** section, above.
7. Select an operator for the filter. The operators that you can choose from are specific to the data type of the property that you selected. The following table lists the operators that are available for each data type.

Note:

For values that you specify (for example, when entering a Contains value for a string), the text that you enter is not case-sensitive.

Table 5-1 Operators by Property Data Type

Property Data Type	Operators
Boolean	• Is True • Is False • Is Blank • Is Not Blank
Date, Timestamp	• Equals • Before • Between • After • Is Blank • Is Not Blank
Integer, Float	• Equals • Between • Greater Than • Less Than • Is Blank • Is Not Blank
List	• Contains **Note:** You can specify a single list value to check per filter. You can add a second filter to query for another list value. • Is Blank • Is Not Blank
Node	• Equals (use the node selector to pick a node) • Is Blank • Is Not Blank
Node List	• Contains (use the node selector to pick one or more nodes) **Note:** If the node list data type property is in multiple node types, use the viewpoint dropdown in the node selector to specify the source viewpoint for the nodes. • Is Blank • Is Not Blank

Table 5-1 (Cont.) Operators by Property Data Type

Property Data Type	Operators
String, Numeric String, Memo, Sequence	• Contains You can specify multiple words (separated by a space) in the query. The search matches on the beginning of each word. For example, when searching for "Net Income Current Year": – Query containing "Net Income" will match on "Net" and "Income" – Query containing "Curr" will match on "Current" – Query containing "rent" will not match on the beginning of any string • Equals • Is Blank • Is Not Blank • Similar To: Performs a fuzzy search on the text that you enter. Enter a value to search for and, optionally, a prefix length that specifies the number of characters at the beginning of the string that must exactly match. The prefix length can be used to reduce the query results to only those that have the same characters at the beginning of the string. For example, for a similar to value of "House", if you do not specify a prefix length, the search can return "House", "Mouse", and "Blouse". If you specify a prefix length of 1, however, the first character of the similar to value must match. Therefore, the search will return only "House". For best results, use Similar To on a complete string, rather than a partial string.

8. Select a value for the filter. The value that you select is specific to the data type of the property in the filter. For example, you select a date from a calendar for the Date data type, and you enter a string value for a list.

> **Note:**
>
> For properties that use an allowed values list, the specified value is evaluated against both the Value and the Label for the property.

9. **Optional**: Repeat steps 4-7 to add additional filters.

> **Note:**
>
> When you add multiple filters, use the **And** or **Or** toggle to specify how to join the filters. The join method (And or Or) applies to all query filters. You cannot use And for some filters and Or for others.

10. When you have finished adding query filters, click **Run** ▷ to run the query.
The results are displayed in the query pane.

11. **Optional**: In the query results, perform an action:
    - **For queries other than Orphan Nodes**: Click a node in the search results to navigate to that node in the viewpoint. Nodes that meet the query criteria are also indicated by a ⚐ icon in the viewpoint.
    - **For Orphan Nodes query results**: Click the **Action** icon ⋮ next to a node in the search results, and then select **Insert Node** to open the node selector and select the parent for the node. You can also drag and drop the node from the query results to a parent node in the viewpoint. You must be in the context of a request to insert the node.

      > **Note:**
      >
      > If the orphan node has children, the number of children is indicated in the query results. If you insert the orphan node under a parent, the children of the orphan node will be inserted as well.

12. **Optional**: To download the results to an Excel file, in the Query Panel, click **Actions** ⋮, and then select **Download Query Results**.
    The viewpoint is downloaded and filtered to the nodes in the query results. The file contains a column for each property that is configured to display in the viewpoint.

    > **Note:**
    >
    > The downloaded query results may exceed the number of results that are displayed on the screen.

After you run a viewpoint query, the query filters are collapsed so that more query results can be displayed. Click **Expand** › to display the query filters again.

To rerun the query, click **Refresh** ⟳.

Editing and Removing Query Filters

To edit a query filter, click on the filter in the query pane. You can change the property, operator, and value for the filter, and then click **Run** ▷ to run the modified query.

> **Note:**
>
> If you change the selected node when searching with the **Below Selected Node** option, you are prompted to select whether you want to run the same query with the original selected node or a new query with the currently selected node.

To remove a query filter, click the **X** on the filter and then click **Run** ▷ to run the modified query.

Saving Queries and Running Saved Queries

To save a query:

1. In the Query Panel, click **Actions** ⋮, and then select **Save**.
2. Enter a name, and optionally a description, and then click **Save**.

> **Note:**
>
> The Visibility field indicates that the query is saved as a private query. If you have *Data Manager* or *Metadata Manager* permission on the viewpoint you can promote it to a public query so that other users can view and run it.

To run a saved query:

1. From the Query Panel, select the query that you want to run.

> **Tip:**
>
> If you already have a query open and you want to run a different saved query, click the **Query Viewpoint** tab to return to the list of saved queries. Private queries are indicated by a **Locked** 🔒 icon. Public queries are indicated by an **Unlocked** 🔓 icon.

2. **Optional**: Edit the saved query filters.
3. Click **Run** ▷ to run the saved query.
4. **Optional**: If you have changed the query filters and you want to save the changed query, in the Query Panel, click **Actions** ⋮, and then select **Save**.

> **Note:**
>
> You can save changes to your private queries. Only users with *Data Manager* or *Metadata Manager* permission on the viewpoint can save changes to public queries.

Working with Saved Viewpoint Queries

Copying Viewpoint Queries

You can copy public queries, and any private queries that you created. When you copy a query, it is saved as a private query. Data Managers and Metadata Managers can promote it to public.

To copy a query:

1. In the Query Panel, click **Actions** ⋮, and then select **Copy**.
2. Enter a name, and optionally a description, and then click **Save**.

> **Note:**
>
> The Visibility field indicates that the query is saved as a private query. If you have *Data Manager* or *Metadata Manager* permission on the viewpoint you can promote it to a public query.

Inspecting Viewpoint Queries

Inspecting a viewpoint query enables you to view the query definition and parameters, and to change the query name and description. You can also assign a code to the query to use instead of the query name when automating the query via the REST API.

To inspect a query:

- **From an open viewpoint**: Click **Query Viewpoint** to display the list of saved queries for that viewpoint. Then, click the **Query action menu** ⋮ next to the query that you want to inspect and select **Inspect**.
- **From an open Query in a viewpoint**: Click the **Query action menu** ⋮ next to the **Run** button, and then select **Inspect**.
- **From the Viewpoint inspector**:
  1. Inspect the viewpoint that contains the query. See Inspecting a Viewpoint.
  2. On the Queries tab, click the name of the query that you want to inspect.

To edit the query name and description, or to assign a code to the query:

1. On the General tab of the query inspector, click **Edit**.
2. Edit the name and description.
3. In the **Code** field, enter a code for the query.
   The code that you enter can be used in the `queryCode` parameter in the Run a Viewpoint Query Profile REST API instead of the `queryName` to identify the query in the API. This allows the query name to be changed without any effect to automated processes.
4. Click **Save**.

Promoting Viewpoint Queries

You must have *Data Manager* or *Metadata Manager* permission on the viewpoint in order to promote a query.

To promote a query:

1. Perform an action:
   - From an open viewpoint, click **Query Viewpoint** to display the list of saved queries for that viewpoint.
   - Inspect the viewpoint that contains the query that you want to promote. See Inspecting a Viewpoint.

2. Click **Action** ⋮ next to the query that you want to promote, and then select **Promote**.
3. Click **Save**.

> **Note:**
>
> If the query that you promote has the same name as an existing public query, an incrementing numeric suffix is appended to the query to make it unique.

Adding a Viewpoint Query to Your Favorites List

You can add a viewpoint query to your list of Favorites by clicking **Action** ⋮ next to the query and then selecting **Add Favorite**. Viewpoint queries in your Favorites list are indicated by a ★ icon. You can remove a viewpoint query from your Favorites list by selecting **Undo Favorite**. See Working with Favorites.

Deleting Viewpoint Queries

You can delete your private queries from the Query panel or the viewpoint inspector. If you have *Data Manager* or *Metadata Manager* permission on the viewpoint, you can delete public queries as well.

To delete a query:

- **From the Query panel in a viewpoint**: From the list of queries in the Query panel, click the **Query action menu** ⋮ next to the query that you want to delete, and then select **Delete**.

> **Tip:**
>
> If you already have a query open and you want to go back to the query list, click the **Query Viewpoint** tab to return to the list of saved queries.

- **From the Viewpoint inspector**:
  1. Inspect the viewpoint that contains the query that you want to delete. See Inspecting a Viewpoint.
  2. Click **Action** ⋮ next to the query that you want to delete, and then select **Delete**.
  3. Click **Save**.

Downloading a Viewpoint

You can download data in a viewpoint to a spreadsheet file so that you can use it offline, share it with other resources, or make mass changes to the data. After making changes to the file, you can then load the file into a request to apply the changes to Oracle Fusion Cloud Enterprise Data Management.

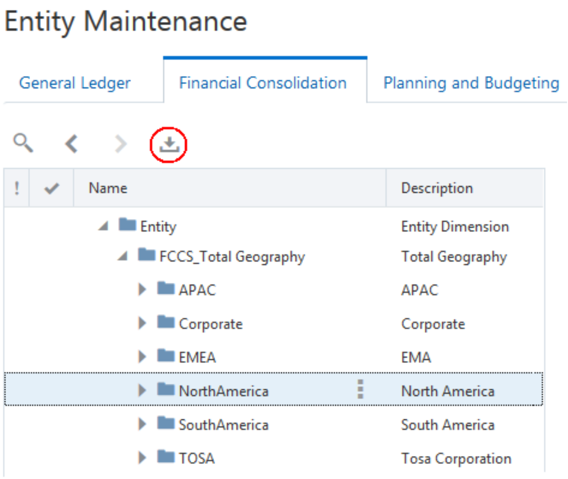
Considerations

- When downloading a hierarchy viewpoint, you have the option to indent either the name, or the name and the description of the node in the downloaded file to make it easier to read.

  > **Note:**
  >
  > When you select these options, whitespace is added to the cells in the Name or Name and Description column (depending on what you select) that corresponds with the level of the node. The downloaded files with indented cells can still be used in viewpoint loads and request file uploads.

- You can download just the column headers for a viewpoint in order to create a template for loading data into the viewpoint.
- For a hierarchy viewpoint, you can also download a specific node and its descendants. This enables you to copy a node's branch from one viewpoint to another.

  For example, you can download the North America node and all of its descendants from the hierarchy viewpoint below and then load it to another viewpoint.



!	✓	Name	Description
		Entity	Entity Dimension
		FCCS_Total Geography	Total Geography
		APAC	APAC
		Corporate	Corporate
		EMEA	EMA
		NorthAmerica	North America
		SouthAmerica	South America
		TOSA	Tosa Corporation

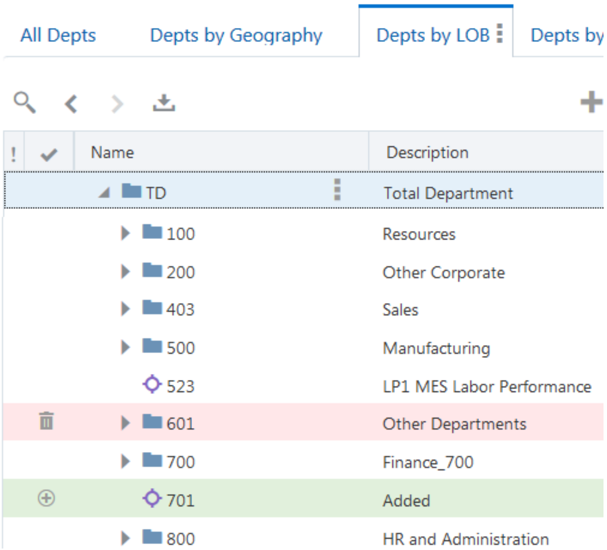
  > **Note:**
  >
  > When downloading a node and its descendants, you can optionally indent the name, or the name and the description in the downloaded file.

- When you download a viewpoint to a file, the viewpoint name must be 30 or fewer characters or a label must be set up. See Inspecting a Viewpoint for information on using a label.

If the viewpoint that you're downloading from contains visualized changes for an uncommitted request, the proposed changes are included in the output file. In other words, the output file contains data as if the request had already been committed.

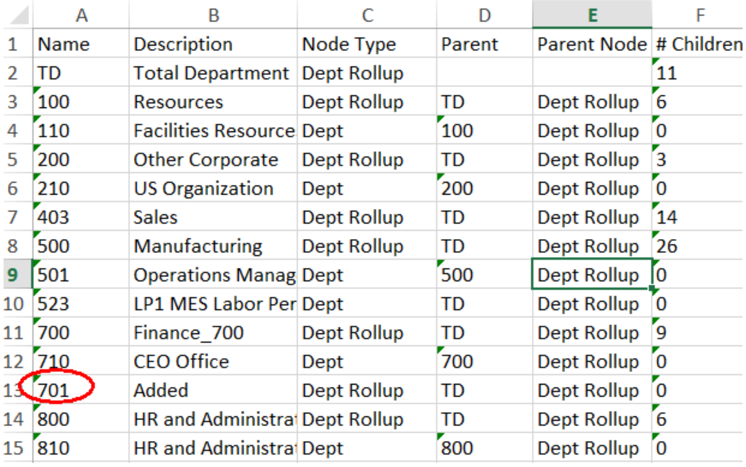
- Nodes that were added, inserted, moved, updated, renamed, or reordered in the request are included in the output
- Removed or deleted nodes (and their descendants) are excluded from the output.

For example, in the Depts by LOB viewpoint, you remove node 601 and add node 701.



!	✓	Name	Description
		TD	Total Department
		100	Resources
		200	Other Corporate
		403	Sales
		500	Manufacturing
		523	LP1 MES Labor Performance
🗑		601	Other Departments
		700	Finance_700
⊕		701	Added
		800	HR and Administration

When you download the viewpoint with request changes visualized, the results include node 701 and exclude node 601.



	A	B	C	D	E	F
1	Name	Description	Node Type	Parent	Parent Node	# Children
2	TD	Total Department	Dept Rollup			11
3	100	Resources	Dept Rollup	TD	Dept Rollup	6
4	110	Facilities Resource	Dept	100	Dept Rollup	0
5	200	Other Corporate	Dept Rollup	TD	Dept Rollup	3
6	210	US Organization	Dept	200	Dept Rollup	0
7	403	Sales	Dept Rollup	TD	Dept Rollup	14
8	500	Manufacturing	Dept Rollup	TD	Dept Rollup	26
9	501	Operations Manag	Dept	500	Dept Rollup	0
10	523	LP1 MES Labor Per	Dept	TD	Dept Rollup	0
11	700	Finance_700	Dept Rollup	TD	Dept Rollup	9
12	710	CEO Office	Dept	700	Dept Rollup	0
13	701	Added	Dept Rollup	TD	Dept Rollup	0
14	800	HR and Administra	Dept Rollup	TD	Dept Rollup	6
15	810	HR and Administra	Dept	800	Dept Rollup	0

To download a viewpoint:

1. From **Views**, open an active view.
2. Select a viewpoint.
3. Perform an action:
   - To download the entire viewpoint:
     a. Move your cursor to the right of the viewpoint name, click the menu icon ⋮, and then select **Download to File**.

  b. In the download window, select an option:
     i. **Download viewpoint data**: Downloads column headers and all of the data in a viewpoint.
     ii. **Download viewpoint data (indented)**: Downloads column headers and all of the data in a viewpoint, and adds whitespace characters to indent the Name column (and the Description column, if you select the **Indent Description** checkbox) corresponding with the level of the node.
     iii. **Download column headers only**: Downloads only the column headers in a viewpoint.

> **Note:**
>
> If your viewpoint contains specialty node types, you can download column headers only.

- To download a specific node and its descendants in a hierarchy viewpoint, select the node, click the **Download** ⤓ button, and select an option:
  a. **Download Selected**: Downloads the currently selected node and all of its descendants.
  b. **Download Selected (Indent Name)**: Downloads the currently selected node and all of its descendants and adds whitespace characters to indent the Name column corresponding with the level of the node.
  c. **Download Selected (Indent Name and Description)**: Downloads the currently selected node and all of its descendants and adds whitespace characters to indent the Name and Description columns corresponding with the level of the node.

Deleting a Draft Viewpoint

You can delete a draft viewpoint if you no longer need it. You can't delete active viewpoints, but you can archive them from the viewpoint inspector.

To delete a draft viewpoint:

1. From **Views**, inspect a view.
2. Select **Definition** and then click **Edit**.
3. Click ⋮ next to a draft viewpoint, and then select **Delete**.
4. Confirm the deletion, and then click **Save**.

For more information, see:

- Understanding Viewpoints
- Understanding the Lifecycle of Data Objects and the Data Chain

Inspecting a Viewpoint

From the viewpoint inspector, you can view and edit information about the viewpoint.

To inspect a viewpoint:

1. From an open view, click ⋮ next to the viewpoint and then select **Inspect**.
2. Select from the options available on these tabs:
   - **General**: Provides information on the data chain objects and the binding status of a viewpoint. See Binding Status.

     > **Note:**
     >
     > If a viewpoint is currently in a blockout, the application or dimension that is blocked out is indicated by a blockout icon (📅). Click the link to inspect the application or dimension and view the blockout details.

     Click **Edit** to change the name, description, and status for the viewpoint or to add a label, see Understanding the Lifecycle of Data Objects and the Data Chain.

     You'll need to create a label if the viewpoint name has more than 30 characters and you need to do any of these tasks for the viewpoint:

     – Load request items from a file
     – Download request items to a file
     – Download the viewpoint to a file
     – Generate subscription requests
     – Drag and drop request items or nodes across viewpoints
     – Create request items from viewpoint comparisons

     The label can contain up to 30 characters but cannot contain any of these special characters:

     \ / * [ ] : ?

     > **Note:**
     >
     > For time labeled viewpoints, you can also change the time label for the viewpoint. However, you cannot deselect the time label to convert the viewpoint to a normal viewpoint. See Working with Time Labeled Viewpoints.

   - **Definition**: Provides information on node type, permissible actions, and editable properties. You can set which actions are allowed for the node types, determine which properties display, and set which properties are editable, see Changing Viewpoint Permissible Actions and Configuring How a Viewpoint Displays Properties.
   - **Properties**: Provides the list of selected properties from the Definition tab. You can edit property names and descriptions, select which properties to display as columns in the viewpoint, select which properties to display in the Locations tab of the viewpoint, and reorder the list of properties, see Configuring How a Viewpoint Displays Properties, Configuring Locations Tab Properties, and Changing the Order of Displayed Properties.
   - **Subscriptions**: Provides a list of subscriptions to other viewpoints. You can create new subscriptions and edit, enable, disable, or delete existing subscriptions, see Subscribing to Viewpoints.

- **Lineage**: Provides a diagram for the viewpoint that shows the subscriptions in which the viewpoint is either a source or a target. See Viewing Viewpoint Lineage.
- **Data Chain**: Provides a clickable graphic so that you can see the node types, hierarchy sets (if applicable), and node sets that comprise the viewpoint.

Changing Viewpoint Permissible Actions

You can select the types of actions that are allowed in a viewpoint. For example, you might configure a viewpoint so that nodes cannot be deleted from the viewpoint.

You can set which of these types of node actions are allowed in a viewpoint:

- **Add**—Allows you to add nodes to the viewpoint
- **Insert**—Allows you to insert nodes in the viewpoint
- **Move**—Allows you to move nodes within the viewpoint.
- **Remove**—Allows you to remove a node from a parent in the hierarchy set. If the node has other relationships in the hierarchy set, they are not affected by removing the node from a single parent.
- **Delete**—Allows you to delete a node from a list or from all hierarchy relationships for the application.
- **Reorder**—Allows you to reorder nodes that are under the same parent.

To edit which node actions are allowed for a viewpoint:

1. From **Views**, inspect a view.
2. Select **Definition** and then click the viewpoint name to inspect.
3. Select **Definition** and then click **Edit**.
4. Change the permissible actions by selecting or clearing actions, and then click **Save**.

Configuring How a Viewpoint Displays Properties

You can configure the properties displayed in a viewpoint and their labels, whether users can edit the properties, and whether the properties are required in the viewpoint.

> **Tip:**
>
> These configurations apply to a specific viewpoint. For example, a property can be displayed with its original name in one viewpoint and with a different label in another viewpoint.

For more information, see Working with Properties.

To configure a viewpoint's properties:

1. Open the **View Inspector** for the viewpoint:
   a. Click **Views**, in the **Actions** column for the view, click ⋮, and then select **Inspect**.
   b. Select Definition and in the **Name & Description** column, click the link for the viewpoint.

2. **To specify the properties displayed in the viewpoint and whether they are editable or required:**

   a. Select the **Definition** tab.

      The **Editable Properties** section lists the properties displayed by the viewpoint and indicates whether those properties are editable. If the viewpoint contains multiple node types, the first node type's properties are displayed.
      Note that in some circumstances, a property may be read-only but the clear option for the property is still available.

Property Editable Scenario	Property in UI Updates	Clear option in UI	File Request Updates	File Request Clears
If Property Editable (viewpoint) is False	Read only	Not Available	Skipped	Skipped
If Property Editable (viewpoint) is True and Property Editable (application override) is False	Read only	Available	Skipped	Valid
If Property Editable (viewpoint) is True and Property Editable (node type override) is False	Validation error	Available	Validation Error	Valid

   b. If the viewpoint contains multiple node types, select the node type that contains the property.

   c. Click **Edit**.

   d. To specify whether a property is editable in the viewpoint, use the check box in the **Editable** column.

   e. To specify whether or not the property is required in the viewpoint, use the check box in the **Required** column.

> **Note:**
>
> You can also specify that the property is required by inspecting the node type (see Adding, Removing, and Configuring a Node Type's Properties). If a property is required at the node type level, you cannot clear that status at the viewpoint level.

   f. To specify which properties are displayed:

      i. Click **Manage**.
      A dialog box displays. The **Selected** column shows the properties currently displayed in the viewpoint. The second column displays all the viewpoint's properties, with check marks next to the properties displayed in the viewpoint.

      ii. To display a property that is currently not displayed, select it.
      A check mark is displayed next to the property name.

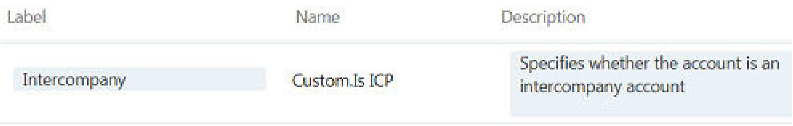
      iii. To stop a property from being displayed, select it.
      The check mark is removed from the property name.

   g. Click **Save**.

3. **To configure how the viewpoint displays properties:**
   a. On the **Properties** tab, click **Edit**.
   b. **Optional**: Click **Sort by** ⇅ to perform a one time sort of the property list alphanumerically by either **Label** or **Namespace and Name**.
   c. **To display a label for a property:** Edit the property's text box in the **Label** column.

      > **Note:**
      >
      > Viewpoint labels must be unique for each property in a viewpoint.
      >
      > See Reserved Column Names for a list of labels that cannot be used for properties in a viewpoint.

      In the following example, the viewpoint will display `Intercompany` as a label for the `Custom.Is ICP` property.

Label	Name	Description
Intercompany	Custom.Is ICP	Specifies whether the account is an intercompany account

   d. **To change a property description:** Edit the property's text box in the **Description** column.
   e. **To add the property to one or more property groups:** Click in the Property Groups column for that property and either select an existing property group or enter the name of a new property group to add the property to.

      The new property group is added to the view and can be managed in the view inspector. See Creating, Editing, and Deleting Property Groups

      > **Note:**
      >
      > A property can belong to more than one group.

   f. **To specify that a property should be displayed as a column in the viewpoint**: Select the **As Column** check box for that property.

      You can select up to 10 properties to display. The name property is required to be selected and counts as one of the 10 properties.
   g. **To add the property to the Locations tab of the property pane:** Select the Locations Tab checkbox. See Configuring Locations Tab Properties.
   h. **To specify the order in which properties are displayed in the viewpoint** (or to restore a property's original name or description): Use the commands on the menu that displays when you place the cursor in the property's row and click ⋮. See Changing the Order of Displayed Properties.

> **Note:**
>
> If you change the display order for properties that you are displaying as columns, the display order for the columns and the properties panel are affected.

i. When you are done configuring properties, click **Save**.

   The **Label** and **Description** columns become read-only.

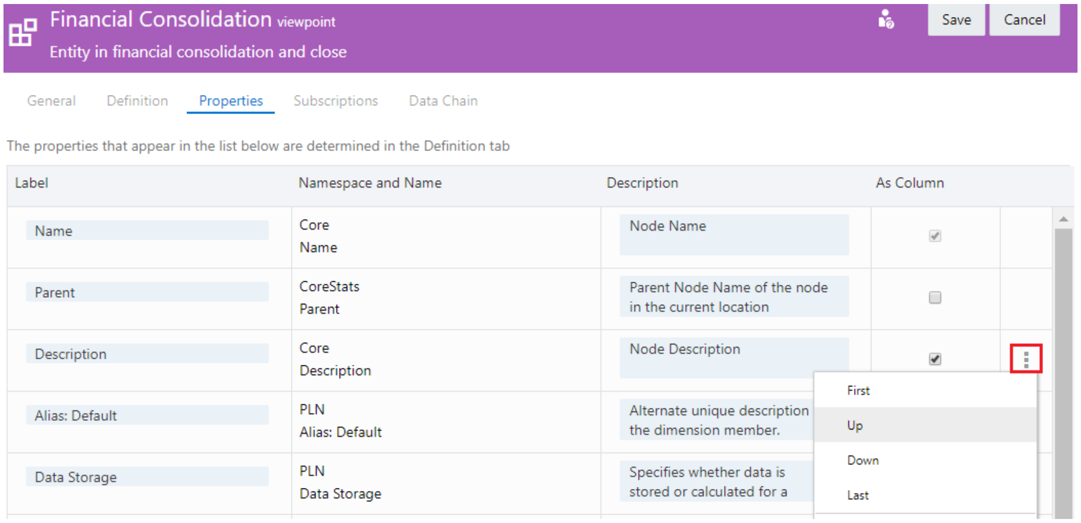
Changing the Order of Displayed Properties

You can configure the order in which properties display in a viewpoint. Display order is specific to a viewpoint so you could have different display orders for different viewpoints.

For more information, see Configuring How a Viewpoint Displays Properties.

To configure the order in which properties display:

1. From an open view, select the viewpoint that you want to configure.
2. Place your cursor to the right of the viewpoint name, click the menu icon ⋮, and then select **Inspect**.
3. Select the **Properties** tab and then click **Edit**.
4. In the far right column, hover the cursor in the cell of the property that you want to move, click ⋮ and then select how you want to move the property:
   - First - Moves the property to be the first property in the list.
   - Up - Moves the property up one row in the list. Use this selection multiple times to move the property up multiple rows.
   - Down -Moves the property down one row in the list. Use this selection multiple times to move the property down multiple rows.
   - Last - Moves the property to be the last property in the list.



Financial Consolidation viewpoint
Entity in financial consolidation and close

The properties that appear in the list below are determined in the Definition tab

Label	Namespace and Name	Description	As Column
Name	Core Name	Node Name	☑
Parent	CoreStats Parent	Parent Node Name of the node in the current location	☐
Description	Core Description	Node Description	☑
Alias: Default	PLN Alias: Default	Alternate unique description the dimension member.	
Data Storage	PLN Data Storage	Specifies whether data is stored or calculated for a	

5. Repeat until the properties are in the desired display order and then click **Save**.

Downloading and Uploading Properties for a Viewpoint

When configuring properties for a viewpoint, you can download the property definitions for a viewpoint to an Excel file for offline reference or to reuse a similar configuration across multiple viewpoints, and you can upload an Excel file with the property definitions instead of configuring them manually through the user interface.

Downloading Properties for a Viewpoint to a File

> **Tip:**
>
> This procedure describes how to download the property configuration in a specific viewpoint. To download property definitions for a single property or for all properties from the property worklist, see Downloading Property Definitions.

To download properties to a file:

1. Inspect the viewpoint that you want to upload properties to. See Inspecting a Viewpoint.
2. On the Properties tab, click **Download Properties to File** ⤓.

   > **Note:**
   >
   > The **Download Properties to File** button is not visible if you are in Edit mode.

After the file is downloaded, you can view it in Excel.

Uploading Properties for a Viewpoint from a File

Considerations when Uploading Properties

- You must have at least one of these roles or permissions when uploading properties:
  - Service Administrator
  - *Owner* permission on the view containing the viewpoint, and at least *Data Manager* or *Metadata Manager* permission on the application or dimension associated with the viewpoint
- The properties in the viewpoint are replaced by the properties in the uploaded file, in the order that they are listed in the file. Therefore, your upload file *must* contain all of the properties for the viewpoint in the order that you want them in the viewpoint. Partial uploads or merge uploads are not supported.
- Property group associations for properties are replaced by the property groups in the uploaded file. Therefore, your upload file must contain all of the property group associations for all of the properties for the viewpoint.
- Invalid or duplicate properties in the viewpoint are skipped when the upload is processed.
- Default values for properties are populated when needed.

> **Tip:**
>
> You can download the current properties for the viewpoint to use as a template that you can modify with additional changes and then upload back to the system. See Downloading Properties for a Viewpoint to a File.

To upload properties to a viewpoint:

1. Inspect the viewpoint that you want to upload properties to. See Inspecting a Viewpoint.
2. On the Properties tab, click **Edit**.
3. Click **Load Viewpoint Properties from File** ⤒.

   > **Note:**
   >
   > You must be in Edit mode to see the **Load Viewpoint Properties from File** button.

4. Browse to the properties file, and then click **Open**.

   > **Note:**
   >
   > Your properties file must be in the correct format. See Viewpoint Properties File Format.

   The property file is processed and the results are displayed.
5. **Optional**: In the result message, for Successful or Successful with Warnings results, click the link to download the processed properties file and review the **Status** and **Message** columns to see the results of each property. If the upload was unsuccessful, review the error message.
6. Click **OK** on the confirmation dialog box, and then click **Save** to save your changes.

   > **Note:**
   >
   > If you click **Cancel**, the properties that you uploaded are not saved to the viewpoint.

Viewpoint Properties File Format

This topic describes the format of files used to load the configuration of properties displayed in a viewpoint. The viewpoint properties file must be an Excel file with a sheet named Properties.

> **Note:**
>
> The file can contain other sheets. For example, when you download properties the file contains a sheet named Summary with summary information. This summary sheet is informational and is not required for upload.

Properties Sheet Format

Considerations

- All header records must be included in the file, in the order specified below.
- There must be a record in the file for each visible property in the viewpoint.
- If a property in the file is not a visible property for the viewpoint, the row is skipped.
- If there are multiple records for the property in the file, the first record in the file is processed. Any subsequent records are skipped and marked with a skipped status and a corresponding message.

Header Columns

- **Property** (Required): The fully-qualified name of the property (for example, `CoreStats.Parent`)
- **Label**: Custom label for the property in the viewpoint
- **Description**: Custom description for the property
- **Property Groups**: (Required) The property groups to associate the property with. See Creating, Editing, and Deleting Property Groups.

  > **Note:**
  >
  > – If the property group exists, the system adds a property association to that group.
  >
  > – If the property group does not exist, the system creates it and then adds the property association to that group.

- **Display as Column**: Boolean value (TRUE/FALSE) to designate whether the property should be displayed as a column in the viewpoint. See Configuring How a Viewpoint Displays Properties.
- **Display for Locations**: Boolean value (TRUE/FALSE) to designate whether the property should be displayed in the Locations tab in the viewpoint. See Configuring Locations Tab Properties.

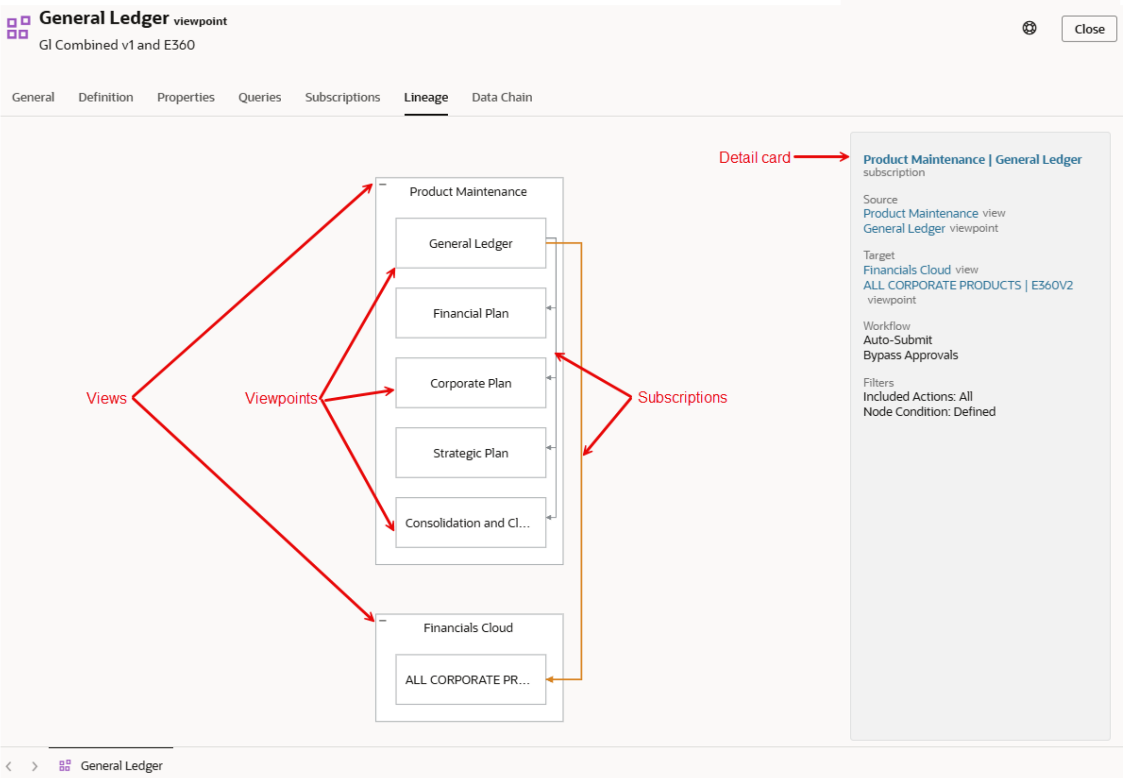
Status Columns

- **Status**: System reserved column that displays the status of the property (**Success** or **Skipped**) after you upload it.
- **Message**: System reserved column that displays an informational message on properties that were skipped.

Viewing Viewpoint Lineage

The Lineage tab of the viewpoint inspector displays a diagram for the viewpoint that shows the subscriptions in which the viewpoint is either a source or a target.

This enables you to see how viewpoints across different applications are connected by subscriptions and how changes are synchronized from one viewpoint to other viewpoints.



Product Maintenance | General Ledger
subscription

Source
Product Maintenance view
General Ledger viewpoint

Target
Financials Cloud view
ALL CORPORATE PRODUCTS | E360V2 viewpoint

Workflow
Auto-Submit
Bypass Approvals

Filters
Included Actions: All
Node Condition: Defined

Each view in the diagram is represented by a box, and the viewpoints in the view are displayed within that box. Only viewpoints with a subscription to or from the current viewpoint are displayed. Subscriptions are represented by arrows between viewpoints. Only subscriptions that are enabled are displayed.

Click a view, viewpoint, or subscription to view details about that item in the detail card. From the detail card, click the name or a view, viewpoint, or subscription to inspect that item.

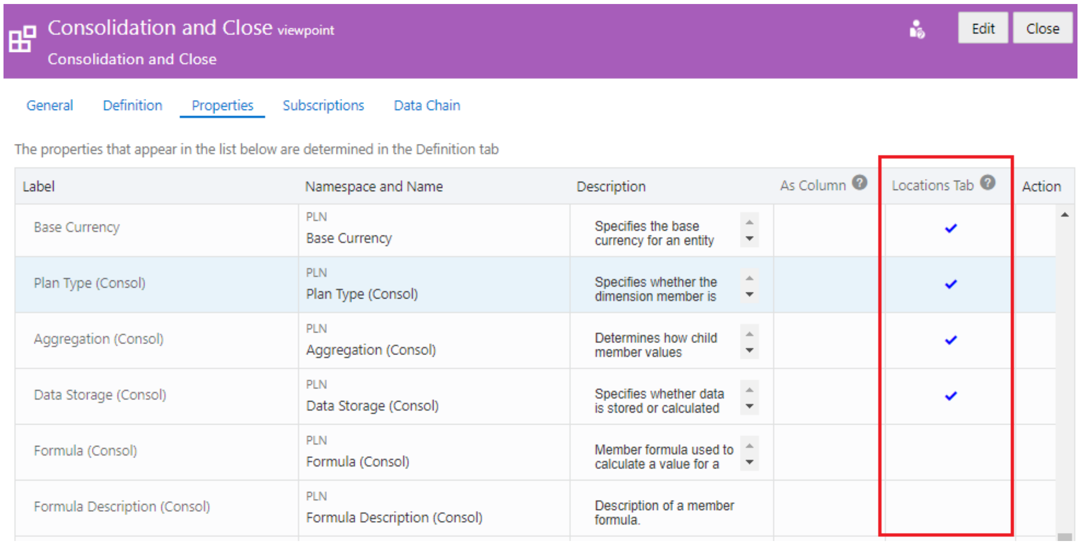
Configuring Locations Tab Properties

The Locations tab identifies all locations of a selected node in a hierarchy viewpoint. Property values for each location can be displayed to determine similarities or differences across locations. You set up the properties to display in the viewpoint inspector. Up to 10 properties can be displayed.

> **Tip:**
>
> Only relationship properties, positional inheriting properties, or derived properties may be different by location.

To configure the properties that display on the Locations tab of a viewpoint:

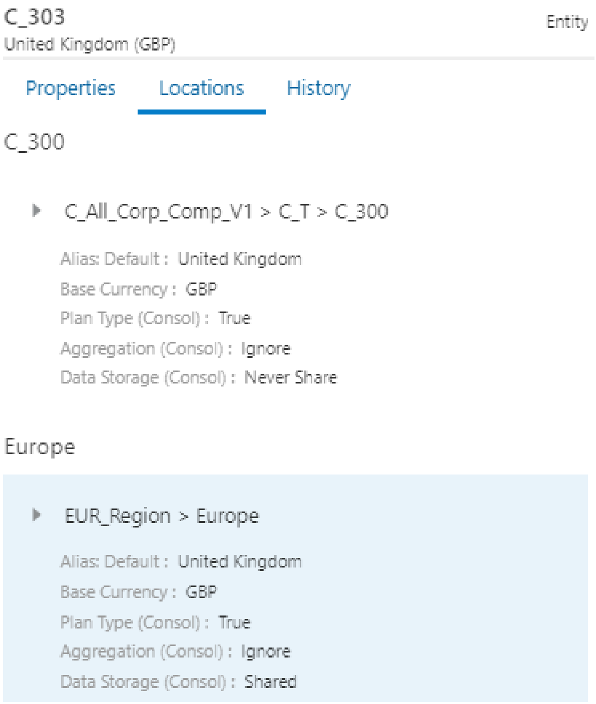
1. From an open view, select the viewpoint that you want to configure.
2. Place your cursor to the right of the viewpoint name, click the menu icon ⋮, and then select **Inspect**.
3. Select the **Properties** tab and then click **Edit**.
4. Select the **Locations Tab** check box for properties that you want displayed, and then click **Save**.



The properties that appear in the list below are determined in the Definition tab

Label	Namespace and Name	Description	As Column	Locations Tab	Action
Base Currency	PLN Base Currency	Specifies the base currency for an entity		✓	
Plan Type (Consol)	PLN Plan Type (Consol)	Specifies whether the dimension member is		✓	
Aggregation (Consol)	PLN Aggregation (Consol)	Determines how child member values		✓	
Data Storage (Consol)	PLN Data Storage (Consol)	Specifies whether data is stored or calculated		✓	
Formula (Consol)	PLN Formula (Consol)	Member formula used to calculate a value for a			
Formula Description (Consol)	PLN Formula Description (Consol)	Description of a member formula.			

For example, we can compare the properties for C_303 in two locations:



C_303
United Kingdom (GBP)

Entity

C_300

C_All_Corp_Comp_V1 > C_T > C_300

Alias: Default : United Kingdom
Base Currency : GBP
Plan Type (Consol) : True
Aggregation (Consol) : Ignore
Data Storage (Consol) : Never Share

Europe

EUR_Region > Europe

Alias: Default : United Kingdom
Base Currency : GBP
Plan Type (Consol) : True
Aggregation (Consol) : Ignore
Data Storage (Consol) : Shared

Configuring Related Viewpoints

Related viewpoints extend the Model After operation by enabling you to create nodes with the same property values and hierarchy relationships as a source node in viewpoints other than the source viewpoint. See Adding a Node by Modeling After an Existing Node.

Related viewpoints are also used in the **Equivalent Nodes with Filter** validation (for all application types except User) and the **Equivalent Nodes** validation (for Oracle Financials Cloud General Ledger applications). See Equivalent Nodes Validations and Related Viewpoints.

Considerations

- Related viewpoints must be in the same view and use the same node type as the viewpoint that you are modeling from. Node type converters are not supported for related viewpoints.
- If you delete or archive a viewpoint that is a related viewpoint, it is removed from the list of related viewpoints. For archived viewpoints, if you subsequently unarchive the viewpoint it is not automatically added back to the list of related viewpoints. You must add the viewpoint back manually.
- You cannot configure a time labeled viewpoint as a related viewpoint.

Equivalent Nodes Validations and Related Viewpoints

When you configure related viewpoints, there are two predefined validations that ensure that nodes in a specified viewpoint also exist in one or more related viewpoints:

- **Equivalent Nodes**: This validation is available for Oracle Financials Cloud General Ledger applications only, and it is enabled by default. It ensures that bottom nodes (that is, nodes where the summary flag is set to False) exist in all related viewpoints in at least one location. See Oracle Financials Cloud General Ledger Validations.

- **Equivalent Nodes with Filter**: This validation is available for all application types except Users applications, and it is disabled by default. It ensures that all nodes that match a specified filter expression (or all nodes, if there is no filter expression) exist in all related viewpoints in at least one location.

To configure related viewpoints:

1. Inspect the viewpoint that you want to specify a related viewpoint for.
2. On the Definition tab, click **Edit** and then locate the **Related Viewpoint** section. You may have to scroll down to see the field.
3. Click in **Related Viewpoints** to display a list of viewpoints in the same view that use the same node type.
4. Click a viewpoint to add it to the list of related viewpoints. You can select more than one viewpoint as related viewpoints. Click the **X** to remove a viewpoint from the list of related viewpoints.
5. **Optional**: In **Equivalent Nodes Filter**, click **Edit Expression** fx and enter an expression to specify that nodes in this viewpoint that match the expression filter must exist in all related viewpoints in at least one location. If no expression is entered, then all nodes in the viewpoint must exist in all related viewpoints in at least one location.

Using Duplicate Viewpoint Tabs

You can open a viewpoint in a duplicate tab within a view in order to work with multiple hierarchies within that viewpoint.

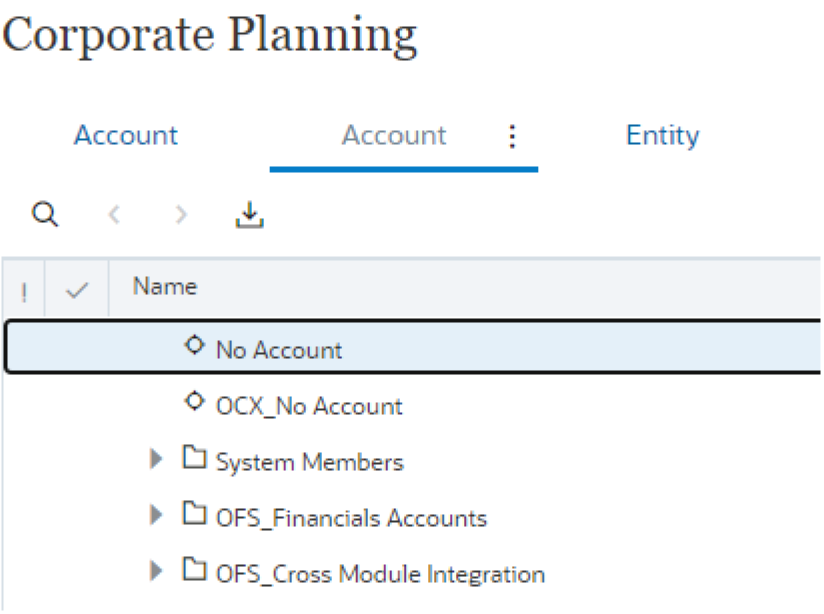
Considerations

- You can perform all of the same operations in the duplicate viewpoint tab as in the original (such as browsing, searching, and performing and validating request actions).
- You can display the original and the duplicate viewpoint tab in a side by side layout (see Displaying Viewpoints Side By Side), and then perform operations across the viewpoint tabs such as:
  - Locating nodes
  - Aligning properties
  - Comparing nodes, relationships, and properties
  - Correcting node differences by dragging and dropping nodes or creating request items from the comparison results
- A viewpoint can be opened in a duplicate tab one time only. After you open a viewpoint in a second tab, you cannot open another instance of the same viewpoint in a third tab.
- After you are finished with the duplicate viewpoint tab, you can remove it from the view. If you close or navigate away from the view, all duplicate viewpoint tabs are removed from the view.

> **Note:**
>
> Opening a viewpoint in a new tab is not the same as copying a viewpoint. Copying a viewpoint creates a new, separate viewpoint in the view, with potentially new nodes and relationships in that viewpoint (see Copying a Viewpoint), whereas opening a viewpoint in a new tab merely displays a single viewpoint in two different tabs so that you can easily work with different sections of that viewpoint at the same time.

To open a viewpoint in a new tab:

1. Navigate to the view that contains the viewpoint that you want to open a duplicate tab of.
2. In the viewpoint that you want to open a duplicate tab of, click **Viewpoint Actions** ⋮ and then select **Open in new tab**.
   The duplicate viewpoint tab is displayed to the right of the original. The duplicate viewpoint tab has the same name as the original, but the name is displayed in a different color text to differentiate it from the original.



3. To remove the duplicate tab, either click **Viewpoint Actions** ⋮ and then select **Remove tab**, or navigate away from the viewpoint and then reopen it to remove all duplicate tabs.

Displaying Viewpoints Side By Side

You can display two viewpoints next to each other to do comparisons and locate and align nodes. If you have opened the viewpoint in a duplicate tab, you can display the original and the duplicate tab side by side to work with multiple hierarchies within that viewpoint.

To display viewpoints side by side:

1. Open a view.

2. Click **Side by Side Layout** ◫ to switch to side by side layout.
3. Use the viewpoint drop down menus on the left and right of the screen to select the viewpoints to display on each side.

For more information on making changes to a viewpoint, see Making Changes Interactively

Locating Nodes

When you display viewpoints in a side by side format, you can select a node in one viewpoint and locate the same node in the other viewpoint.

The Locate Node feature looks for the node being searched for in this order:

1. If the source node type also exists in the target viewpoint, the node is searched for in that node type by name, then alternate name, and then node link (see About Node Links).
2. If the source node type does not exist in the target viewpoint, then any node type converters that exist for the source node type are used to search for the node in other target node types by name, then alternate name, and then node link.

> **Note:**
>
> Node type converters are needed to locate nodes of different node types across two viewpoints in side by side layout, see Working with Node Type Converters.

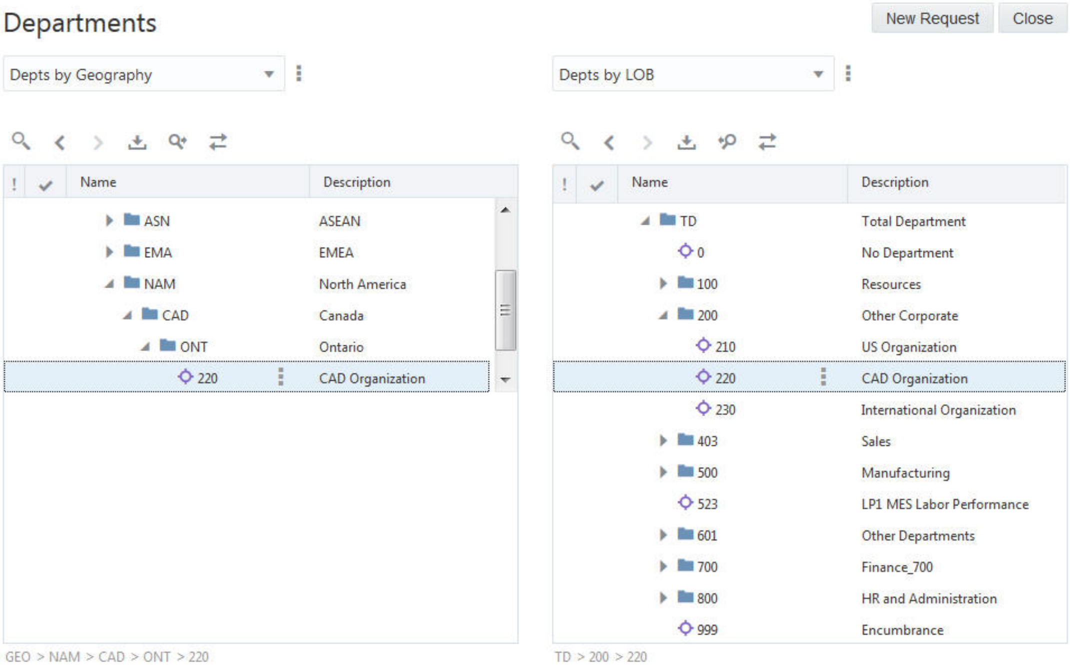
When you locate nodes between hierarchy viewpoints, the Locate Node feature attempts to find the same node under the same parent in the target viewpoint as in the source viewpoint. If the node does not exist under that parent in the target viewpoint, the Locate Node feature finds the first occurrence of the node.

Locate is a one-time find of a node. If you want to locate multiple nodes, see Aligning Nodes.

You can locate a node by:

- Selecting a node in a viewpoint.

  For example, from the Depts by Geography viewpoint, you select node 220 and click **Locate Node** 🔍 to find it in the Depts by LOB viewpoint.



Departments

New Request | Close

Depts by Geography

Name	Description
ASN	ASEAN
EMA	EMEA
NAM	North America
CAD	Canada
ONT	Ontario
220	CAD Organization

GEO > NAM > CAD > ONT > 220

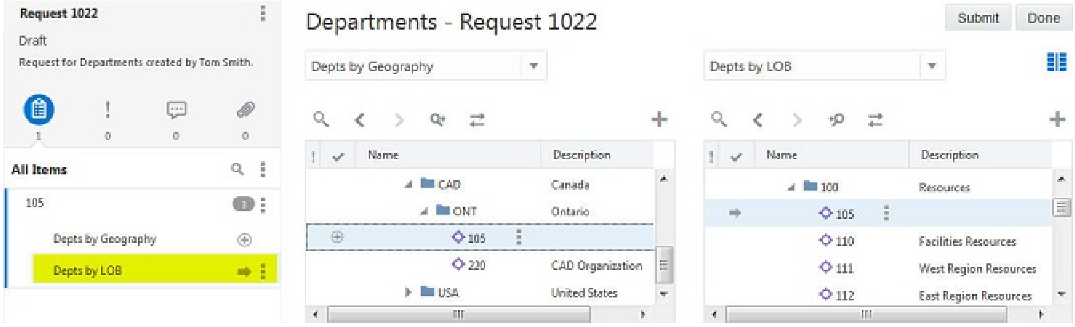
Depts by LOB

Name	Description
TD	Total Department
0	No Department
100	Resources
200	Other Corporate
210	US Organization
220	CAD Organization
230	International Organization
403	Sales
500	Manufacturing
523	LP1 MES Labor Performance
601	Other Departments
700	Finance_700
800	HR and Administration
999	Encumbrance

TD > 200 > 220

- Selecting the node in a request item.

  For example, from the request item for node 105, you select Depts by LOB to navigate to that node in the Depts by LOB viewpoint.



Request 1022
Draft
Request for Departments created by Tom Smith.

All Items

105
Depts by Geography
Depts by LOB

Departments - Request 1022

Submit | Done

Depts by Geography

Name	Description
CAD	Canada
ONT	Ontario
105	
220	CAD Organization
USA	United States

Depts by LOB

Name	Description
100	Resources
105	
110	Facilities Resources
111	West Region Resources
112	East Region Resources

To locate a node in side by side viewpoints:

1. Display two viewpoints in side by side format
2. Select a node in one viewpoint.
3. Click 🔍 in the viewpoint where you selected the node.

> **Note:**
>
> While the locate process is running, other operations on the page cannot be performed.

The node is highlighted in the other viewpoint. The located node is a node with the same node name or alternate name and node type or, if the node types are different, the same node name or alternate name and a node type converter if applicable.

Aligning Nodes

When you display viewpoints in a side by side format, you can align nodes and properties so that you can see similarities and differences. Align mode remains on and keeps both viewpoints visually aligned as you navigate a viewpoint. For example, nodes are aligned when you click on a node in:

- A hierarchy or list
- Search results
- Request items
- Viewpoint validation results
- Navigation links (bread crumbs)

Nodes are aligned in this order:

1. If the source node type also exists in the target viewpoint, nodes are aligned using the node name, then alternate name, and then node link (see About Node Links).
2. If the source node type does not exist in the target viewpoint, nodes are aligned using any node type converters that exist for the source node type in other target node types by name, then alternate name, and then node link.

> **Note:**
>
> Node type converters are needed to align nodes of different node types across two viewpoints in side by side layout, see Working with Node Type Converters.

To align nodes across viewpoints:

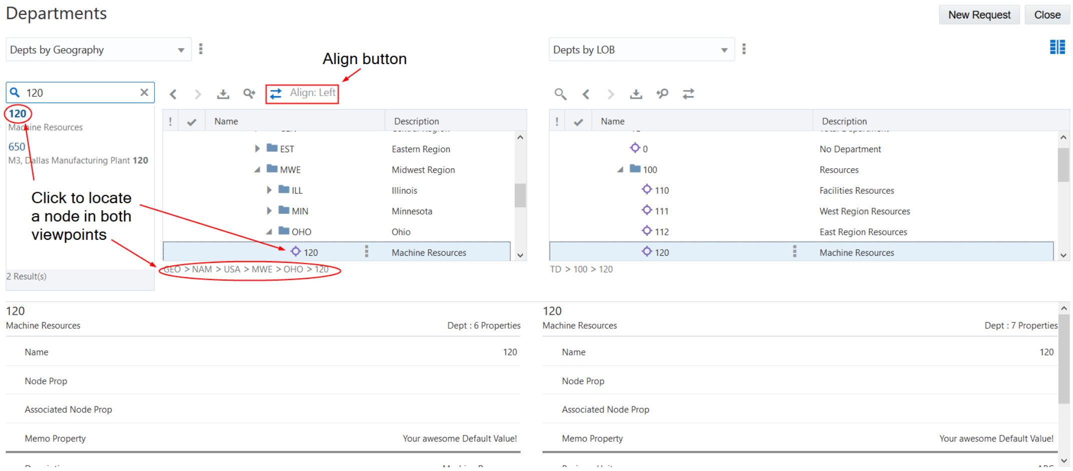
1. Select two viewpoints.

   If the two viewpoints do not use the same node type, then a node type converter needs to be defined.

2. Select a node and then click ⇄ for the viewpoint where the node was selected.

   > **Note:**
   >
   > While the align process is running, other operations on the page cannot be performed.

   The screenshot below shows that when you click ⇄ and then select node **120** in the search result, the navigation links, or the viewpoint on the left, the node is also displayed in the viewpoint on the right (if the node exists in that viewpoint). It also shows the list of properties for each viewpoint separated by a horizontal line. The properties above the line are the properties that the two viewpoints have in common or that are mapped by a node type converter. The properties below the line are ones that are different between the viewpoints.



Align button

Click to locate a node in both viewpoints

120 120
Machine Resources

650
M3, Dallas Manufacturing Plant **120**

!	✓	Name	Description
		EST	Eastern Region
		MWE	Midwest Region
		ILL	Illinois
		MIN	Minnesota
		OHO	Ohio
		120	Machine Resources

GEO > NAM > USA > MWE > OHO > 120

!	✓	Name	Description
		0	No Department
		100	Resources
		110	Facilities Resources
		111	West Region Resources
		112	East Region Resources
		120	Machine Resources

TD > 100 > 120

120
Machine Resources — Dept : 6 Properties

Name	120
Node Prop	
Associated Node Prop	
Memo Property	Your awesome Default Value!

120
Machine Resources — Dept : 7 Properties

Name	120
Node Prop	
Associated Node Prop	
Memo Property	Your awesome Default Value!

> **Note:**
>
> You can align nodes from either viewpoint.

6 Validating a Viewpoint

You can validate a viewpoint to determine if there are any data integrity issues. You might validate a viewpoint after an import process from an external source. Or you may want to validate a viewpoint before you export data from the viewpoint to an external application.

> **Note:**
>
> You cannot validate time labeled viewpoints.

Validations are run for all nodes, relationships, and properties in the viewpoint and may differ depending on the application type, dimension type, and node types for the viewpoint. See Understanding Validations and Constraints.

> **Note:**
>
> If you open a viewpoint in a duplicate tab, you can run validations from the original tab for that viewpoint only. You cannot run a validation directly from the duplicate tab. However, because both the original and the duplicate tabs display the same viewpoint, any validation issues that are displayed in the original tab will also be displayed in the duplicate tab.

By default, system validations, application and custom validations with a severity of Error or Warning (see Configuring Validation Enforcement and Severity), property validations, and constraints are run when you validate a viewpoint. You can select to run specific application and custom validations, property level validations, and constraints.

> **Note:**
>
> Application and custom validations with a severity of Ignore are not run during viewpoint validation.

To run all validations for all nodes in a viewpoint:

1. From **Views**, open an active view and select a viewpoint.
2. Place your cursor to the right of the viewpoint name, click **Actions** ⋮, and then select **Validate**.

   System validations, application and custom validations with a severity of Error or Warning, property validations, and all constraints are run.

To select specific validations to run or specific nodes to validate:

1. From **Views**, open an active view and select a viewpoint.

2. Click **Validate** from the left side of the page, and then use the drop down menus to select the following:
   - The viewpoint to validate
   - The nodes to validate:
     - **All Nodes**: Validates all nodes in the selected viewpoint
     - **Selected Node and Descendants**: Validates the selected node and all of its descendants
     - **Selected Node Only**: Validates only the node currently selected in the viewpoint.
   - The validations to run:
     - **All Validations**: System validations, application and custom validations with a severity of Error or Warning, property validations, and all constraints are run.
     - **Selected Validations**: Use the check boxes to enable or disable these categories of validations:
       - **Validations**: Allows you to select from all enabled application level and custom validations with a severity of Error or Warning.
       - **Properties**: Allows you to select from properties that you have access to in order to run property level validations and any custom validations that use that property as a trigger on them.
       - **Constraints**: Allows you to select from all enabled constraints

       For the options that you selected, click **Select Validations** , and then use the validation selector to pick the specific validations, property level validations, and constraints to run.
3. Click **Validate** You must have selected at least one validation, property level validation, or constraint to run in order to be able to run the validation.
4. **Optional**: If there are validation errors, click **Download Validation Results** to download the validation errors to a file. This enables you to review the errors offline. You can correct the errors in the file and load the corrected file as request items. See Making Changes Using a Load File.

   > **Note:**
   >
   > Validation failures on a maximum of 1000 nodes are displayed on screen. When you download the validation results to a file, the full set of validation failures is downloaded.

Resolving Validation Issues

You can browse validation results, navigate to invalid nodes, and fix issues. As you fix validation issues, the validation result window is automatically updated to show only the remaining issues for the viewpoint. For more information on validations, see Understanding Validations and Constraints.

To resolve validation issues:

1. After you run a viewpoint validation, if there are validation errors to fix, click **Requests**, and then click + to create a new request or open an existing request.
2. Click **Validate** from the left side of the page to display the validation results.

> **Note:**
>
> If a node exists in multiple locations in a hierarchy viewpoint that allows shared nodes, the node may be listed multiple times in the validation results. The node may have the same or different issues for each location.

3. Perform an action:
   - Correct the validations issues on-screen:
     a. Select a node in the validation results to navigate to that node in the viewpoint.

        Validation messages for an invalid node are displayed for a selected node in the validation results. Invalid properties are outlined in red in the property pane. Click on an invalid property to view its validation messages.

     b. Perform request actions, such as update properties, change parent, or delete node in the viewpoint to resolve validation issues. For more information on changing a node's parent, see Changing Parent Nodes.

        When you have resolved all issues for a node, the node is removed from the validation results list.

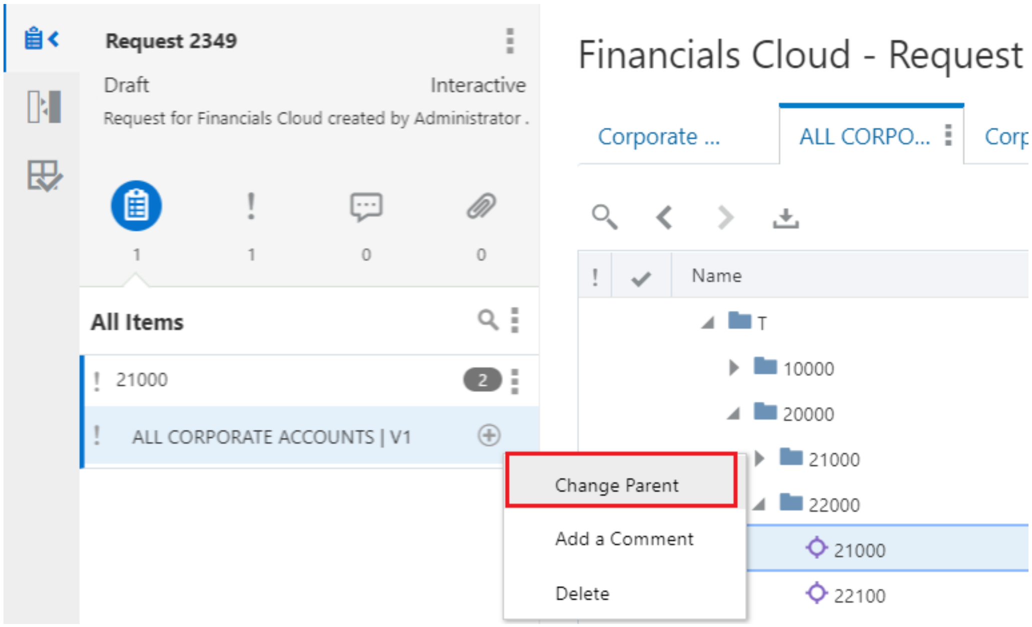
   - Click **Download Validation Results** to download the validation errors to a file so that you can review and correct them offline. The validation errors are sorted by validation severity. You can then load the corrected file as request items. See Making Changes Using a Load File.

> **Note:**
>
> To clear validation results from the validation tab, click the X in the top right corner of the pane.

Changing Parent Nodes

For request items that have a missing or invalid parent, use one of these methods to change the parent.

- From the context menu for the request item instance that has the invalid parent, select **Change Parent**, select a valid parent, and then click **OK**.



- Select the request item with the invalid parent and drag and drop it from the Request Details pane to a parent node in the viewpoint.

7

Comparing Viewpoints

You can systematically compare two viewpoints and locate node differences between them. For example, you can use a compare to identify nodes which need to be inserted into an alternate hierarchy.

Videos

Your Goal	Watch This Video
Learn about comparing viewpoints	Comparing Viewpoints

The ability to compare data from different viewpoints is an integral part of being able to synchronize data across applications.

> **Note:**
>
> Node type converters are needed to compare nodes of *different node types* across two viewpoints in side by side layout.

You can compare viewpoints interactively (see Running Viewpoint Comparisons), or for comparisons that you run frequently you can create a Compare Profile (see Creating, Editing, and Deleting Compare Profiles) to save the parameters in a reusable profile that can be run by users with access to both viewpoints.

Compare profiles also enable you to filter the nodes being compared by defining a node expression.

For more information see:

- Running Viewpoint Comparisons
- Creating, Editing, and Deleting Compare Profiles
- Correcting Node Differences From Compare Results
- Working with Node Type Converters

Running Viewpoint Comparisons

You can compare two viewpoints to determine the differences between them. You can also compare two different hierarchies in the same viewpoint by opening a duplicate tab for a viewpoint.

You can compare viewpoints to:

- Find Missing Nodes: Finds nodes that exist in one viewpoint that do not exist in the other viewpoint.
- Compare Relationships: Finds nodes that exist in both viewpoints but that have different parents.

> **Note:**
>
> Compare Relationships identifies nodes that have different parents and does not distinguish between a missing relationship and a different relationship, particularly when a node has multiple parents in each viewpoint. When downloading the compare results, the **Different Relationships** tab in the downloaded file always contains a parent of the node in the target viewpoint.

- Compare Property Values: Finds nodes that exist in both viewpoints but that have differences in one or more property values.

> **Note:**
>
> When running compares for properties with the String or Boolean data type, both the displayed value and the actual value for properties are evaluated for differences. For example, if the displayed value is "True" for one Boolean property and "Yes" for a second Boolean property, those properties are not identified as having different property values.

> **Tip:**
>
> You can specify the properties that you want to display as columns to make the comparison easier. See Configuring How a Viewpoint Displays Properties.

You can run multiple types of comparisons at the same time. For example, you can run a comparison for missing nodes and nodes that have different parents.

> **Tip:**
>
> For comparisons that are run repeatedly, you can create compare profiles that enable you to save a set of comparison parameters in a reusable profile. Then, instead of manually specifying the parameters each time you want to run that compare, you select the predefined profile from the drop down menu. See Creating, Editing, and Deleting Compare Profiles.

If the Alternate Name property is available on the node types being compared, both the node names and the alternate names are evaluated during the comparison. A match on the name property takes precedence over a match on the alternate name.

If the nodes being compared have node links (see About Node Links), those links are also evaluated after name and alternate name during the comparison. Linked nodes are treated as the same node when running compares. For example, if you run a comparison to look for missing nodes and a node in your source has a link to a node in the target, that node will not be displayed as missing in the target.

For hierarchy viewpoints, you can compare the entire hierarchy, or you can select a top node to compare a portion of the hierarchy. You can also compare only the bottom nodes of the entire hierarchy or of a selected top node.

As you correct differences between the viewpoints, the compare results are automatically updated.

> **Note:**
>
> If you run a comparison of viewpoints that use different node types, you'll get a message indicating that you need to set up node type converters. For more information, see Working with Node Type Converters.

Comparing Two Different Viewpoints

1. From the left pane, click **Compare** to open the **Compare** tab.
2. Perform an action:
   - To select a compare profile:
     a. Click the drop down menu under the **Compare Viewpoints** header.
        The options to interactively compare **Left to Right** and **Right to Left** are displayed first, followed by your predefined **Compare Profiles**. Compare profiles are displayed only when both viewpoints are active. Profiles that include any archived viewpoints are not displayed.
     b. Select a compare profile to run.
        The viewpoints in the profile are selected automatically. All other compare parameters are disabled.
     c. **Optional**: After you select a compare profile to run, click **Compare Actions** ⋮ and then select **Inspect** to inspect or edit the compare profile parameters.

     For information on creating compare profiles, see Creating, Editing, and Deleting Compare Profiles.
   - To build an interactive viewpoint comparison:
     a. Select the viewpoints to compare.
     b. Select a direction for the compare.

        For example, for missing nodes:
        – If you select Left to Right, the compare determines which nodes are in the left viewpoint but not in the right viewpoint.
        – If you select Right to Left, the compare determines which nodes are in the right viewpoint but not in the left viewpoint.
     c. Select the nodes that you want to compare (select one):
        – **All Nodes**: Compares all of the nodes in the viewpoint.
        – **Bottom Nodes**: Compares all of the bottom nodes in the viewpoint.
        – **Selected Node and Descendants**: Compares the node that you selected and all of the nodes underneath it. (Hierarchy viewpoints only)
        – **Bottom Nodes for Selected Node**: Compares the bottom nodes underneath the node that you selected. (Hierarchy viewpoints only)
     d. Click the type of comparison that you want to run (select one or more):
        – **Find Missing Nodes**
        – **Compare Relationships** (Hierarchy viewpoints only)
        – **Compare Property Values**

e. **Optional**: When comparing property values, click **Select Properties to Compare** ☑ and then use the property selector to select specific properties for comparison. Click a property on the right to select it. Click **X** to remove a selected property.

3. Click **Run Compare** ▷ to run the compare for the selected viewpoints or compare profile.

   The nodes in the compare results are displayed in the **Compare** pane. For hierarchy viewpoints, the parent name is also displayed. If the compare results exceed the maximum that are able to be displayed on the screen, you can download the results to see the full set.

   Click a node to see it in the viewpoint where it exists. See Aligning Nodes.

   > **Note:**
   >
   > For property comparisons, **Align: Left** is selected automatically, and when you click on a node in the compare results, the node is located in both viewpoints.
   >
   > For relationship comparisons, you can click ⇄ on either viewpoint and then click a node in the compare results to locate the node in both viewpoints.

   You can create a request to resolve node differences between the viewpoints, see Correcting Node Differences From Compare Results. As you resolve each difference, the nodes are removed from the compare results.

   > **Note:**
   >
   > Each shared node under a different parent is listed separately in the compare pane.

4. **Optional:** Change the comparison criteria and then click **Compare Refresh** ⟳ to run a new comparison.

5. **Optional:** Download the compare results to an Excel file by clicking **Action** ⋮ in the Compare Viewpoints header and selecting **Download to file**. The downloaded file contains the full compare results, not just the results that are displayed on the screen, and the viewpoints are displayed on separate tabs in the downloaded file.

   > **Note:**
   >
   > When comparing relationships, the **Different Relationships** column in the downloaded file always contains the parent of the node in the source viewpoint, even if the node doesn't have a parent in the target viewpoint.

Comparing Different Hierarchies in the Same Viewpoint

1. Open a duplicate tab of the viewpoint that contains the hierarchies that you want to compare. See Using Duplicate Viewpoint Tabs.

2. From the left pane, click **Compare** to open the **Compare** tab.
3. Perform an action:
   - To select a compare profile:
     a. Click the drop down menu under the **Compare Viewpoints** header.
     The options to interactively compare **Left to Right** and **Right to Left** are displayed first, followed by your predefined **Compare Profiles**. Compare profiles are displayed only when both viewpoints are active. Profiles that include any archived viewpoints are not displayed.
     b. Select a compare profile to run.
     A duplicate tab of the viewpoint is added to the view (if it does not already exist) and is displayed in the side by side layout. All other compare parameters are disabled.
     c. **Optional**: After you select a compare profile to run, click **Compare Actions** ⋮ and then select **Inspect** to inspect or edit the compare profile parameters.

     For information on creating compare profiles, see Creating, Editing, and Deleting Compare Profiles.
   - To build an interactive viewpoint comparison:
     a. Select the original and duplicate tabs of the viewpoint to compare.
     b. Select a direction for the compare.
     c. Select the nodes that you want to compare (select one):
        – **Selected Node and Descendants**: Compares the node that you selected and all of the nodes underneath it.
        – **Bottom Nodes for Selected Node**: Compares the bottom nodes underneath the node that you selected.

     > **Note:**
     >
     > Because the original and duplicate tabs display the same viewpoint, the **All Nodes** and **Bottom Nodes** options will not generate any compare differences.

     d. Click the type of comparison that you want to run (select one or more):
        – **Find Missing Nodes**
        – **Compare Relationships**
        – **Compare Property Values**
     e. **Optional**: When comparing property values, click **Select Properties to Compare** and then use the property selector to select specific properties for comparison. Click a property on the right to select it. Click **X** to remove a selected property.
4. **Optional**: When comparing property values, click **Select Properties to Compare** and then use the property selector to select specific properties for comparison. Click a property on the right to select it. Click **X** to remove a selected property.

5. Click **Run Compare** ▷ to run the compare for the selected viewpoints.

Working with Compare Results

After you run the compare, the nodes in the compare results are displayed in the **Compare** pane.

> **Note:**
>
> Each shared node under a different parent is listed separately in the compare pane.

You can perform the following actions on your compare results:

- Click a node to see it in the viewpoint where it exists. See Aligning Nodes.

  > **Note:**
  >
  > For relationship and property comparisons, the direction (Align Left or Right) is selected automatically based on the direction of the compare, and when you click on a node in the compare results the node is located in both viewpoints under the selected node that is being compared. For relationship comparisons, you can click ⇄ on either viewpoint and then click a node in the compare results to locate the node in both viewpoints under the selected node that is being compared.

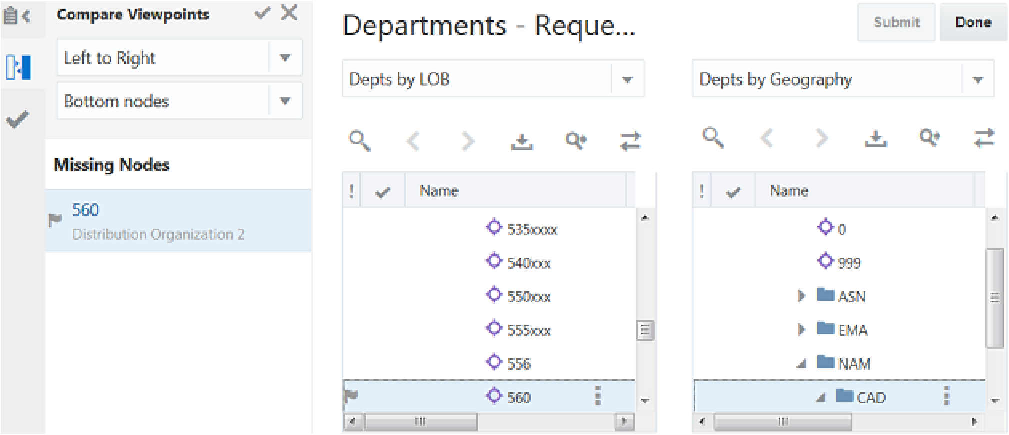
- Create a request to resolve node differences between the viewpoints, see Correcting Node Differences From Compare Results. As you resolve each difference, the nodes are removed from the compare results.

Viewpoint Comparison Examples

This topic contains examples of different types of viewpoint comparisons and how you can use them.

Missing Nodes Example

In this example, we compared the bottom nodes in the Depts by LOB viewpoint to the Depts by Geography viewpoint. The results show that there is one node in the Depts by LOB viewpoint that is not in the Depts by Geography viewpoint.



Compare Viewpoints

Left to Right

Bottom nodes

Missing Nodes

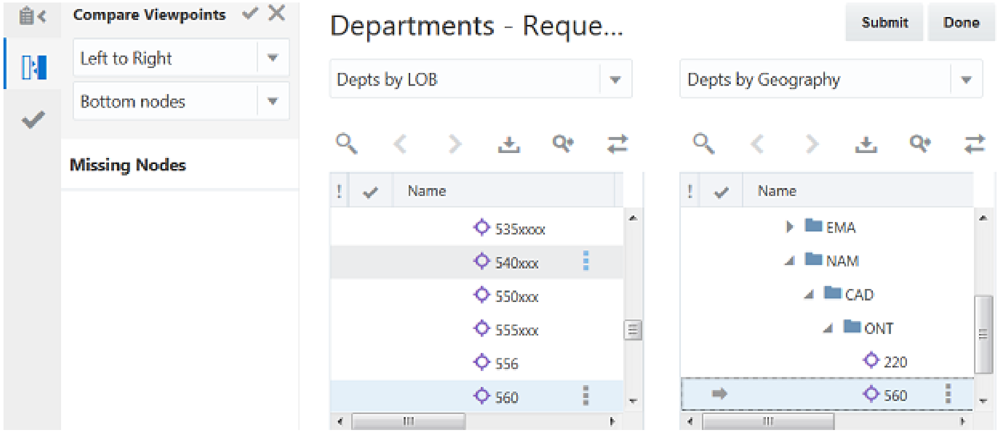
560
Distribution Organization 2

Departments - Reque...

Depts by LOB: 535xxxx, 540xxx, 550xxx, 555xxx, 556, 560

Depts by Geography: 0, 999, ASN, EMA, NAM, CAD

To resolve the difference, we created a request and then dragged node 560 from the LOB viewpoint to the Geography viewpoint. As soon as the node is added to the Geography viewpoint, node 560 is removed from the compare results. We submitted the request to commit the change to the Geography viewpoint.



Compare Viewpoints

Left to Right

Bottom nodes

Missing Nodes

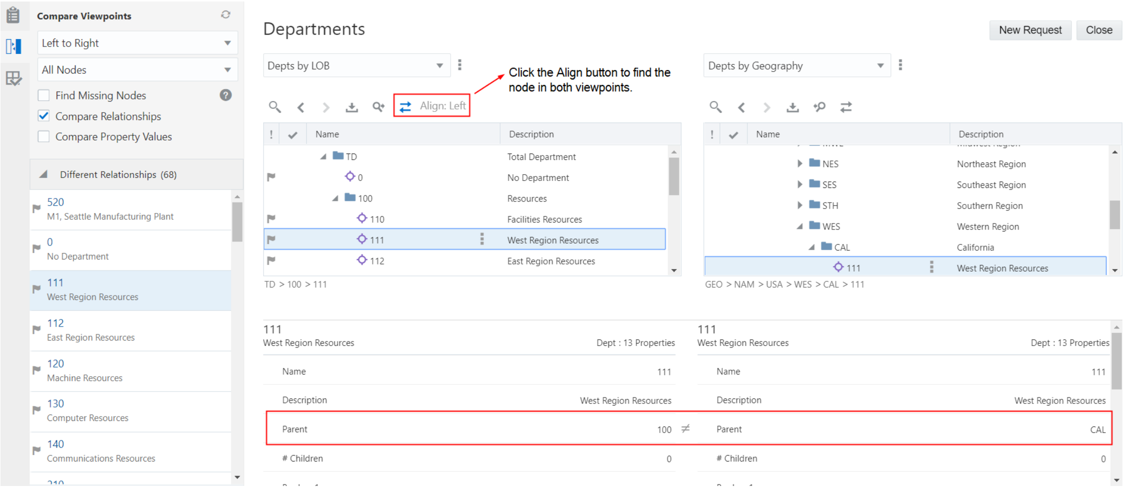
Departments - Reque...

Depts by LOB: 535xxxx, 540xxx, 550xxx, 555xxx, 556, 560

Depts by Geography: EMA, NAM, CAD, ONT, 220, 560

Compare Relationship Example

In this example, we compared the relationships of all nodes in the Depts by LOB and the Depts by Geography viewpoints. All of the nodes that exist in both viewpoints but that have different parents are displayed. Because nodes are being aligned based on the left-hand viewpoint, when we click node **111** in the compare results, the node is displayed in both viewpoints.



Compare Viewpoints

Left to Right

All Nodes

□ Find Missing Nodes

☑ Compare Relationships

□ Compare Property Values

Different Relationships (68)

520
M1, Seattle Manufacturing Plant

0
No Department

111
West Region Resources

112
East Region Resources

120
Machine Resources

130
Computer Resources

140
Communications Resources

Departments

Depts by LOB

Click the Align button to find the node in both viewpoints.

Name	Description
TD	Total Department
0	No Department
100	Resources
110	Facilities Resources
111	West Region Resources
112	East Region Resources

TD > 100 > 111

Depts by Geography

Name	Description
NES	Northeast Region
SES	Southeast Region
STH	Southern Region
WES	Western Region
CAL	California
111	West Region Resources

GEO > NAM > USA > WES > CAL > 111

111
West Region Resources — Dept : 13 Properties

Name	111
Description	West Region Resources
Parent	100 ≠
# Children	0

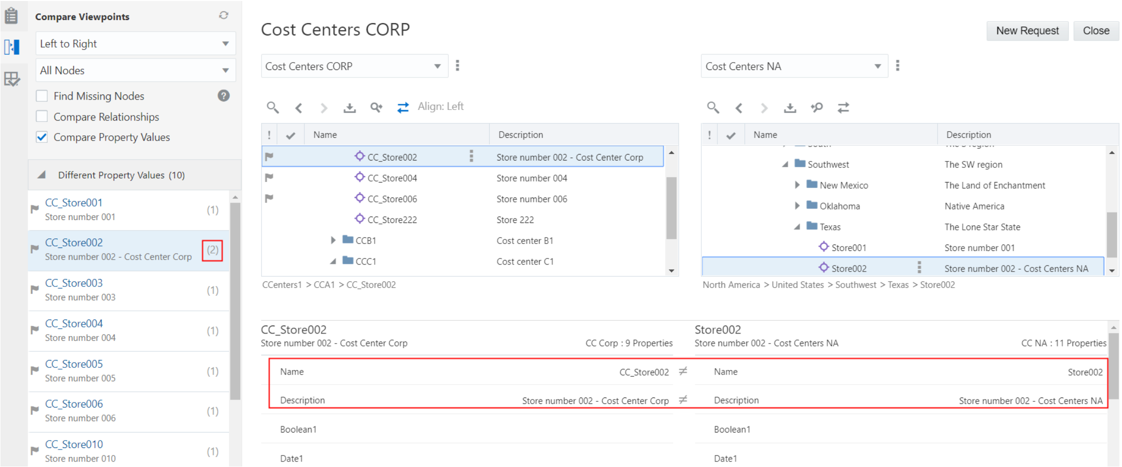
111
West Region Resources — Dept : 13 Properties

Name	111
Description	West Region Resources
Parent	CAL
# Children	0

The properties panel shows the different values for the node parent.

Compare Properties Example

In this example, we compared the properties of all nodes in the Cost Centers CORP and Cost Centers NA viewpoints. The compare results showed two property differences for the Store002 node. We clicked CC_Store002 in the compare results. Because **Align:Left** was selected automatically, the node is displayed in both viewpoints and the properties panel showed differences in the name and the description properties for this node.



Left to Right

All Nodes

□ Find Missing Nodes

□ Compare Relationships

☑ Compare Property Values

Different Property Values (10)

Node	Description	Differences
CC_Store001	Store number 001	(1)
CC_Store002	Store number 002 - Cost Center Corp	(2)
CC_Store003	Store number 003	(1)
CC_Store004	Store number 004	(1)
CC_Store005	Store number 005	(1)
CC_Store006	Store number 006	(1)
CC_Store010	Store number 010	(1)

Cost Centers CORP

Cost Centers CORP

Name	Description
CC_Store002	Store number 002 - Cost Center Corp
CC_Store004	Store number 004
CC_Store006	Store number 006
CC_Store222	Store 222
CCB1	Cost center B1
CCC1	Cost center C1

CCenters1 > CCA1 > CC_Store002

Cost Centers NA

Name	Description
Southwest	The SW region
New Mexico	The Land of Enchantment
Oklahoma	Native America
Texas	The Lone Star State
Store001	Store number 001
Store002	Store number 002 - Cost Centers NA

North America > United States > Southwest > Texas > Store002

CC_Store002
Store number 002 - Cost Center Corp — CC Corp : 9 Properties

Property	Value
Name	CC_Store002 ≠
Description	Store number 002 - Cost Center Corp ≠
Boolean1	
Date1	

Store002
Store number 002 - Cost Centers NA — CC NA : 11 Properties

Property	Value
Name	Store002
Description	Store number 002 - Cost Centers NA
Boolean1	
Date1	

Correcting Node Differences From Compare Results

Using the results from running a compare, you can correct differences and rationalize changes between two viewpoints. You can create or open a draft request and then make your changes or you can generate a new request based on the differences in the comparison.

Correcting Missing Nodes

For missing nodes, you can drag and drop nodes as a method of inserting them into a viewpoint. To drag and drop nodes from a source to a target, you need at least *Participant(Read)* permission on the source data object and at least *Participant(Write)* permission on the target data object.

To correct a missing node difference:

1. Open two viewpoints in a side by side layout.
2. Start a new request or open an existing request.
3. In one of the viewpoints, select a node whose node type allows child nodes.
4. Drag the node from that viewpoint to the other viewpoint.

> **Note:**
>
> The parent node where you are dropping the node must allow child nodes of this node type. If the viewpoints use different node types, node type converters must be set up for the drag and drop to be successful.

- If a node with the same name does not exist in the target node type, the new node is added to the viewpoint under the parent where it was dropped.
- If a node with the same name does exist in the target node type and the hierarchy set being used allows shared nodes, then the existing node is inserted to the viewpoint under the parent where it was dropped.

Correcting Relationship and Property Differences

Use the Align feature to view existing nodes in both viewpoints with relationship or property differences, see Aligning Nodes.

> **Note:**
>
> The direction (Align Left or Right) is selected automatically based on the direction of the compare when you compare properties or relationships. When you click on a node in the compare results, the node is located in both viewpoints under the selected node that is being compared.

- Relationship differences can be resolved by moving the node in either viewpoint or inserting a shared node (if allowed).
- Property differences can be resolved by changing the value in either viewpoint or copying a value across viewpoints.

Creating Request Items from Comparison Results

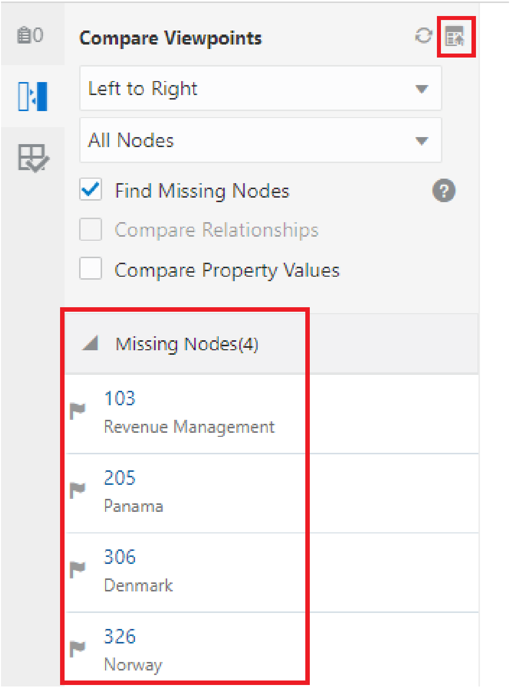
Instead of making changes manually to correct differences between the viewpoints, you can create a request directly from the comparison results. After running a compare, you can create a request, generate request items and actions from the compare results, and then submit the request to commit the changes.

> **Note:**
>
> For nodes that are added or inserted where the parent is not known (for example, when the source viewpoint is a list and the target is a hierarchy), an administrator can define an expression to calculate the parent value for each request action and store it for the node in the viewpoint. See Calculated and Stored Properties.

> **Caution:**
>
> When comparing different hierarchies in the same viewpoint (see Using Duplicate Viewpoint Tabs), request items to insert nodes into the target viewpoint may not get created. Since the source hierarchy also exists in the target viewpoint, the generated request file rows to insert the nodes may be skipped because the nodes are already located under the parents in the source hierarchy.
>
> To resolve this situation, you can download the request file attachment, change the Parent column to identify the proper parent in the target hierarchy, and load the modified file back into the request. Alternatively, you can manually insert the nodes into the target hierarchy (see *Correcting Missing Nodes* above).

To create request items from comparison results:

1. After running the comparison, click **New Request** or open a draft request.
2. Click **Create Request Items from Compare** . See example below.



Left to Right

All Nodes

☑ Find Missing Nodes

□ Compare Relationships

□ Compare Property Values

Missing Nodes(4)

103	Revenue Management
205	Panama
306	Denmark
326	Norway

> **Note:**
>
> If you have filtered the viewpoint comparison by selecting specific nodes (bottom nodes only, for example) or specific properties, request items are created for the filtered compare results only.

3. When all changes have been added to the request, click **Submit** to commit the changes or **Done** to close the request without submitting.

8 Subscribing to Viewpoints

Subscriptions enable you to share data between viewpoints by subscribing a target viewpoint to a source viewpoint. When an update is made to the source viewpoint, a request is automatically generated to make the same change in the target viewpoint.

For example, you can set up viewpoints in your Planning and Financial Consolidation and Close applications to subscribe to a viewpoint in your Oracle Financials Cloud General Ledger application. When you add a node in the General Ledger application (such as adding a new cost center), the system automatically generates requests to add that node to the Planning and Financial Consolidation and Close applications.

Videos

Your Goal	Watch This Video
Learn about subscribing to viewpoints.	▶ Overview: Synchronizing Data Changes

Benefits

Automating data synchronization across different viewpoints using subscriptions offers several benefits, including:

- **Data Consistency**: By adding nodes in target viewpoints automatically when they are added to source viewpoints, you maintain data consistency across multiple applications.
- **Flexible Configuration Options**: You can subscribe to an entire viewpoint, or you can configure your subscription to synchronize only specific actions, top nodes, or node conditions. This lets you tailor your subscriptions to synchronize the data in the source viewpoint on a more granular level. See Adding Filters and Conditions to Subscriptions.
- **Streamline Request Processing**: The system can combine target requests when certain conditions are met (see Combining Target Requests). This streamlines the request process by reducing the number of requests that need to be processed to synchronize data.
- **Faster Adaptation to Changes**: You can configure subscriptions to submit changes from the source viewpoint in the target viewpoint automatically and to bypass the approval process if all conditions are met. This speeds up the time it takes for changes in source viewpoints to be propagated to target viewpoints.

Considerations

- When you create a subscription, you specify whether the system should attempt to submit the request in the target viewpoints automatically or if the subscription assignee should submit it manually:
  - **Auto-submit**—enables you to synchronize changes instantly between viewpoints. The system will validate the subscription request and automatically submit the changes to the target viewpoints.

> **Note:**
>
> If any of the subscriptions used to generate the request do not have **Auto-submit** enabled, then the request is not auto-submitted. Instead, the subscription assignee is notified so that they can take corrective action.
> If items in a subscription request are skipped with either a "No updates required" or "Row with the same key was already processed" status and all other auto-submit conditions are met, the request is submitted. If items are skipped with any other status (even if other items in the same request are skipped with one of the two statuses above), the entire subscription request is not submitted.

  - **Manual submit**—enables the subscription assignee to review and edit the target request before submitting the changes.

- For auto-submitted subscription requests, select **Bypass Approvals** to skip the Approve stage and move the request to the Commit stage automatically if all of the request items are valid.

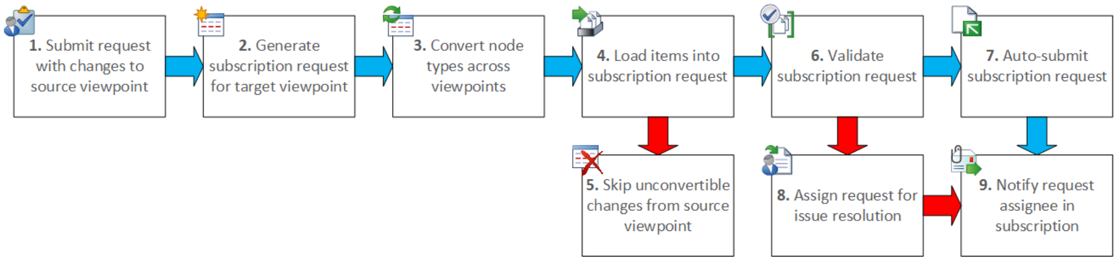
  For subscription requests that contain items from multiple subscriptions:

  - If all subscriptions in the request have Bypass Approvals enabled, all request items in the subscription are moved to the Commit stage.
  - If any of the subscriptions in a combined request (see Combining Target Requests) do not have Bypass Approvals enabled then all of the request items in that request must be approved, even if some of the request items were generated from subscriptions that had Bypass Approvals enabled.

- With subscriptions, the context in the source is relevant to the context in the target. For example, if you add a child node in the source, the node will be added to the same parent in the target when possible.
- When you archive a viewpoint or a dimension, subscriptions that use that viewpoint or dimension are automatically ignored and will not generate subscription requests. When you unarchive a viewpoint or a dimension, subscriptions are no longer ignored and are returned to their previous state (Enabled or Disabled).
- You can run a report to see a complete list of subscriptions that have been defined across all views and applications. For more information, see Subscriptions Reports.

Subscription Process Overview

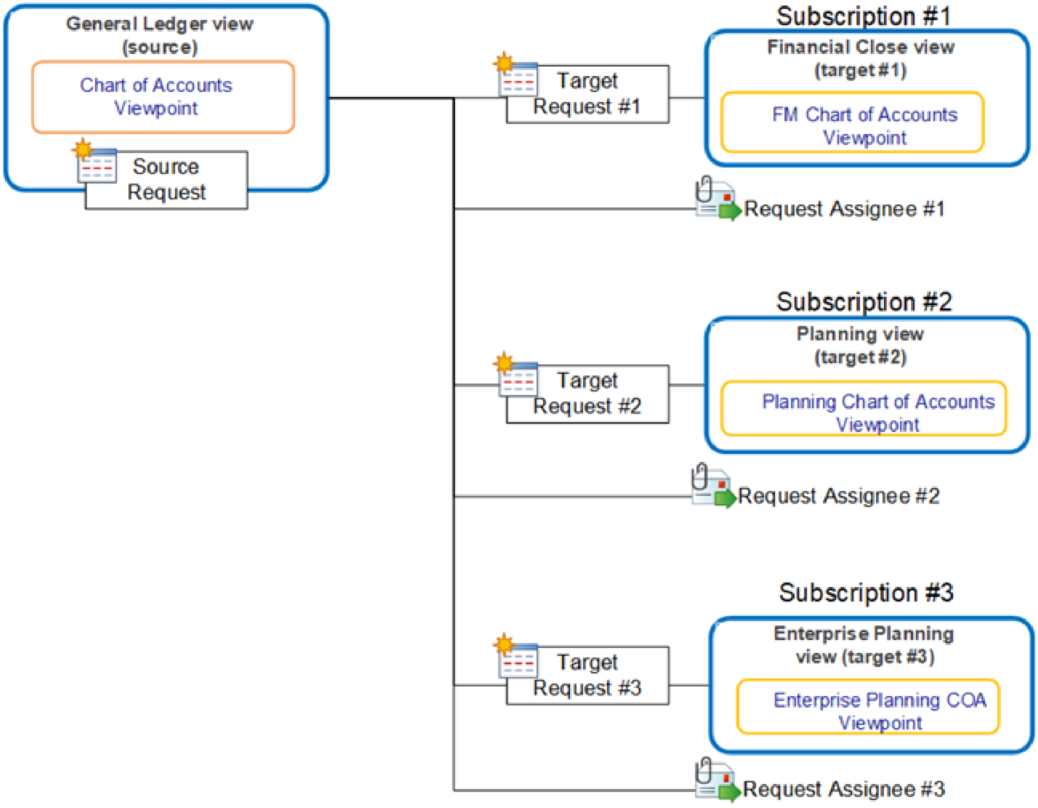
You create a subscription by subscribing from a target viewpoint to a source viewpoint. After the subscription is created, the subscription process follows the flow in the following image. Note that this flow represents an overview of the subscription process and not a series of steps to be performed by a user.


1. Submit request with changes to source viewpoint
2. Generate subscription request for target viewpoint
3. Convert node types across viewpoints
4. Load items into subscription request
6. Validate subscription request
7. Auto-submit subscription request
5. Skip unconvertible changes from source viewpoint
8. Assign request for issue resolution
9. Notify request assignee in subscription

1. The subscription process begins when a user submits a request with changes in a source viewpoint.
2. The system generates individual or combined target requests for all target viewpoints. See Combining Target Requests.
3. The system converts the source node types to the target node types using node type converters. See Subscription Prerequisites.
4. If the node types are identical or if the item can be converted by using a node type converter, the item is loaded to the target subscription request.
5. If the item cannot be converted (for example, if a node type converter for that node type doesn't exist), the item is skipped.
6. The system validates all of the items that were loaded to the subscription target request.
7. If the items in the target subscription request are valid, no items are skipped, and auto-submit is enabled, the system submits the items in the target subscription request and the request status is changed to **Completed**.
8. If the items in the target subscription request are not valid (for example, if the parent can't be identified), or if auto-submit is not enabled, the request remains in **Draft** status and the assignee must take action on it.
9. The system sends notifications to the request assignee with the status of the request. See Request Notifications.

Subscription Example

The example below shows a Chart of Accounts viewpoint in a General Ledger view as the subscription source, with target viewpoints in Financial Close, Planning, and Enterprise Planning views subscribed to it. Each subscription has a separate assignee. When an update is made to the Chart of Accounts viewpoint, target subscription requests are automatically generated to make the same update in the three subscribing viewpoints, and notifications are sent to each assignee.


General Ledger view (source)
Chart of Accounts Viewpoint
Source Request
Target Request #1
Subscription #1
Financial Close view (target #1)
FM Chart of Accounts Viewpoint
Request Assignee #1
Target Request #2
Subscription #2
Planning view (target #2)
Planning Chart of Accounts Viewpoint
Request Assignee #2
Target Request #3
Subscription #3
Enterprise Planning view (target #3)
Enterprise Planning COA Viewpoint
Request Assignee #3

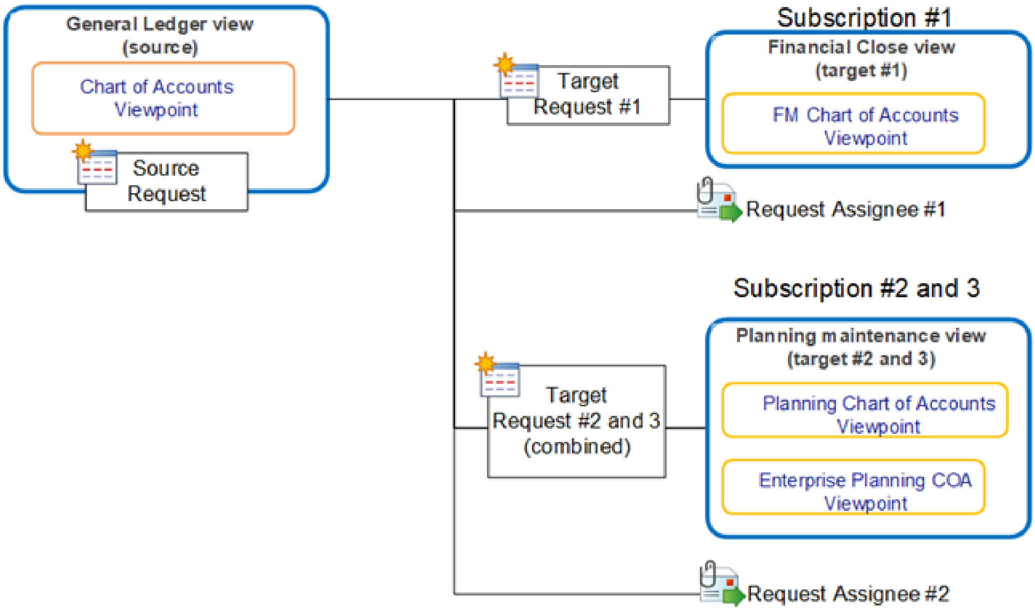
Combining Target Requests

Request items are combined in target requests if these conditions are met:

- The source viewpoints exist in the same view.
- The target viewpoints exist in the same view. They do not need to be in the same view as the source viewpoints.
- The subscriptions have the same assignee.

> **Tip:**
>
> If you want to create a separate request for this subscription without combining it with other requests with the same view and assignee, select **Create Standalone Request** in the subscription definition. See Creating, Editing, and Validating Subscriptions.

In the following example, the Planning and Enterprise Planning viewpoints were added to a Planning Maintenance view, and both subscriptions have the same assignee. Because two of the target viewpoints are in the same view and have the same assignee, the target requests for the Planning and Enterprise Planning updates are combined into a single request.


General Ledger view (source)
Chart of Accounts Viewpoint
Source Request
Target Request #1
Subscription #1
Financial Close view (target #1)
FM Chart of Accounts Viewpoint
Request Assignee #1
Subscription #2 and 3
Planning maintenance view (target #2 and 3)
Planning Chart of Accounts Viewpoint
Enterprise Planning COA Viewpoint
Target Request #2 and 3 (combined)
Request Assignee #2

Creating, Editing, and Validating Subscriptions

You create a subscription by navigating to the target viewpoint and selecting the source viewpoint to subscribe to. When a change is committed in the source viewpoint, requests are generated to perform the same change in the target viewpoints.

Best Practice

When you create a node type converter, it is a best practice to convert a source node type to a single node type in the target viewpoint. If you convert a source node type to multiple target node types, the subscription request could become ambiguous and the record may be skipped.

> **Tip:**
>
> After you create a subscription, use the Lineage tab in the viewpoint inspector to visualize the subscriptions that your viewpoint is the source and target of. See Viewing Viewpoint Lineage.

Subscription Prerequisites

These conditions must be met in order to be able to create a subscription:

- The source and target viewpoints both must have an Active status.
- The subscription creator and the default and alternate assignees must have sufficient permissions and data access. See Security for Requests.
- If the source and target viewpoints both have a hierarchy type node set, the two hierarchy sets must be different, and the two viewpoints must have at least one equivalent node type between them. That is, the source and the target viewpoints must have the same node type in the same application, or else a node type converter must be set up to convert the source node type to the target node type.
- If the source viewpoint has a list type node set and the target viewpoint has a hierarchy type node set, the two viewpoints must have at least one equivalent node type between

them. That is, the source and the target viewpoints must have the same node type in the same application, or else a node type converter must be set up to convert the source node type to the target node type.

- If the source and target viewpoints both have a list type node set or if the source has a hierarchy type node set and the target has a list type node set, the two viewpoints must have at least one different node type and a node type converter must be set up to convert the source node type to the target node type.

Considerations

- For subscription actions that add or insert nodes where the source viewpoint has a list type node set and the target viewpoint has a hierarchy type node set, the value of the parent in the hierarchy viewpoint can be calculated for each request action. An administrator must define an expression to calculate the parent value. See Calculated and Stored Properties.
- For subscriptions with calculated node names, the node names are calculated in the source viewpoint and the resulting names are passed to the target viewpoints. If you want to calculate the names in the target viewpoint, you must set up a property transformation in the node type converter to transform the name property into a blank value or one of the calculated name keywords. See Calculated Names and Subscriptions.
- Parent transformations in node type converters are run for top nodes in source hierarchy viewpoints when request items for a subscription request are generated. This enables top nodes which were added in a source viewpoint to be placed under a different parent in a target hierarchy viewpoint.
- For target nodes that use name transformations, if the target node is linked to the source node, the target nodes can be Removed or Deleted from the target viewpoints by using the node link information to identify the target nodes.

> **Note:**
>
> Node links are established when nodes are synchronized using subscriptions, when creating request items from a compare, or when nodes are matched and merged (see About Node Links). If there are nodes in the target viewpoint with name transformations that are not linked to source nodes (for example, they were not created or updated from the source viewpoint with a subscription), then they cannot be Removed or Deleted from a target viewpoint using a subscription.

- If your target viewpoint name has more than 30 characters, you must create a label that has 30 characters or less. The label is used instead of the viewpoint name in the subscription request file. See Inspecting a Viewpoint.
- You cannot create subscriptions with time labeled viewpoints as either the source or target viewpoint.

Creating a Subscription

A target viewpoint can subscribe to multiple source viewpoints. To create a subscription:

1. From **Views**, open the view that contains the target viewpoint.
2. Select the target viewpoint.
3. Place your cursor to the right of the viewpoint name, click ⋮, and then select **Inspect**.
4. Select the **Subscriptions** tab, and then click **Create**.

5. In the Create Subscriptions dialog box, perform these actions:
   1. Select the source viewpoint that you want to subscribe to.

      > **Note:**
      >
      > Only the viewpoints that match the conditions in Subscription Prerequisites are displayed for selection.

   2. **Optional**: Enter a name and description for the subscription. By default, the subscription name is "{Source View Name} | {Source Viewpoint Name}".
   3. Click **Create**.

After you click **Create**, the subscription is displayed in the Subscription Inspector. From here, you can edit the subscription settings.

Editing, Enabling, Disabling, or Deleting Subscriptions

To edit a subscription:

1. From **Views**, open the view that contains the target viewpoint.
2. Select the target viewpoint.
3. Place your cursor to the right of the viewpoint name, click ⋮, and then select **Inspect**.
4. Select **Subscriptions**, and then click the name of the subscription that you want to edit.

   The subscription is displayed in the subscription inspector.
5. **Optional**: On the General tab, click **Edit**, and then modify the subscription name or description.
6. On the Definition tab, click **Edit**, and then perform an action:
   - In **Assignees**, select the default and alternate request assignees. See Assigning Subscription Requests.

     > **Note:**
     >
     > Only users that have at least *Participant(Write)* permission on the dimension in the target viewpoint are available to be selected. The ✉ icon indicates that the user has an email address configured and is able to receive request notifications.

   - **Optional**: In **Workflow**, select one or more of these options:

Field	Description
Enabled	Enables the subscription. When you create a subscription, it is disabled by default. You must have defined a default or alternate assignee in order to enable a subscription. You can also enable and disable subscriptions from the Subscriptions tab of the viewpoint inspector. See *To enable, disable, or delete a subscription*, below. **Note:** When a subscription is disabled, no subscription requests or request items are generated for the target viewpoint.
Create Standalone Request	Specifies that this subscription always creates a separate request. It will not be combined with other requests that have the same view and assignee.
Collaborate	Includes alternate assignees who are not assigned the request as collaborators on the request. This enables the alternate assignees to make changes to request items and add comments or attachments even when the request is not assigned to them. See Collaborating on a Request.
Auto-Submit	Enables the system to automatically submit the subscription if it passes validation and meets either of the following conditions: – No items are skipped – Items are skipped with either a "No updates required" or "Row with the same key was already processed" status. If items are skipped with any other status (even if other items in the same request are skipped with one of the two statuses above), the entire subscription request is not submitted.
Bypass Approvals	Enables the system to bypass approvals for auto-submitted requests **Note:** This option is available only when **Auto-Submit** is selected.

7. **Optional**: In **Copy Nodes** use the **Copy Action** and, optionally, the **Properties to Match** fields on the Definition tab of the subscription inspector to specify a copy or model after operation (see Performing Copy and Model After Operations in Subscriptions):

8. **Optional**: On the Filters tab, for subscriptions in hierarchy viewpoints only you can add top node filters if you don't want to subscribe to the entire viewpoint. Top node filters enable you to select the branches in the source viewpoint that you want to subscribe to. See Adding Filters and Conditions to Subscriptions

9. Click **Save**.

> **Note:**
>
> You cannot edit a subscription to change the source viewpoint to subscribe to. To change the source viewpoint, delete the existing subscription and create a new subscription with the desired source viewpoint.

To enable, disable, or delete a subscription:

1. Inspect the viewpoint that contains the subscription that you want to enable, disable, or delete. See Inspecting a Viewpoint.
2. On the Subscriptions tab, click **Edit**.
3. In the **Action** column of the subscription that you want to take action on, click ⋮, and then select an option:
   - **Enable** (available only if the subscription is disabled)
   - **Disable** (available only if the subscription is enabled)
   - **Delete**
4. Click **Yes** to confirm, and then click **Save**.

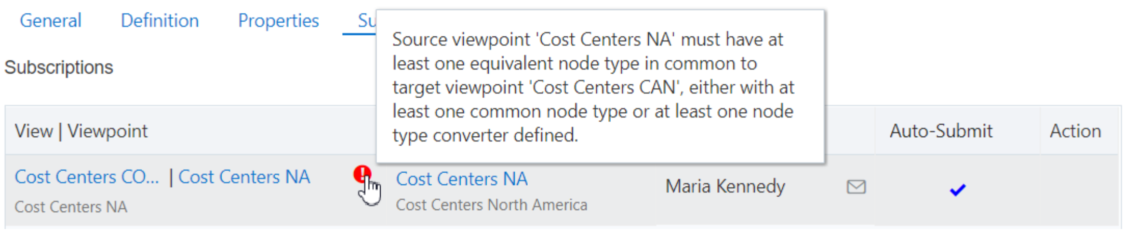
Validating Subscriptions

When you create a subscription, the system ensures that it conforms to the prerequisites listed above. After a subscription is created, however, the configuration of the viewpoints and the users may change, which could result in an invalid subscription. When you navigate to the subscription panel, all existing subscriptions for a viewpoint are validated to verify that they meet the rules of a subscription. If a subscription is invalid, you can take action to resolve the issue.

Resolving Validation Errors

The examples below illustrate common subscription validation errors and the actions to resolve them.

If the source and target viewpoints do not have a node type in common or if a node type converter is not set up between the source and the target viewpoints, the system displays the following error:

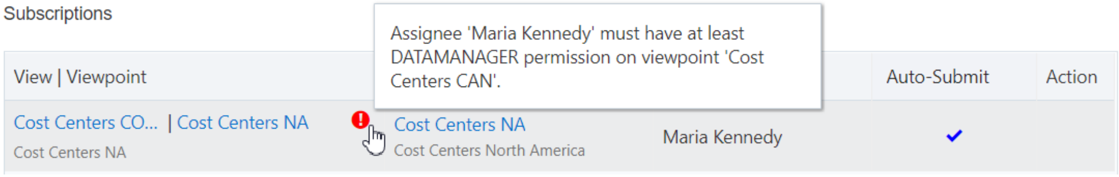


Source viewpoint 'Cost Centers NA' must have at least one equivalent node type in common to target viewpoint 'Cost Centers CAN', either with at least one common node type or at least one node type converter defined.

View | Viewpoint				Auto-Submit	Action
Cost Centers CO... | Cost Centers NA Cost Centers NA	Cost Centers NA Cost Centers North America	Maria Kennedy	✉	✓	

To resolve this, create a node type converter to convert the source node type to the target node type, see Working with Node Type Converters.

If the subscription assignee does not have at least *Participant(Read)* permission to the source viewpoint and *Participant(Write)* permission to the target viewpoint, the system displays the following error:



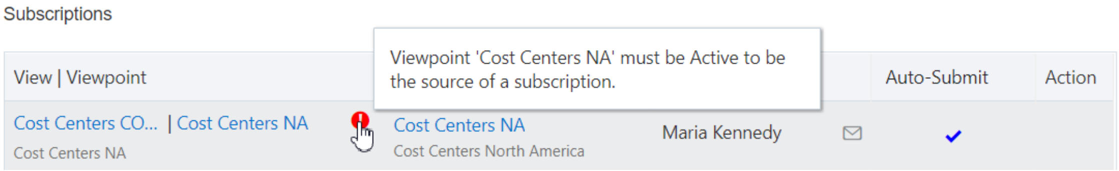
Subscriptions

Assignee 'Maria Kennedy' must have at least DATAMANAGER permission on viewpoint 'Cost Centers CAN'.

View | Viewpoint				Auto-Submit	Action
Cost Centers CO... | Cost Centers NA Cost Centers NA	❗	Cost Centers NA Cost Centers North America	Maria Kennedy	✓	

To resolve this, ensure that the subscription assignee has at least *Participant(Read)* permission to the source viewpoint and *Participant(Write)* permission to the applications in the target viewpoint, see Assigning Application Permissions and Assigning View Permissions.

If the source or target viewpoint is no longer active, the system displays the following error:



Subscriptions

Viewpoint 'Cost Centers NA' must be Active to be the source of a subscription.

View | Viewpoint					Auto-Submit	Action
Cost Centers CO... | Cost Centers NA Cost Centers NA	❗	Cost Centers NA Cost Centers North America	Maria Kennedy	✉	✓	

To resolve inactive viewpoint errors, change the status of the archived viewpoint back to Active. See Understanding the Lifecycle of Data Objects and the Data Chain.

Performing Copy and Model After Operations in Subscriptions

You can configure subscriptions to perform copy or model after operations for nodes that are added to the target viewpoint.

Terminology Note:

- **Copy**: Create a new node with the same property values as an existing node. See Adding a Node to a Hierarchy.
- **Model After**: Create a new node with the same property values and hierarchy relationships as an existing node. See Adding a Node by Modeling After an Existing Node.

For subscriptions, when adding nodes from a source viewpoint into a target viewpoint these operations enable you to copy property values from the target viewpoint, and to insert the new nodes into hierarchies that only exist in the target viewpoint.

You can also specify a set of properties to match on so that when a node is added in the target, the property values are copied from a sibling node that has the same values for the properties that you specified. For example, when adding an employee node in a subscription you can match on Location so that when the employee is added to the target viewpoint, the system automatically copies property values from a sibling node in the target viewpoint that has the same location.

Configuring a Subscription For Copy or Model After

Use the **Copy Action** and, optionally, the **Properties to Match** fields on the Definition tab of the subscription inspector to specify a copy or model after operation, as follows:

- **Copy Action**: Enables you to specify a copy or model after operation:
  - **None**: Do not perform a copy or model after operation.
  - **Properties**: Perform a copy operation to copy properties of the specified node in the target viewpoint.

- **Model After**: Perform a model after operation to copy properties and relationship values of the specified node in the target viewpoint.

> **Note:**
>
> When the target viewpoint is a list, selecting either **Properties** or **Model After** will copy properties only.

- **Properties to Match** (hierarchy source viewpoints only): Enables you to select one or more properties to identify a node that will be copied for each node that is added to the target viewpoint. The first sibling of an added node that has the same values of the properties that you select is used for the copy or model after operation.

> **Tip:**
>
> The **Properties to Match** option enables you to perform copy or model after operations in the target viewpoint when you did not perform a copy or model after operation in the source viewpoint.

- **Ignore Source Parent** (hierarchy source viewpoints only): Enable to ignore source parents for new nodes being added to target viewpoints. When enabled, new nodes are inserted under all of the parents of the node being modeled after when the source parent is not being brought over to the target viewpoint.

> **Tip:**
>
> Enable this option when performing a Model After operation where the parents of the nodes in the source viewpoint do not exist in the target viewpoint.

See Creating, Editing, and Validating Subscriptions.

Copy and Model After Processing in Subscriptions

After you configure your subscription with a copy or model after operation, those operations are processed in the target viewpoints as follows:

- **Generating the subscription request file:**
  - When you copy or model after a node in the source viewpoint, the name of that node is placed in the `Copy Node` column and the action is placed in the `Copy Action` column of the request file for the subscription request.
  - When you add a node (including add actions that result from inserting or moving a node under a top node filter) and you have configured **Properties to Match**, the system checks to see if there are other sibling nodes in the source that have the same property values as the ones that you specified to match. If so, the first sibling that has the same values is placed in the `Copy Node` column of the request file for the subscription request and the action is placed in the `Copy Action` column

> **Note:**
>
> All **Properties to Match** must have exact matching values on the source node, and none of them can be null or blank. If any of the properties to match has a null or blank value, the `Copy Node` and `Copy Action` columns are not populated in the request file for the subscription.

See Request Load File Format.

- **Processing the subscription request file:**
  - **Copy operations**: Defined property values of the node in the `Copy Node` column of the subscription request file are copied in the target viewpoint.

    > **Note:**
    >
    > Only property values from the source viewpoint that are different from the values in the target viewpoint are copied.

  - **Model After operations**:
    - Defined property values of the node in the `Copy Node` column of the subscription request file are copied in the target viewpoint.

      > **Note:**
      >
      > Only property values from the source viewpoint that are different from the values in the target viewpoint are copied.

    - Parents of the node in the `Copy Node` column of the subscription request file are copied in the target viewpoint.

      > **Note:**
      >
      > Only relationship values from the source viewpoint that are different from the values in the target viewpoint are copied.

> **Note:**
>
> If the `Core.Name` property for the node type of a node being added via a copy or model after operation is configured to be calculated and stored (see Calculated and Stored Properties), the name of the new node is automatically calculated.

For an example of a model after operation in a subscription, see Model After Subscription Example.

Model After Subscription Example

The following example demonstrates how property values are processed in model after operations in subscriptions.

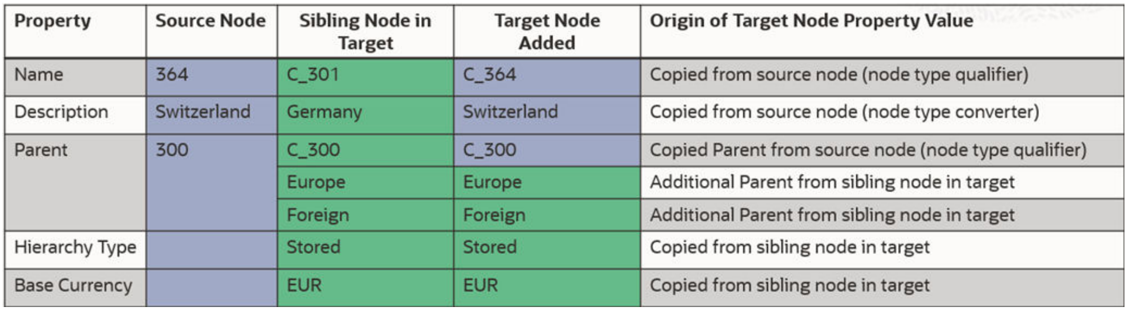
In this example, we have a Consolidation and Close viewpoint that is subscribed to a GL viewpoint. We have configured the subscription as follows (see Configuring a Subscription For Copy or Model After):

- **Copy Action**: Model After
- **Properties to Match**: Summary

When we add the node in the source, the system finds the first sibling node with the same value for the **Summary** property to use as the node to model after. In this example, we are adding a node in the source with Summary set to **No** (so, a bottom node) under the **300** (EMEA) parent. The first sibling node under 300 with Summary set to No is **301** (Germany). So that is the node that will be used for the model after operation.

In the source viewpoint, the node that we are adding is under one parent: **300** (EMEA). However, because we are performing a model after operation, in the target viewpoint in addition to adding the node under the equivalent parent, **C_300**, we also want to insert the node anywhere else that the **301** (Germany) node exists. In this case, that means inserting the node under two additional parents, **Europe** and **Foreign**.

The following table shows the result of the add in the source viewpoint with the origin of the property values that are created in the target viewpoint:



Property	Source Node	Sibling Node in Target	Target Node Added	Origin of Target Node Property Value
Name	364	C_301	C_364	Copied from source node (node type qualifier)
Description	Switzerland	Germany	Switzerland	Copied from source node (node type converter)
Parent	300	C_300	C_300	Copied Parent from source node (node type qualifier)
		Europe	Europe	Additional Parent from sibling node in target
		Foreign	Foreign	Additional Parent from sibling node in target
Hierarchy Type		Stored	Stored	Copied from sibling node in target
Base Currency		EUR	EUR	Copied from sibling node in target

Note the following about the property values that are created for the node that gets added in the target viewpoint:

- The **Name** and **Description** property values are copied from the source node.
- The **C_300** parent is copied from the source node (the target node type has a qualifier that adds a "C_" prefix).
- The **Europe** and **Foreign** parents are copied from the first matching sibling node in the target viewpoint.
- The **Hierarchy Type** and **Base Currency** property values are copied from the first matching sibling node in the target viewpoint.

Assigning Subscription Requests

When you create a subscription, you can optionally identify a default assignee for the requests generated by that subscription, as well as alternate assignees if the default assignee is out of the office.

The assignee is added to the target subscription requests as the request creator. This enables assignees to view and modify draft requests that are assigned to them.

You can assign users as the default assignee, and users or groups as alternate assignees. For groups, at least one person in the group must have complete *Participant (Write)* permission (all actions allowed and all properties able to be edited) on the dimension in the target viewpoint in order for the group to be assigned.

> **Note:**
>
> If you select a group as an alternate assignee, when subscription requests are generated, they are assigned one by one to the available members of the group that have the appropriate permissions until all available members of the group have been assigned a request. A user is not a repeated until all valid members of the groups have been assigned a request in the latest round of assignment. Then, the assignment starts over with the first available group member again.

Enable the **Collaborate** option to add alternate assignees who are not currently assigned a request as collaborators. This enables them to make changes to request items and add comments or attachments. See Creating, Editing, and Validating Subscriptions.

Assignee Priority

Users and groups are assigned to subscriptions in this order:

1. The request is always assigned to the default assignee if the assignee is available and has the appropriate permissions.
2. If the default assignee is out of the office (see Setting Your Preferences) or does not have appropriate permissions, the request is assigned to the users listed in the alternate assignees as follows:
   a. Request assignment begins with the first user listed in the alternate assignees and continues down the list of alternate assignees.
   b. If an alternate assignee is out of the office or does not have sufficient permissions, the request is assigned to the next alternate in the list.
   c. After all valid users in the alternate assignee list have been assigned a request, the current assignment round is considered completed and the next round of assignments is initiated.

   > **Note:**
   >
   > If a user is listed multiple times in the alternate assignees (for example, if a user is listed as an alternate assignee and is also part of a group that is listed as alternate assignees), the user is assigned a request only one time per assignment round.

If you assign a default assignee but no alternate assignees and the default assignee is not available, the request is assigned to the default assignee so that it can be worked when the default assignee is back in the office. If you assign alternate assignees but not a default assignee and none of the alternates are available, then the subscription is skipped and a notification is sent to the subscription owner.

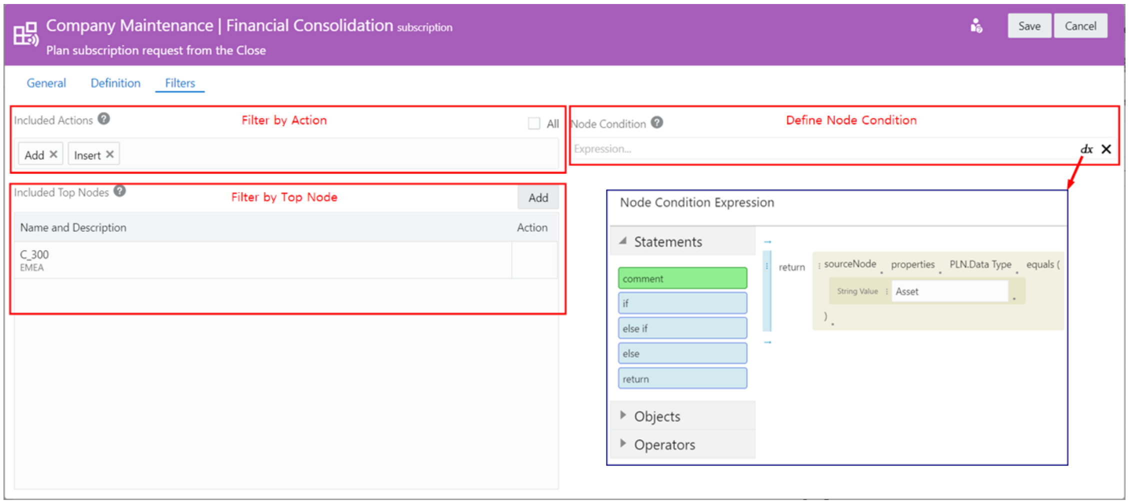
> **Note:**
>
> The subscription request assignee does not need any permissions on the source viewpoint.

Assignees receive email notifications when the request in the source viewpoint is submitted. See Request Notifications.

Adding Filters and Conditions to Subscriptions

You can configure a subscription for specific actions, top nodes (for hierarchy viewpoints), or node conditions. For example, you can subscribe to only Add and Insert actions, or to nodes in a hierarchy viewpoint under the Europe top node, or to nodes whose account type is Asset. When the subscription request is generated, only the request items from the source viewpoint that meet the filters and conditions that you specify are created for the target viewpoint.

Combining filters and conditions enables you to select the nodes and actions that you want to subscribe to on a more granular level. The following example combines the filters and conditions described above to subscribe only to Add and Insert actions on nodes with the Asset account type under the EMEA top node.



Company Maintenance | Financial Consolidation subscription
Plan subscription request from the Close

Included Actions: Add ✕ Insert ✕ □ All

Included Top Nodes: C_300 EMEA

return sourceNode . properties . PLN.Data Type . equals (String Value : Asset .) .

To add filters and conditions, see:

- Filtering by Actions
- Filtering by Top Node
- Defining Node Conditions

Filtering by Actions

By default, when you subscribe to a source viewpoint a subscription request item for the target viewpoint is generated for each action taken in the source. Action filters enable you to subscribe to only a subset of actions in the source viewpoint. For example, you can subscribe only to Add and Insert actions, or only to property updates. When users perform actions in the source viewpoint, only the actions that you specify will generate a target subscription request item.

Considerations

- You can add action filters for subscriptions in both hierarchy and list viewpoints.
- If you clear the **All** check box but you do not specify a subset of actions for the action filter, source requests that contain property updates will still generate request items for those property updates in the target viewpoint.
- Similarly, if a source request item contains an action that you have not included in your filter as well as a property update, a request item just for the property update is generated for the target viewpoint. For example, if your filter does not include a Move action and a source request item contains both a Move and a property update, a request item for the target viewpoint is created for the property update only.

  The specific property updates are handled by the node type converter between the source and target viewpoints. See Working with Node Type Converters.

> **Note:**
>
> If your filter contains Insert but not Add, or Remove but not Delete, Adds and Deletes in the source are converted to Inserts and Removes in the target when the subscription request items are processed.

To add an action filter:

1. From **Views**, open the view that contains the target viewpoint.
2. Select the target viewpoint.
3. Place your cursor to the right of the viewpoint name, click ⋮, and then select **Inspect**.
4. Select **Subscriptions**, and then click the name of the subscription that you want to add an action filter to.
5. In the Subscription Inspector, select the Filters tab, and then click **Edit**.
6. In **Included Actions**, clear the **All** check box.
7. Click in **Included Actions** to display a drop down menu of actions that are available in the source viewpoint, and then select an Action to add.
8. When you are finished selecting actions, click **Save**.

To remove an action from the action filter, click **Edit**, and then click the **X** next to the action that you want to remove.

Filtering by Top Node

For subscriptions in hierarchy viewpoints, you can add top node filters that enable you to subscribe to a portion of a source viewpoint instead of the entire node set. You can specify multiple top nodes to subscribe to. When an update is made to the source viewpoint, if the update is within a branch that you have selected, a request is generated to make the same update in the target viewpoint. If the update in the source viewpoint is not within a branch that you have selected, no subscription request items are generated for that branch.

Considerations

- You can choose any node in a hierarchy as a top node for a filter, including nodes in the same ancestor chain or bottom-level nodes with no children.

- You cannot add top node filters to a subscription if the source viewpoint is based on a list type node set.
- Some request actions in the source viewpoint are translated to equivalent actions in the target viewpoint:
  - Add and Insert actions in the source may become Update actions in the target if the node already exists in the target.
  - If you move a node in the source out of the top node branch in the target, it becomes a Remove action in the target.
  - If you move a node in the source into the top node branch in the target, it becomes an Update action in the target, which may become an Insert if shared nodes is enabled or a Move if shared nodes is not enabled.
- When a parent node that is located under a different top node is inserted or moved under a node that is specified in a subscription top node filter, the descendants of the parent node are also inserted and moved as part of the subscription request.

To add a top node filter:

1. From **Views**, open the view that contains the target viewpoint.
2. Select the target viewpoint.
3. Place your cursor to the right of the viewpoint name, click ⋮, and then select **Inspect**.
4. Select **Subscriptions**, and then click the name of the subscription that you want to add a top node filter to.
5. In the Subscription Inspector, select the Filters tab, and then click **Edit**.
6. In **Included Top Nodes**, click **Add**.
7. In the node selector, select one or more top nodes to subscribe to. Perform an action:
   - Type the name of a top node that you want to subscribe to in the search bar and hit Enter to search for a node directly.
   - Click ▶ and drill to the top node that you want to subscribe to.

   Nodes that you select are indicated by a check mark and added to the Selected tab. Click the **X** to remove a node from the Selected tab.

   > **Note:**
   >
   > By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** ☰ to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.

8. When you have finished selecting top nodes, click **OK**, and then click **Save**.

To remove a top node filter:

1. On the Definition tab of the subscription inspector, click **Edit**.

2. In Included Top Nodes, place your cursor in the Action column of the top node that you want to remove, click ⋮, and then select **Remove**.

Defining Node Conditions

You can configure a node condition expression to define the nodes that you want to subscribe to in a source viewpoint. For example, you can subscribe only to active nodes, or nodes at a specific level in a hierarchy. When you configure a node condition expression, only actions taken on the nodes that are evaluated to True for the expression will generate subscription request items for the target viewpoint.

Only properties that are configured for the node types associated with the viewpoint are available to be selected in the expression builder.

> **Note:**
>
> Node condition expressions are not applied to request items with the Remove, Rename, or Delete action.

To define a node condition expression:

1. From **Views**, open the view that contains the target viewpoint.
2. Select the target viewpoint.
3. Place your cursor to the right of the viewpoint name, click ⋮, and then select **Inspect**.
4. Select **Subscriptions**, and then click the name of the subscription that you want to add an action filter to.
5. In the Subscription Inspector, select the Filters tab, and then click **Edit**.
6. In Node Condition, click **Define Expression** *fx* to launch the expression builder.
7. In expression builder, create the node condition expression. Because you are evaluating nodes in the subscription viewpoint for the condition, you use the *sourceNode* object in the expression builder.

   The expression builder guides you to create an expression that evaluates to a Boolean value only. See Building Expressions.
8. When you are finished creating the node condition expression, click **Apply** to return to the Subscription Inspector, and then click **Save**.

Filtering by Inclusion Property

You can specify that a node will be included or excluded from a subscription by configuring an inclusion property on the node.

You can use this inclusion property to, for example, add a node in a source viewpoint without immediately adding it to the target viewpoints. Then, when you are ready to add the node to the target viewpoints you can change the inclusion property value in the source viewpoint to generate the subscription request actions to add that node to any subscribing target viewpoints.

> **Note:**
>
> The inclusion property must be a node level property.

An inclusion property is a Boolean or List data type property that controls whether a node in a source viewpoint gets sent to the target viewpoint as part of a subscription.

- If the inclusion property is a Boolean data type, the node is added to the target viewpoints when the Boolean value is True, and removed from target viewpoints when the value is False. Typically, you use a Boolean inclusion property when there is a single subscribing target viewpoint and you want to specify whether or not to include a node in that subscription.
- If the inclusion property is a List data type, you must specify the inclusion value (either by entering the value or selecting the value from an Allowed Values list). Use a List data type property when you have multiple subscribing target viewpoints and you want to specify which target viewpoints the node should be included in subscriptions for.

When the inclusion property is set in the source viewpoint to include the node (that is, has a value of `True` for Boolean or contains the inclusion value for a list), request actions for the node are generated for the target viewpoint by the subscription. If the node does not already exist in the target viewpoint, it is added or inserted.

When the inclusion property is set in the source viewpoint to exclude a node (that is, has a value of `False` for Boolean or does not contain the inclusion value for a list), the node is removed or deleted if it exists in the target viewpoint. If it does not exist, no request actions are generated for the excluded node.

Best Practice

If you want to retain property values and child relationships for nodes that you are excluding from the target viewpoint, it is a best practice to remove the Delete action from the **Included Actions** filter on the subscription so that the node is removed only. See Filtering by Actions.

Considerations

- You can configure an inclusion property filter using any Boolean or List data type property from any node type in the source viewpoint.

  > **Note:**
  >
  > You can configure an existing property as an inclusion property, or you can create a new property to use as the inclusion property for this subscription. See Creating Properties Manually.

- The inclusion property can have a Default Type of `None`, `Derived and Stored`, or `Specified`.
  The Specified default type can be useful in these scenarios:
  – By default, you want to include all nodes in a subscription, and you will explicitly specify which nodes you do not want to include. You can set the default value to include the node (has a value of `True` for Boolean or contains the inclusion value for a list) and then change the property only for the nodes that you want to exclude.
  – By default, you want to exclude all nodes in a subscription, and you will explicitly specify which nodes you want to include. You can set the default value to exclude the

node (has a value of `False` for Boolean or does not contain the inclusion value for a list) and then change the property only for the nodes that you want to include.

The Derived and Stored default type can be useful when you want to use a derived expression to determine which nodes are included in a subscription.

If you have a default type of `None`, you will have to specify the inclusion property value for every node.

- The Default Type cannot be `Derived` or `Inherited`. After you designate a property as an inclusion property, you cannot change that property's default type to be either derived or inherited.

> **Note:**
>
> You can designate a `Derived and Stored` property as an inclusion property.

- Node expression, top node, and action filters are applied to the generated request actions. For example, if the inclusion property on a node is set to include but the node is under a top node that has been filtered out of the subscription, request actions are not generated for that node.
- Some request actions in the source viewpoint are translated to equivalent actions in the target viewpoint:
  - If you modify the inclusion value in the source viewpoint from False to True (or to contain the inclusion value, for a list), the node is added (or inserted, in a hierarchy set) in the target viewpoint if it doesn't already exist.
  - If you modify the inclusion value in the source viewpoint from True to False (or not to contain the inclusion value, for a list), the node is deleted (or removed, in a hierarchy set) if it exists in the target viewpoint.
  - If you do not modify the inclusion value in the source viewpoint, request actions that you perform in the source viewpoint are propagated to the target viewpoint if the inclusion property is set to True (or to contain the inclusion value, for a list).
- For shared nodes when the target viewpoint is a hierarchy:
  - If the inclusion property is updated to either include or exclude a node in the source viewpoint, request actions are generated for all instances of the node.
  - If the inclusion property is not updated and the node is set to be included, request actions are generated only for the instances where the action was performed.
  - If the inclusion property is not updated and the node is set to be excluded, request actions are not generated for any instance of the node.

  For example, suppose you have a shared node under parents A, B, and C in the source viewpoint.

  - If you update the inclusion property to include a node under parent A in the source viewpoint, request actions are generated to add the node under parent A and insert the node under parents B and C in the target viewpoint. Similarly, if you update the inclusion property to exclude a node under parent A in the source viewpoint, request actions are generated to delete the node under parent A and remove it from parents B and C in the target viewpoint.
  - If you do not update the inclusion property and the node is set to be included, request actions are generated only for the instance of the node where you performed the action. So if you edited a property for the node under parent B in the source viewpoint,

a request action is generated to modify that property only for the node under parent B in the target viewpoint.

- If you do not update the inclusion property and the node is set to be excluded, request actions are not generated for the nodes no matter which instance in the source viewpoint you perform an action on.

> **Tip:**
>
> When you have multiple subscribing viewpoints, you can use a List data type inclusion property if there is only one user, such as a Service Administrator, who will determine which viewpoints will receive which nodes. If there are different users who each determine whether or not a property is sent to the viewpoint that they administer, you can set up multiple Boolean data type inclusion properties and use property-level data access to control which users can view or edit the Boolean property for their viewpoint.
>
> For example, you can configure data access so that only the Planning administrator is able to edit the "Include for Planning?" property, and only the Consolidation administrator is able to edit the "Include for Consolidation?" property in the source viewpoint. See *Boolean Data Type Inclusion Property Example* in Inclusion Property Examples.

> **Note:**
>
> If you are already using a node condition to filter nodes out of subscriptions based on the value of a Boolean or List data type node level property (see Defining Node Conditions), you should replace the node condition filter with an inclusion property filter.
>
> If you are using a node condition filter on a relationship level property, you should keep the existing node condition filter because you cannot create an inclusion property filter on a relationship level property.

Configuring the Inclusion Property

1. Inspect the viewpoint that contains the subscription that you want to configure the inclusion property for, and click the Subscriptions tab.
2. Create a new subscription or edit an existing subscription. See Subscribing to Viewpoints.
3. On the Filters tab, click **Edit**.
4. From the **Inclusion Property** drop down menu, select a property to configure as the inclusion property. Only node properties with the Boolean or List data type are available to be selected.
5. If the property that you selected has the List data type, in **Inclusion Value** enter an inclusion value (or select the value if the property uses Allowed Values).

> **Note:**
>
> If the property has application or node type overrides for Allowed Values, all of the allowed values from all overrides are available to be selected as inclusion values.

6. Click **Save**.

Inclusion Property Examples

List Data Type Inclusion Property Example

To illustrate the concept of using a list data type inclusion property to specify which subscribing target viewpoints to include changes to for a node in the source, suppose you have a GL source viewpoint with subscriptions set up to synchronize changes in both a Planning and a Consolidation viewpoint. You can create an inclusion property to determine whether or not to include a node in a subscription to either or both viewpoints as follows:

1. Create a new property to use as your inclusion property with the following details:

Field	Value
Template	`Custom.List Template`
Name	`Applications to Include`
Description	`Determines which target applications this node should be included in subscriptions for.`
Level	`Node`
Default Type	`Specified`
Default Value	`Plan, Consol`
Editable	`True`
Inherited?	`None`

   See Creating Properties Manually

2. Add the property to the appropriate node types in the GL (source) viewpoint. See *Adding a Property to a Node Type* in Editing Property Parameters.
3. In the subscription for the Planning (target) viewpoint, specify the following for the inclusion property:
   - **Inclusion Property**: `Applications to Include`
   - **Inclusion Value**: `Plan`
4. In the subscription for the Consolidation (target) viewpoint, specify the following for the inclusion property:
   - **Inclusion Property**: `Applications to Include`
   - **Inclusion Value**: `Consol`

Now, when you work with nodes in the GL viewpoint, you can control whether or not they are included in the target viewpoint subscriptions by editing the `Applications to Include` property:

- If the `Applications to Include` property contains the values of `Plan` or `Consol`, the node is added to the Planning or Consolidation target viewpoints.

- If the `Applications to Include` property does not contain the values of `Plan` or `Consol`, the node is removed from the Planning or Consolidation target viewpoints.

Boolean Data Type Inclusion Property Example

For Boolean data type inclusion properties, let's return to the scenario with a GL source viewpoint with subscriptions set up to synchronize changes in both a Planning and a Consolidation viewpoint, and this time by default nodes added to the GL viewpoint get added to Planning automatically but do not get added to Consolidation unless the Consolidation administrator determines that they should be.

You can create two Boolean inclusion properties with a default of True for Planning and False for Consolidation, and then you can give each administrator data access only to the property for their viewpoint to determine whether or not to include a node from the GL viewpoint.

1. Create two new properties to use as your inclusion properties with the following details: .

Field	Property 1	Property 2
Template	`Custom.Boolean Template`	`Custom.Boolean Template`
Name	`Include in Planning?`	`Include in Consol?`
Description	`Determines whether or not this node should be included in subscriptions for Planning.`	`Determines whether or not this node should be included in subscriptions for Consolidation.`
Level	`Node`	`Node`
Default Type	`Specified`	`Specified`
Default Value	`True`	`False`
Editable	`True`	`True`
Inherited?	`Deselected`	`Deselected`

   See Creating Properties Manually

2. Add the property to the appropriate node types in the GL (source) viewpoint. See *Adding a Property to a Node Type* in Editing Property Parameters.
3. Grant the Planning and Consolidation viewpoint administrators write data access to the `Include in Planning?` and `Include in Consol?` properties, respectively. See Configuring Data Access.

Now, by default new nodes will be included in subscriptions for the Planning viewpoint but not for the Consolidation viewpoint. However, the viewpoint administrators can now edit the properties in their respective subscriptions to include or exclude specific nodes in their applications:

- The Planning administrator can change the `Include in Planning?` property to `False` in the Planning subscription in order to exclude the node in the Planning (target) viewpoint.
- The Consolidations administrator can change the `Include in Consol?` property to `True` in the Consolidation subscription in order to include the node in the Consolidation (target) viewpoint.

9
About Requests

In Oracle Fusion Cloud Enterprise Data Management, all changes to data are made by using requests. Requests can be user-generated or system-generated by using subscriptions, and they can occur across hierarchies within a single application or across applications depending on how you structure the view in which the request is made. These changes to data are organized into request items. A request item is a group of change actions for a specific node. Requests also enable you to visualize changes before you commit them.

Videos

Your Goal	Watch This Video
Learn more about using requests to make changes to data.	Overview: Understanding Requests in Enterprise Data Management Cloud

Change actions that can be performed in requests are:

- Add, insert, move, reorder, remove, and delete nodes
- Add and insert top nodes
- Update properties

There are three ways to make changes to data in Cloud EDM:

- Interactively (manual changes to viewpoints). For example, in Cloud EDM, you add node Florida as a child node of Southeast-US and you move node Georgia from parent South-US to parent Southeast-US. With each change you make in the user interface, an item is added to the request. See Making Changes Interactively.
- By subscribing a target viewpoint to a source viewpoint. For example, you subscribe an Entity viewpoint in your Planning application to an Entity viewpoint in your General Ledger application. When a cost center is added in the Entity viewpoint in the General Ledger application, the system generates a request to add that cost center in the Entity viewpoint in your Planning application as well. See Subscribing to Viewpoints.
- Loaded from a file. For example, you are creating a new United States hierarchy and need to add 50 new nodes (one to represent each state). Each change in the load file is a separate item in the request. In this scenario, there would be 50 items in the request, one for each state node being added. See Making Changes Using a Load File.

Whether you make changes interactively, through a subscription, or by using a load file, all proposed changes are visualized in the viewpoints where you are making the changes or loading the file. Changes are visible in unique colors and icons so that you can see which parts of the hierarchy or list were changed and what areas may be affected by the change.

While making changes to viewpoints, the system performs validation checks to determine the validity of the changes that are being proposed. If items in the request have validation issues, you can see the reason for the failure and decide how to resolve the issue.

When a request is submitted, validation checks run again, the changes are committed, and the viewpoint is updated with the changes.

You cannot perform any request actions in time labeled viewpoints. See Working with Time Labeled Viewpoints.

Request Types and Process Flow

There are four types of requests. Interactive requests are created manually by users. Subscription requests are generated from a subscription. Merge requests are generated from an import or load in Merge mode. Consolidation requests are combined requests that are made up of multiple consolidated source requests.

Interactive Requests Process Flow

The process flow for creating requests interactively is:

1. Open a view, see Opening a View.
2. Create a request, see Making Changes Using Requests.
3. Make changes to viewpoints. Each change that you make to a view is associated with an item in the request and is visualized using icons and colors, see Visualizing Request Changes.
   - Interactively, see Making Changes Interactively.
   - Load from a file, see Using a Load File.
4. Add comments to the request, see Adding Request Comments.
5. Validate the request or request items, see Validating Requests.
6. Submit the request, see Submitting a Request.
7. View submitted requests, see Working with Request Activity.

Subscription Requests Process Flow

The process flow for working with requests that were created by a subscription is:

1. Subscribe a target viewpoint to a source viewpoint, see Creating, Editing, and Validating Subscriptions.
2. When a change is committed in the source request, the subscription assignee receives a notification, see Request Notifications.
3. Review changes to viewpoints in a subscription request. Each change received through a subscription is associated with an item in the request and is visualized using icons and colors, see Visualizing Request Changes.
4. Add comments to the request, see Adding Request Comments.
5. Submit the request, see Submitting a Request.
6. (**Optional**) Compare the request file attachment and request item details to review how the request items in the subscription request were processed, see Reviewing Request File Attachments.
7. View completed requests, see Working with Request Activity.

Import and Load Requests Process Flow

The process flow for working with requests that are generated as a result of an import or a load in merge mode is:

1. Import data to an existing dimension or load data to an existing viewpoint using Merge mode. See Importing Dimensions or Running Viewpoint Loads.

   If the request is in Submit or Approve stage, it is displayed in the request view. If it is closed, it is displayed in the inspector. You can view the import results in Transaction History.

   > **Note:**
   >
   > For imports, after the import is complete you can click the import request link to navigate to the import request.

2. Do one of the following:
   - If the import or load contained request actions for a source viewpoint of a subscription, subscription requests are generated for the target viewpoints. See **Subscription Requests Process Flow** above.
   - If the request is in the Submit or Approve stage, continue with the following steps.
3. Review changes in the request. Each change received through a merge import or load is associated with an item in the request and is visualized using icons and colors, see Visualizing Request Changes.
4. Add comments to the request, see Adding Request Comments.
5. Submit the request, see Submitting a Request.
6. View completed requests, see Working with Request Activity.

Consolidation Request Process Flow

See Understanding Request Consolidation.

Request Items Limits by Request Type

Each request type supports the following maximum number of request items:

- Interactive (either interactively or via a file load): 10,000 request items
- Subscription: 12,000 request items (this allows room for up to 10,000 initial request items plus an additional 2,000 request items that are generated as part of the subscription process)
- Import or Load in merge mode: 20,000 request items
- Consolidation: 12,000 request items

Request Actions

This table describes the valid request actions that can be performed in a list and a hierarchy.

Request Action	What Happens	Hierarchy	List
Add node	You add a new node to a list or hierarchy.	Yes	Yes

Request Action	What Happens	Hierarchy	List
Delete node	You delete a node from all parents in all hierarchies and delete it from the node type for the application.	Yes	Yes
Update node properties	You make changes to node properties.	Yes	Yes
Rename node	You change the name of a node.	Yes	Yes
Insert node	You insert an existing node under a parent node in the hierarchy. You can insert a node from a list node set, a different hierarchy set, or the same hierarchy set if the hierarchy set allows shared nodes.	Yes	No
Move node	You move an instance of a node from one parent to another.	Yes	No
Reorder node	You change the order of nodes under a parent.	Yes	No
Remove node	You remove a node from a specific parent. Other instances of the node (under different parents) are not removed.	Yes	No
Add top node	You add a new node to be a top node in the hierarchy.	Yes	No
Insert top node	You select an existing node to be a top node in the hierarchy.	Yes	No

Predefined Members in Cloud EDM

Planning Modules, Financial Consolidation and Close, Enterprise Profitability and Cost Management, and Tax Reporting applications all support predefined members. Predefined members are nodes where the `Enterprise Predefined Member` property is set to `True`, and they have the following restrictions:

- They cannot be edited in Oracle Fusion Cloud Enterprise Data Management.
- They can only be moved, reordered, or have child nodes added under them in Cloud EDM.

Visualizing Request Changes

Changes made to a view are visible in unique colors and icons so that you can see which parts of the hierarchy or list were changed and what areas may be affected by the change.

Icon	Color	Meaning
⊕	green	Node added Adding a Node to a List Adding a Child or Sibling to a Hierarchy Adding a Top Node
➡	green	Node inserted Inserting a Node Selecting a Top Node
⊖	red	Node removed Removing a Node
🗑	red	Node deleted Deleting a Node
✥	yellow	Node moved Moving a Node
⇅	yellow	Node reordered Reordering a Node
✏	yellow	Node property updated Updating Node Properties

10

Making Changes Using Requests

Before you can make changes to a view, you must create a request. Each change that you make to a view is associated with an item in the request.

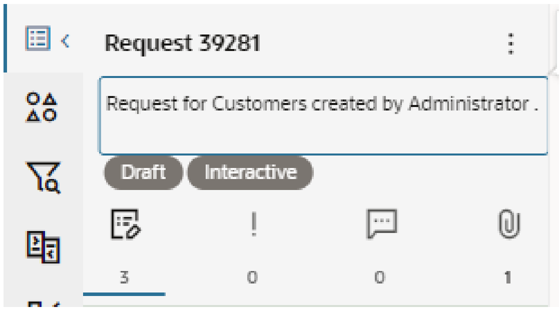
Videos

Your Goal	Watch This Video
Learn about making changes using requests.	▶ Managing Complex Data Changes With Requests.

Creating Requests

To create a request:

1. Open a view.
2. Click **New Request**. The Requests pane opens.
3. Change the title of the request by clicking the default name "**Request *nnnn***" and entering a new name.
4. Change the description of the request by clicking in the description and entering a new description.

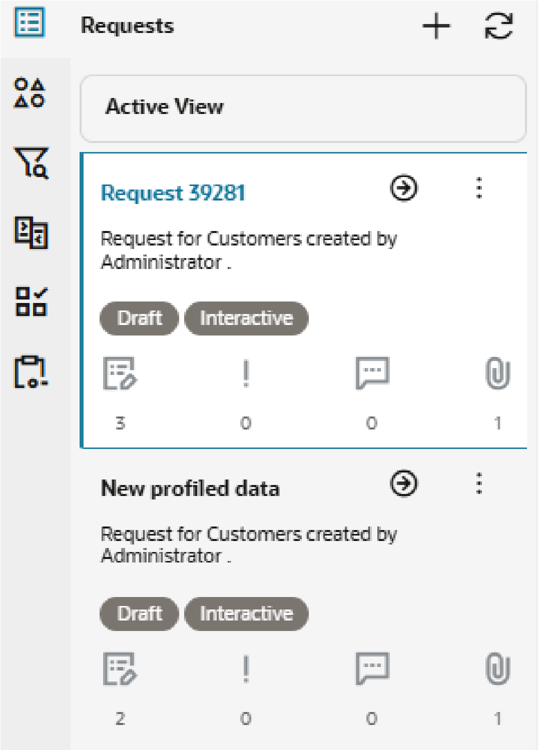


Request 39281

Request for Customers created by Administrator .

Working with the Requests Pane

From an open view, click **Requests** ☰ to open the Requests pane.



Requests

Active View

Request 39281

Request for Customers created by Administrator .

Draft Interactive

3 0 0 1

New profiled data

Request for Customers created by Administrator .

Draft Interactive

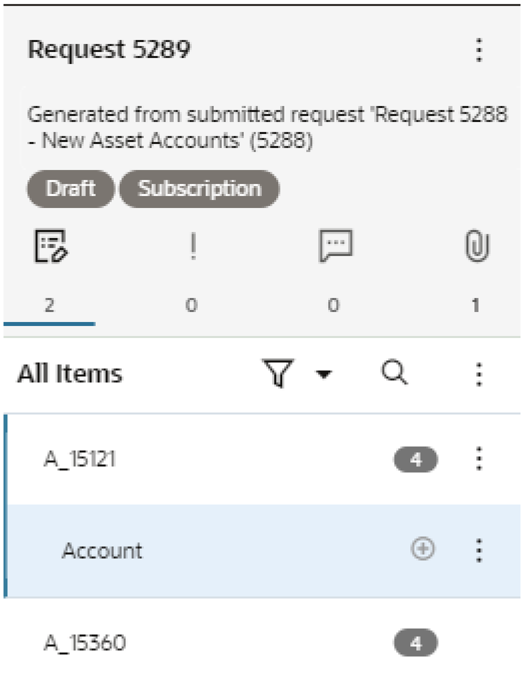
2 0 0 1

From the Requests pane, use these navigation tools:

- + to add a new request.
- ⟳ to refresh the list of displayed requests. This is helpful when users are working on a request at the same time. For more information, see Concurrent Editing of a Request.
- **Active View** to return to the active view after viewing request information.
- ➔ to open a request. When you're done viewing the request item details, click ☰ ‹ to go back to the Requests pane.
- ⋮ to display the action menu from which you can inspect or delete the request.

Working with a Request

Opening a request displays the following information:



Request 5289 ⋮

Generated from submitted request 'Request 5288 - New Asset Accounts' (5288)

Draft Subscription

2 0 0 1

All Items

A_15121 4

Account

A_15360 4

The following tabs are available:

- **Items** 📝: Displays the request items in a request. See Working with Request Items.
- **Request Issues** !: Displays the request items with validation issues. See Working with Request Validation Issues.
- **Comments** 💬: Displays request comments. See Working with Request Comments.
- **Attachments** 📎: Enables you to view and download request attachments. See Reviewing Request File Attachments.

From the request pane, click **Search** 🔍 to search for a request item, and click the **Action** menu ⋮ to inspect, validate, or delete a request item.

Making Changes Interactively

You can make changes interactively to a list or hierarchy viewpoint.

Best Practice

If you need to make mass changes to a viewpoint, it's a best practice to download the viewpoint to Excel, make mass updates in the spreadsheet, and then load the changes to the viewpoint using a request. For more information, see Making Changes Using a Load File.

To make changes interactively to a view:

1. Open the view that you want to make changes to.
2. Open a draft request or create a new request.

Videos

Your Goal	Watch This Video
Learn about making changes with requests.	Managing Complex Data Changes With Requests

See these topics to make changes to a list viewpoint:

- Adding a Node to a List
- Deleting a Node
- Updating Node Properties
- Searching for Nodes

See these topics to make changes to a hierarchy viewpoint:

- Adding a Top Node
- Selecting a Top Node
- Adding a Node to a Hierarchy
- Inserting a Node
- Moving a Node
- Reordering a Node
- Removing a Node
- Deleting a Node
- Updating Node Properties
- Searching for Nodes

Adding a Node to a List

You can add a new node or an existing node from another viewpoint to a list. To add a node from another viewpoint, the viewpoints need to be in the same view and use a node type converter. You can drag and drop the node from a viewpoint or a request item.

> **Note:**
>
> You can add a node to a viewpoint if Add is an allowed action for the node type.

You can also copy a node to create a new node. Properties are copied to the new node and can be changed if needed.

Adding a New Node

To add a new node to a list:

1. Open the view that you want to make changes to and then either open a draft request or create a new request.
2. From a list viewpoint, do one of the following: .

- Click + and then select **Add New**.
- To copy an existing node, click ⋮ to the right of the node name to launch the context menu, and then select **Copy**.
- To model after an existing node, click ⋮ to the right of the node name to launch the context menu, and then select **Model After**.

  > **Note:**
  >
  > You must have configured at least one related viewpoint in order to use model after in a list viewpoint. See Configuring Related Viewpoints.

> **Note:**
>
> A new node is added to the top of the list with a default name, for example Copy of 101. The default sort order for list viewpoints is by node creation date with the newest nodes at the top of the list.
>
> The node name may be automatically calculated if the node has been set up to calculate the name value. See Calculating and Storing the Name of a Node.

3. Modify the name for the new node and update node properties in the property pane.
4. Continue to make changes, or click **Done** to close the request.

Adding a Node from Another Viewpoint

You can add a node from another viewpoint by using the menu action or by dragging and dropping the node. You'll use side by side layout to display the viewpoints on the screen.

To add a node from another viewpoint:

1. Open the view that you want to make changes to and then either open a draft request or create a new request.
2. From a list viewpoint, perform an action:
   - Click + and then select **Add From**.
     a. Select a viewpoint (only viewpoints that have mapped node types are available).
     b. Browse or search for a node and select it.
   - **Drag and drop from another viewpoint**: Click ◫ to open viewpoints in side by side layout and drag and drop the node to the target list viewpoint.
   - **Drag and drop from a request item**: Open the request that has the request item to copy and drag and drop the request item to the target list viewpoint.

> **Note:**
>
> To be able to drag and drop the node to the target viewpoint, a node type converter must be set up.

3. If the node selected or dragged is a parent in a hierarchy viewpoint, select an option and then click **OK**:
   - **Node only**: Add only the selected or dragged node.
   - **Node and Children**: Add the node and its children.
   - **Node and Descendants**: Add the node and its descendants.
   - **Bottom Nodes**: Add only the bottom nodes of the selected or dragged node.

> **Note:**
>
> If multiple node types are available, you're prompted to select one.

Adding a Node to a Hierarchy

You can add a node as a sibling or child of an existing node. You can also copy a node to create a new node under the same parent as the selected node. Properties are copied to the new node and can be changed if needed. To create a single instance of a node use the Copy option. If you want to create a node under all parents that share the node in the viewpoint, use the Model After option.

> **Note:**
>
> The Model After option can only be used to copy a shared node.

To add additional top nodes to a view, see Adding a Top Node.

Some considerations for adding nodes to a hierarchy:

- You can add a node to a viewpoint if Add is an allowed action for the node type.
- Be aware that the request item count increases by two when you add a node to a hierarchy viewpoint. This is because there's one action to add the node and one action to insert the node into the hierarchy.
- You can't add orphan nodes to a hierarchy viewpoint but you can insert the node into the viewpoint. An orphan node is created when the node's parent is deleted. To add an orphan node, you can create a list viewpoint that includes the node type and then insert the orphan node into the desired hierarchy.

To add a node to a hierarchy:

1. Do one of the following:
   - Open the view that you want to make changes to and then either open a draft request or create a new request.

- From **Requests**, open a draft request. This opens the view, viewpoint, and the request.

2. Select a hierarchy viewpoint to work with.
3. Find and select a node, click ⋮ to the right of the node name to launch the context menu, and then select an option:
   - **Add Child**: To create a new node as a child of the selected node.
   - **Add Sibling**: To create a new node as a sibling of the selected node.
   - **Copy**: To create a new node by copying the selected node and inserting the node under the same parent.
   - **Model After**: To create new nodes by copying the selected node and inserting the node under all parents of the selected node in the current or related viewpoints. See Adding a Node by Modeling After an Existing Node.

   > **Note:**
   >
   > Menu options are available depending on the configuration of the node that you select. For example, if you select a bottom level node that does not allow children, the Add Child option is not available. If you select the top node, then Add Sibling is not available.

4. If more than one node type is available, select a node type for the new node; otherwise, go to the next step.

   > **Note:**
   >
   > A new node is added with a default name, for example Copy of 101.
   >
   > The node name may be automatically calculated if the node has been set up to calculate the name value. See Calculating and Storing the Name of a Node.

5. Modify the name for the new node and update node properties.
6. Continue to make changes, or click **Done** to close the request.

Adding a Node by Modeling After an Existing Node

The model after action enables you to create a node with the same property values and hierarchy relationships as an existing node. You can create the new node in the same viewpoint (for hierarchy viewpoints), or you can create the node in specified related viewpoints in the same view.

This enables you to copy an existing node and insert it into all of the related viewpoints in a single operation rather than having to perform individual request actions.

For example, suppose you had a list viewpoint and two hierarchy viewpoints that each used different hierarchy sets. In the list viewpoint, you configure the two hierarchy viewpoints as related viewpoints. Then, when you perform a model after action in the list viewpoint, a request is generated with these request items:

- An Add action in the list viewpoint for the copy of the existing node

- Insert actions in both hierarchy viewpoints that insert the new node under the same parents as the existing node in those hierarchy sets. For shared nodes, the new node is inserted under all of the parents of the original node in the hierarchy.

Considerations for Model After

- If you do not have permission to perform an action that is generated as part of a model after action, a validation error is displayed. For example, if you do not have Insert permission on a hierarchy set in a viewpoint and you perform a model after that generates that Insert action, the request action is generated and a validation error for that action is displayed.
- To perform a model after action in a list viewpoint, you must first specify at least one related viewpoint (see Configuring Related Viewpoints). If there are no related viewpoints for a list viewpoint, use the Copy action instead.

Considerations for Model After Using Related Viewpoints

- In order to perform a model after in a viewpoint other than the one containing the source node, you must first configure that viewpoint as a related viewpoint. See Configuring Related Viewpoints.
- For model after operations, related viewpoints are one-way. That is, if in Viewpoint A you set Viewpoint B as a related viewpoint, when you perform a model after action in Viewpoint A it will generate a request action to insert the node in Viewpoint B, but when you perform a model action in Viewpoint B it will not generate a request action to insert the node in Viewpoint A.
- If you specify a list viewpoint as a related viewpoint, property update actions are generated for each additional property in the list viewpoint for that node. For example, if list viewpoint A has list viewpoint B as a related viewpoint and list viewpoint B contains properties for a node that aren't in list viewpoint A, when you perform a model after action in list viewpoint A the additional properties in list viewpoint B are also copied.

Modeling After a Node

1. Open the viewpoint that you want to make changes to and then either open a draft request or create a new request.
2. Select the node instance in the viewpoint that you want model after.
3. Perform an action:
   - For list viewpoints, click **Actions** ⋮, and then click **Model After**.

     > **Note:**
     >
     > You must have at least one related viewpoint configured in order to use model after in a list viewpoint.

   - For hierarchy viewpoints, click **Actions** ⋮, then **Model After**, and then select an option:
     - **This Viewpoint**: Copies the selected node and inserts it under all parents of the selected node in the current viewpoint.
     - **Related Viewpoints**: Copies the selected node and inserts it under all parents of the selected node in the current viewpoint and all related viewpoints.

> **Note:**
>
> This option is available only if you have configured at least one related viewpoint.

The new node is created with the name "Copy of (original node)".

> **Note:**
>
> If the `Core.Name` property for the node type is configured to be calculated and stored (see Calculated and Stored Properties), the new node name is automatically calculated.

4. Modify the name for the new node and update node properties as needed.

> **Note:**
>
> You can also perform model after operations using a request load file (see Performing Copy and Model After Operations in a Request Load File) and using a subscription (see Performing Copy and Model After Operations in Subscriptions).

Adding a Top Node

You can create a new node as a top node for a hierarchy viewpoint. If you want to select an existing node to be a top node, see Selecting a Top Node.

To create a new top node:

1. Do one of the following:
   - Open the view that you want to make changes to and then either open a draft request or create a new request.
   - From **Requests**, open a draft request. This opens the view, viewpoint, and the request.
2. Select a hierarchy viewpoint to work with.
3. Click + to launch the menu and then select **Add Top Node**.
4. If more than one node type is available, select a node type for the new top node; otherwise, go to the next step.

   > **Note:**
   >
   > A new top node is added with a default name, for example New Dept.

5. Modify the name for the new top node and update node properties.
6. Continue to make changes, or click **Done** to close the request.

Inserting a Top Node

You can select an existing node to be a top node for a hierarchy viewpoint. If you want to create a new node as a top node see, Adding a Top Node.

To insert a top node:

1. Do one of the following:
   - Open the view that you want to make changes to and then either open a draft request or create a new request.
   - From **Requests**, open a draft request. This opens the view, viewpoint, and the request.
2. Select a hierarchy viewpoint to work with.
3. Click **Add Top Node** + and then select **Insert Top Node**.
4. Select the viewpoint for the node that you want to insert. Viewpoints are filtered to:
   - The current viewpoint if the shared nodes option is enabled.
   - Viewpoints that share the same hierarchy set as the current viewpoint and have shared nodes enabled.
   - Viewpoints in the view that use the same node type as the current view or have source node types that can be converted to node types for the current view.

   > **Tip:**
   >
   > If no viewpoints are available:
   >
   > - To insert a node from the same viewpoint, make sure that shared nodes is enabled for the viewpoint dimension.
   > - To insert a node from a different viewpoint in the view, make sure that the other viewpoint uses the same node type or has a node type converter set up. See Working with Node Type Converters.

5. In the node selector, select the nodes to insert. You can type the name of a node that you want to insert in the search bar and hit **Enter** to search for a node directly, or you can click ▶ and drill to the nodes that you want to insert.
   - **When inserting nodes from the same viewpoint or a viewpoint with the same node type**: You can select multiple nodes to insert. Nodes that you select are indicated by a check mark and added to the **Selected** tab. Click the **X** to remove a node from the Selected tab.
   - **When inserting nodes from a source viewpoint with different node types than the target**: You can either select multiple nodes to insert by selecting nodes and adding them to the **Selected** tab or you can select a single node and any related nodes by clicking the **Node menu** ⋮ next to the node and then selecting the type of insert:
     - **Node Only**
     - **Node and Children**

- **Node and Descendants**
- **Bottom Nodes**

> **Note:**
>
> The **Node menu** ⋮ is available only when a single node is selected. If you select an additional node, the menu is no longer available.

6. **Optional**: By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** ☰ to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.
7. When you have finished selecting top nodes, click **OK** to close the node selector.
8. Continue to make changes, or click **Done** to close the request.

Updating Node Properties

You can update node properties in a viewpoint. When you edit properties, you are doing so for the application that you're working in. Property changes in one application do not affect other applications.

> **Note:**
>
> Property changes that you make on a shared node affect that shared node in all of its locations in the viewpoint.

For more information, see Working with Node Properties.

To update node properties:

1. Open a draft request or create a new request.
2. Select a node to update.
   Properties of the selected node are displayed in edit mode in the Property pane.
3. Enter new property values as required. If you are updating a Node data type property, you can select from a single viewpoint that uses the node set configured for the property.

   > **Tip:**
   >
   > By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** ☰ to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.

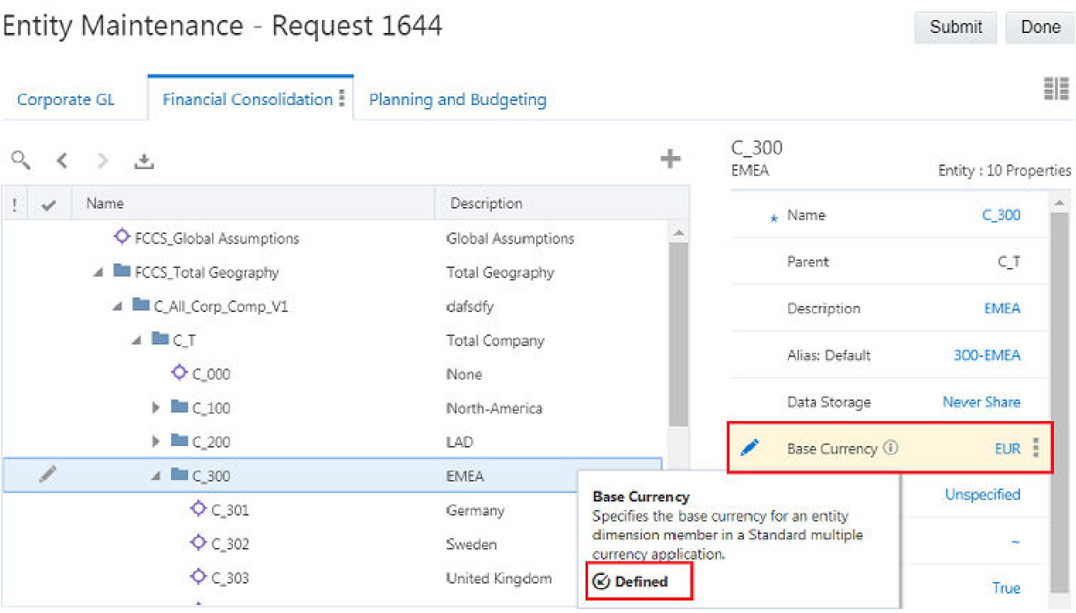
4. Click **Done**.

Note that in some circumstances, updating a property may not be permitted but the clear option for the property is still available.

Property Editable Scenario	Property in UI Updates	Clear option in UI	File Request Updates	File Request Clears
If Property Editable (viewpoint) is False	Read only	Not Available	Skipped	Skipped
If Property Editable (viewpoint) is True and Property Editable (application override) is False	Read only	Available	Skipped	Valid
If Property Editable (viewpoint) is True and Property Editable (node type override) is False	Validation error	Available	Validation Error	Valid

Using Inherited Properties

1. Create a request.
2. Define the property value for the parent node.



Entity Maintenance - Request 1644

Name	Description
FCCS_Global Assumptions	Global Assumptions
FCCS_Total Geography	Total Geography
C_All_Corp_Comp_V1	dafsdfy
C_T	Total Company
C_000	None
C_100	North-America
C_200	LAD
C_300	EMEA
C_301	Germany
C_302	Sweden
C_303	United Kingdom

C_300
EMEA
Entity : 10 Properties

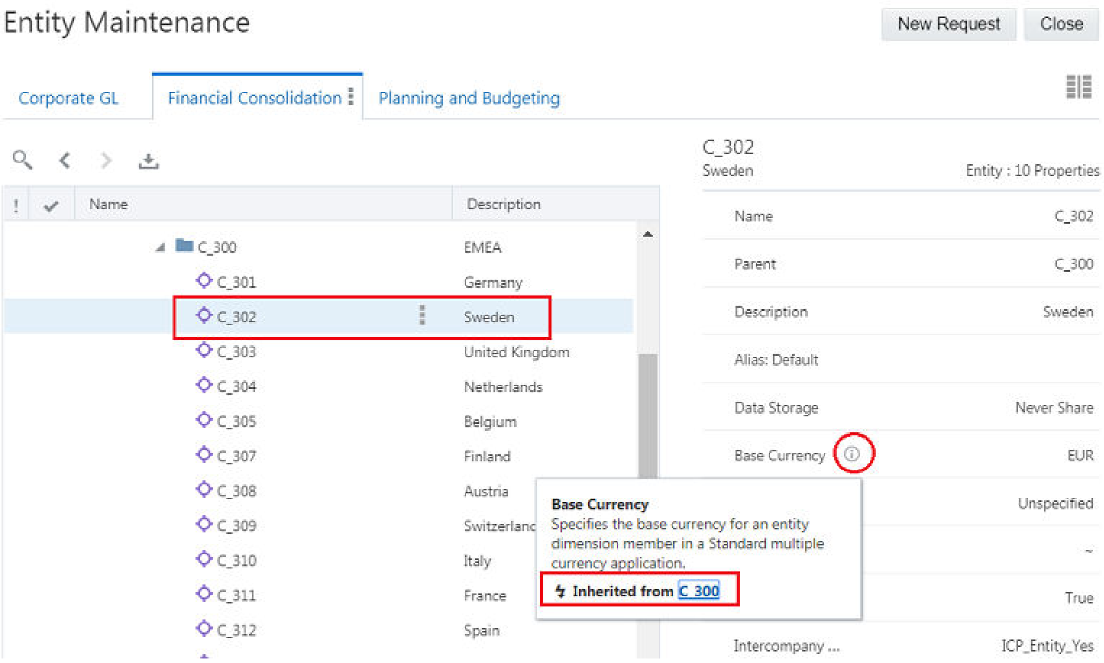
Name	C_300
Parent	C_T
Description	EMEA
Alias: Default	300-EMEA
Data Storage	Never Share
Base Currency	EUR
	Unspecified
	-
	True

Base Currency
Specifies the base currency for an entity dimension member in a Standard multiple currency application.
Defined

> **Note:**
>
> If the descendant nodes already have defined property values, you'll need to clear the property values before you'll see the inherited values. You can clear the property values one at a time or you can clear them using a request load file.

3. Submit the request.

Notice that the value that you set for the ancestor node is now automatically set for its descendants.



Entity Maintenance

!	✓	Name	Description
		C_300	EMEA
		C_301	Germany
		C_302	Sweden
		C_303	United Kingdom
		C_304	Netherlands
		C_305	Belgium
		C_307	Finland
		C_308	Austria
		C_309	Switzerlanc
		C_310	Italy
		C_311	France
		C_312	Spain

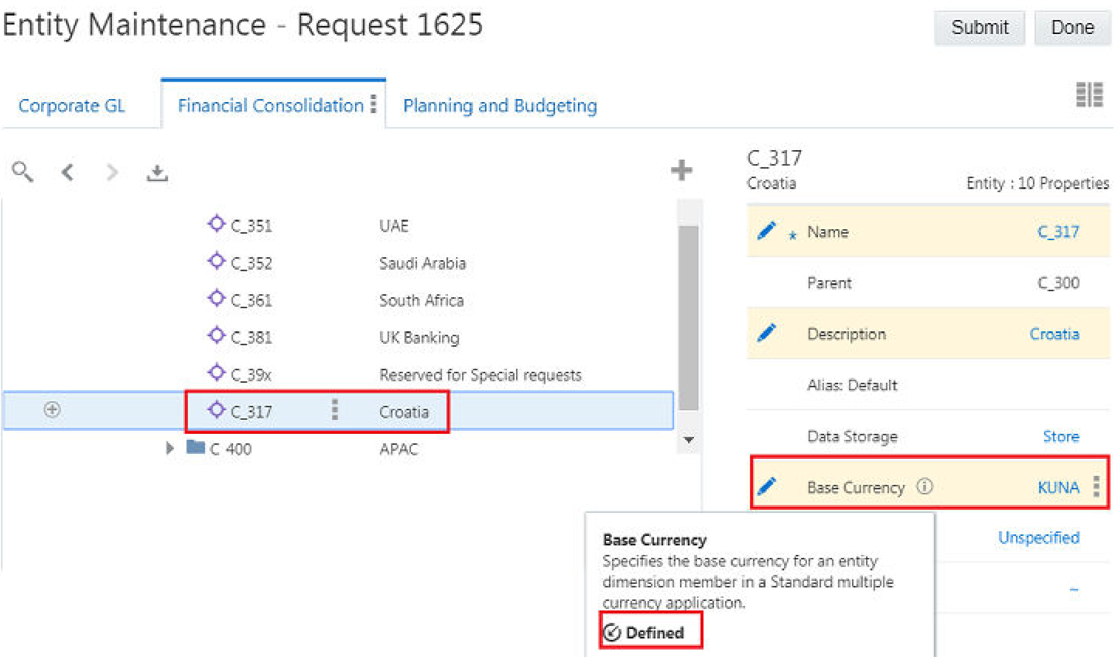
C_302
Sweden
Entity : 10 Properties

Name	C_302
Parent	C_300
Description	Sweden
Alias: Default	
Data Storage	Never Share
Base Currency	EUR
	Unspecified
	~
	True
Intercompany ...	ICP_Entity_Yes

Base Currency
Specifies the base currency for an entity dimension member in a Standard multiple currency application.
Inherited from C_300

Overriding an Inherited Property Value

If a descendant node requires a different property value from the one being inherited from the ancestor node, you can change the property value for the node to override the inherited property. The property value changes to a defined value instead of an inherited value.



Entity Maintenance - Request 1625

Name	Description
C_351	UAE
C_352	Saudi Arabia
C_361	South Africa
C_381	UK Banking
C_39x	Reserved for Special requests
C_317	Croatia
C 400	APAC

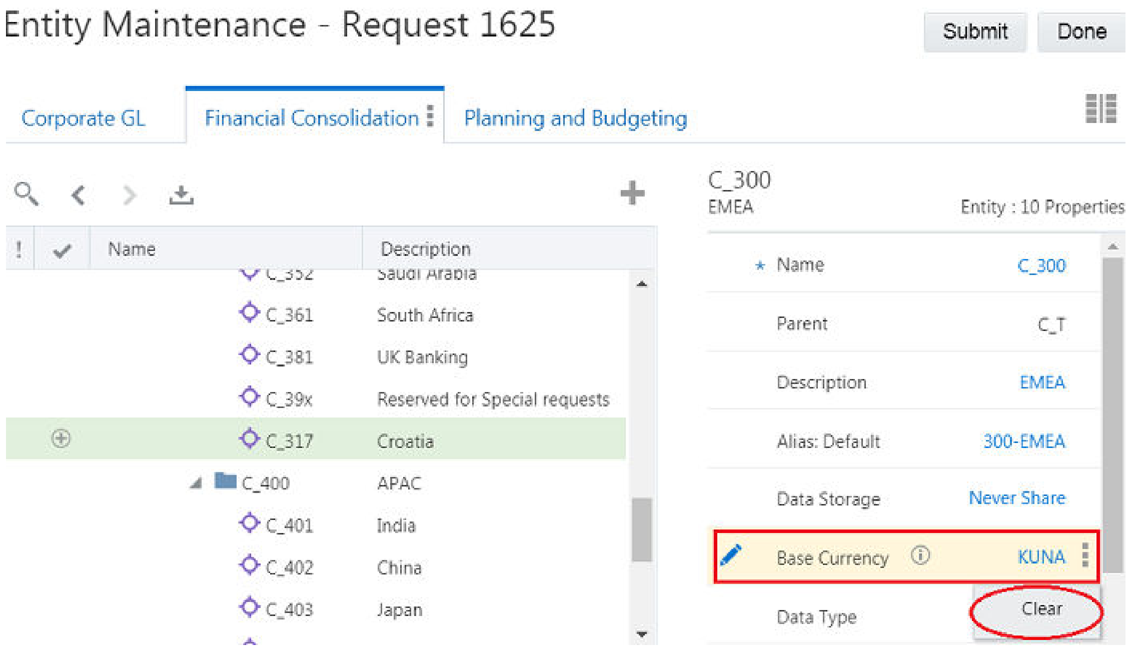
C_317
Croatia
Entity : 10 Properties

Name	C_317
Parent	C_300
Description	Croatia
Alias: Default	
Data Storage	Store
Base Currency	KUNA
	Unspecified
	~

Base Currency
Specifies the base currency for an entity dimension member in a Standard multiple currency application.
Defined

Clearing a Property Value

You can clear the property value for a defined property. Clearing the property value deletes the stored value and returns the value to the default, inherited, or derived value.



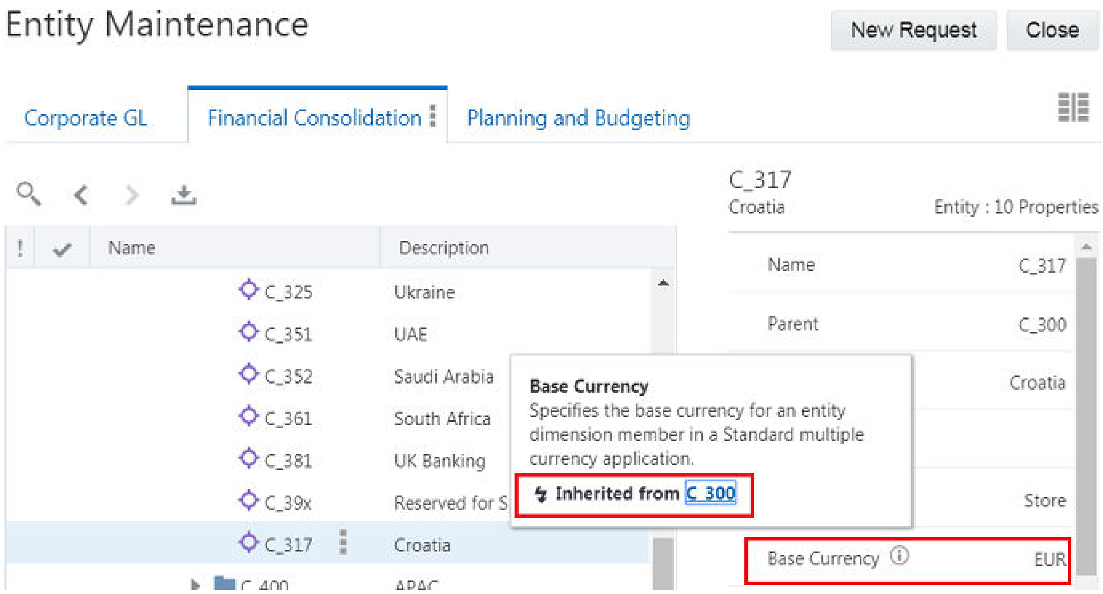
Entity Maintenance - Request 1625

Name	Description
C_352	Saudi Arabia
C_361	South Africa
C_381	UK Banking
C_39x	Reserved for Special requests
C_317	Croatia
C_400	APAC
C_401	India
C_402	China
C_403	Japan

C_300
EMEA
Entity : 10 Properties

Name	C_300
Parent	C_T
Description	EMEA
Alias: Default	300-EMEA
Data Storage	Never Share
Base Currency	KUNA
Data Type	

As soon as you clear the property, the default, inherited, or derived value displays.



Entity Maintenance

Name	Description
C_325	Ukraine
C_351	UAE
C_352	Saudi Arabia
C_361	South Africa
C_381	UK Banking
C_39x	Reserved for S
C_317	Croatia
C_400	APAC

C_317
Croatia
Entity : 10 Properties

Name	C_317
Parent	C_300
	Croatia
	Store
Base Currency	EUR

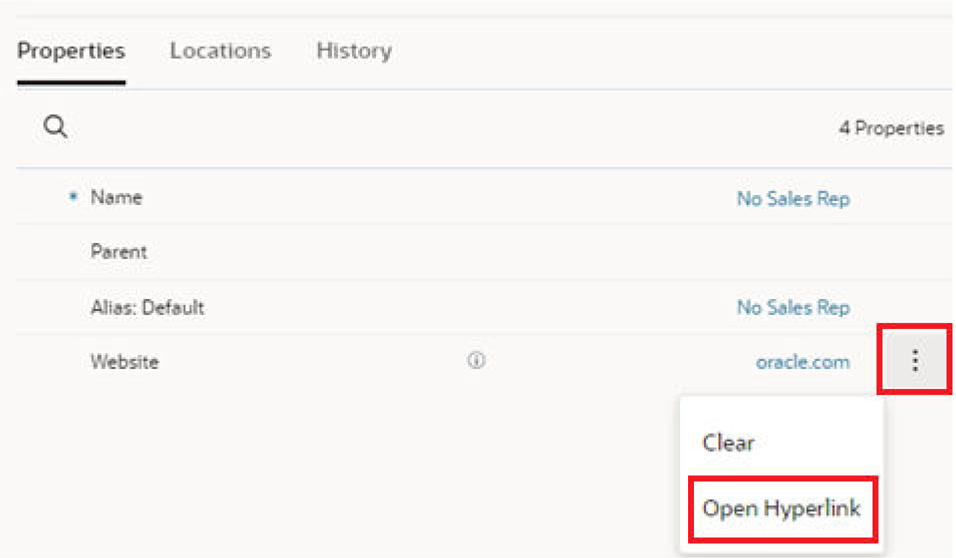
Base Currency
Specifies the base currency for an entity dimension member in a Standard multiple currency application.
Inherited from C_300

Navigating to Hyperlinks in Properties

You can navigate to the target of the hyperlink property in two ways:

- From the properties pane of a viewpoint, click the hyperlink to navigate to the link target or right-click the link to perform standard link operations, such as Open in New Tab, Save Link As, etc.).
- In the context of a request, the property is displayed in plain text so that it can be edited.

  Click **Actions** ⋮ and select **Open Hyperlink** to navigate to the link target.



* Name	No Sales Rep
Parent	
Alias: Default	No Sales Rep
Website ⓘ	oracle.com

Inserting Nodes and Related Nodes

Inserted nodes can be from:

- A list node set
- A different hierarchy set
- The same hierarchy set if the hierarchy set allows shared nodes

You perform the insert by using the Insert Here menu option from the target parent or by dragging and dropping the nodes from another viewpoint in side by side layout or from a draft request item.

For example, your company hires five new employees who all report to the same manager. The new employees are working in five different locations. You can add the five new employees to the manager in one viewpoint and then drag and drop each new employee to the appropriate office in a different viewpoint.

When inserting nodes from a hierarchy viewpoint, you can insert a single node or you can insert related nodes:

- Node and children
- Node and descendants
- Bottom nodes

> **Note:**
>
> Related node options are only available if the selected node has descendants in the hierarchy.

Drag and Drop

To insert nodes by drag and drop from another viewpoint:

1. Open a view and click **New Request**.
2. Click **Side by Side Layout** ◫ and select two viewpoints for the side by side layout-one that you want to drag nodes from and one that you want to insert the nodes into.

   > **Note:**
   >
   > If you have opened a viewpoint in a duplicate tab, you can insert nodes under a different parent in the viewpoint by dragging them from one viewpoint tab to the other. See Using Duplicate Viewpoint Tabs.

3. Select the node that you want to insert, and then drag and drop it to the target parent.
4. Select the type of insert and then click **OK**.

> **Tip:**
>
> You can also drag and drop nodes from the node clipboard. See Working with the Node Clipboard.

Insert Here Menu

To insert a node using the **Insert Here** menu option:

1. Open the view where you want to insert a node.
2. Open a draft request or create a new request.
3. Select a node to be the parent of the node you are inserting, click ⋮ to the right of the node name, and then select **Insert Here**.
4. Select the viewpoint for the node that you want to insert. Viewpoints are filtered to:
   - The current viewpoint if the shared nodes option is enabled.
   - Viewpoints that share the same hierarchy set as the current viewpoint and have shared nodes enabled.
   - Viewpoints in the view that use the same node type as the current view or have source node types that can be converted to node types for the current view.

> **Tip:**
>
> If no viewpoints are available:
>
> - To insert a node from the same viewpoint, make sure that shared nodes is enabled for the viewpoint dimension.
> - To insert a node from a different viewpoint in the view, make sure that the other viewpoint uses the same node type or has a node type converter set up. See Working with Node Type Converters.

5. In the node selector, select the nodes to insert. You can type the name of a node that you want to insert in the search bar and hit **Enter** to search for a node directly, or you can click ▶ and drill to the nodes that you want to insert.
   - **When inserting nodes from the same viewpoint or a viewpoint with the same node type**: You can select multiple nodes to insert. Nodes that you select are indicated by a check mark and added to the **Selected** tab. Click the **X** to remove a node from the Selected tab.
   - **When inserting nodes from a source viewpoint with different node types than the target**: You can either select multiple nodes to insert by selecting nodes and adding them to the **Selected** tab or you can select a single node to insert and any related nodes by clicking the **Node menu** ⋮ next to the node and then selecting the type of insert:
     - **Node Only**
     - **Node and Children**
     - **Node and Descendants**
     - **Bottom Nodes**

     > **Note:**
     >
     > The **Node menu** ⋮ is available only when a single node is selected. If you select an additional node, the menu is no longer available.
6. **Optional**: By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** ☰ to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.
7. When you have finished selecting nodes, click **OK** to close the node selector.
8. Click **OK**.

Moving a Node

You can move a node from one parent in a viewpoint to a different parent in the same viewpoint. You can perform the move either by dragging and dropping the node that you want to move or by using the Move Here menu option from the new parent to locate the node and move it.

> **Note:**
>
> You can move nodes in hierarchy viewpoints only.

To move a node using drag and drop:

1. Open the viewpoint where you want to move a node.
2. Open a draft request or create a new request.
3. Select the node that you want to move, and then drag and drop it to a new parent.

> **Note:**
>
> If you have opened a viewpoint in a duplicate tab (see Using Duplicate Viewpoint Tabs):
>
> - If you drag and drop a node under a new parent within the same viewpoint tab, the node is moved.
> - If you drag and drop a node under a new parent from one tab to the other, the node is inserted.

To move a node using the **Move Here** menu option:

1. Open the viewpoint where you want to move a node.
2. Open a draft request or create a new request.
3. Select a node which will be the target parent of the node to be moved, then click ⋮ to the right of the node name, and then select **Move Here**.
4. In the node selector, select one or more nodes to move. Perform an action:
   - Type the name of a node that you want to move in the search bar and hit Enter to search for a node directly.
   - Click ▶ and drill to the node that you want to move.

   Nodes that you select are indicated by a check mark and added to the Selected tab. Click the **X** to remove a node from the Selected tab.

   > **Note:**
   >
   > By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** ☰ to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.

5. When you have finished selecting nodes, click **OK** to close the node selector.

6. Click **Done**.

Reordering a Node

You can change the order of nodes under a parent based on the sorting configuration on the hierarchy set (see Creating Hierarchy Sets). If reordering is enabled for parent or bottom nodes, nodes can be reordered by dragging and dropping the node, using the Reorder menu option to move the node up or down one position from its current location, or using the Reorder action code in a request load file. For example, if your external application requires a specific order of nodes for calculations or shared members, you can use this feature to ensure the correct order.

There are three settings on the hierarchy set (see Creating Hierarchy Sets) that determine the sort order for parent and bottom nodes:

- **Use Custom Order**: Enables you to reorder nodes in the hierarchy set.
- **Group Parent Nodes First**: Places parent nodes at the beginning of the siblings when there is a mix of parent and bottom nodes.
- **Sort Bottom Nodes By**: Enables you to select the sorting order (Alphanumeric or Custom) for the bottom nodes under the parents if **Group Parent Nodes First** is enabled.

> **Note:**
>
> The sorting configuration on the hierarchy set determines if parent nodes, bottom nodes, or both can be reordered. For example, if **Use Custom Order** and **Group Parent Nodes First** are both enabled but **Sort Bottom Nodes By** is set to Alphanumeric, you can reorder parent nodes but not bottom nodes.

Considerations

- You can reorder nodes in hierarchy viewpoints only.
- When you interactively move or insert a node in a viewpoint that allows reordering, the system places the node at the bottom of the parent's child list. You must manually reorder the node after moving or inserting it into the viewpoint.
- When you register a Planning application, nodes and hierarchical relationships are displayed in the order they were in the application and reordering of nodes is enabled by default.

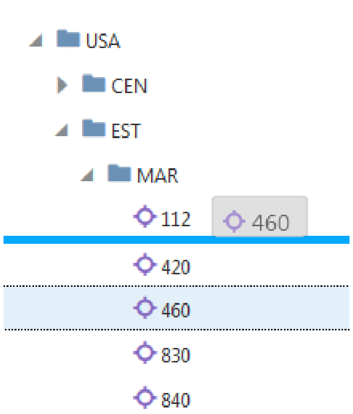
To configure the ability to reorder nodes:

1. Enable custom sort order and configure the sorting options for parent and bottom nodes on the hierarchy set used by the viewpoint.
   a. Inspect the hierarchy set, see Inspecting a Hierarchy Set.
   b. On the **Definition** tab, click **Edit**, then use the **Use Custom Order**, **Group Parent Nodes First**, and **Sort Bottom Nodes By** fields to establish the sort order for parent and bottom nodes (see Creating Hierarchy Sets), and then click **Save**.
2. Enable reordering of nodes as an allowed action for the viewpoint.
   a. Inspect the viewpoint, see Inspecting a Viewpoint.
   b. On the **Definition** tab, click **Edit**, select **Reorder**, and then click **Save**.

To reorder a node using drag and drop:

1. Open the viewpoint where you want to reorder a node.
2. Open a draft request or create a new request.
3. Select the node that you want to move, and then drag and drop it to a new location under the same parent.

> **Note:**
>
> Make sure that you drop the node between nodes in the hierarchy. If you drop the node on top of a node, it may become a child of that node. In this example, we're reordering node 460 so that it is between nodes 112 and 420.



To reorder a node using the **Reorder** menu option:

1. Open the viewpoint where you want to reorder a node.
2. Open a draft request or create a new request.
3. Select the node that you want to move, click ⋮ to the right of the node name, select **Reorder**, and then select **Up** or **Down**.

> **Note:**
>
> The node moves up or down one position from its current location under the parent.

Removing a Node

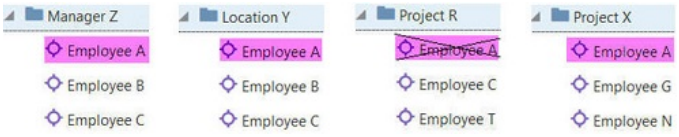
You can remove nodes from viewpoints that use a hierarchy structure.

> **Note:**
>
> The remove option is not available for a viewpoint that uses a list.

When you remove a node, the parent/child relationship is removed from the hierarchy set for the viewpoint. If the node has other parents in the hierarchy set (such as a shared node), those relationships are not affected, and the node remains in the hierarchy set in those relationships.

If the node does not have any other parents, the node no longer exists in that hierarchy set, but still exists in the node type and may exist in other hierarchy sets.

Example



> **Note:**
>
> To delete a node from all parents in all hierarchies or from a viewpoint that uses a list, see Deleting a Node. For example, if Employee A no longer works for the company, you can delete Employee A, and the node is deleted from all hierarchies.

To remove a node:

1. Open the view where you want to remove the node.
2. Open a draft request or create a new request.
3. Find the node that you want to remove, then click ⋮ to the right of the node name, and the select **Remove**.
4. Click **Done**.

Deleting a Node

When you delete a node, it is deleted from all parents in all hierarchies in an application. You can also delete nodes from viewpoints that use lists. In both cases, the node is also deleted from the node type for the application.

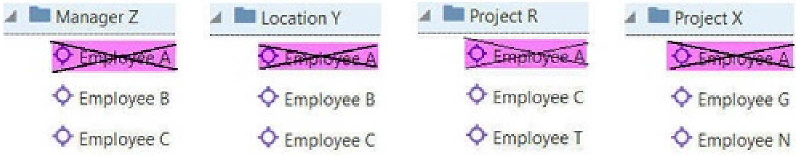
Considerations

- When you delete a node from a hierarchy viewpoint, two request actions are added to the request: a Delete action for the node type, and a Remove action for the hierarchy set.
- Deleting a parent in a hierarchy does not delete all of its children. Instead, it removes the relationships between the deleted parent and its children from the hierarchy set. This allows the children (and their descendants) to be reinserted into the hierarchy at a later time if desired.

  If you want to delete a node and all of its children, you must enter separate delete actions for each node to be deleted.
- To remove a specific node from a parent in the hierarchy set but not affect other relationships that use the node, see Removing a Node.

Example

Employee A is a node in multiple hierarchies as shown below. Employee A has left the company and you need to delete the employee from all hierarchies.



To delete a node:

1. Open the view where you want to delete a node.
2. Open a draft request or create a new request.
3. Find the node that you want to delete, then click ⋮ to the right of the node name, and then select **Delete**.

Working with the Node Clipboard

The node clipboard provides a convenient way to construct a list of nodes that you can add, insert, or move into one or more viewpoints.

After you enable the node clipboard, use the checkboxes next to nodes in the viewpoint to select nodes from hierarchies or lists and place them on a central clipboard. Then, from the clipboard you can select sets of nodes to drag and drop to perform add or insert operations.

Placing and Removing Nodes from the Node Clipboard

Considerations for Placing and Removing Nodes

- You can select nodes from multiple viewpoints to place on the node clipboard. Use the viewpoint selector on the node clipboard to switch your view to the other viewpoints that you have placed nodes from.
- Nodes on the clipboard are grouped by node type within a viewpoint.
- When you place a node on the clipboard it is automatically selected. Click off of the node on the clipboard to select a different node and deselect the one that you just added.
- Nodes on the clipboard persist if you navigate away from the clipboard (for example, by navigating to a request). However, they do not persist on the clipboard if you disable and then enable the clipboard again or if you close and reopen the view.
- The nodes on the clipboard are node instances. That means that if shared nodes are enabled, you can place multiple instances of the shared node to the clipboard. The node parent is displayed under the node name on the clipboard to help you identify the node instance.

Placing Nodes on the Node Clipboard

1. From a viewpoint, click the **Node Clipboard** 📋 tab on the left side of the page.
2. Click **Enable Clipboard**.
3. From the viewpoint, select a checkbox next to a node to place it on the clipboard.

Removing Nodes from the Node Clipboard

You can remove nodes from the node clipboard in several ways:

- Deselect the checkbox next to the node in the viewpoint

- From the node clipboard, click the **Action Menu** ⋮ next to the node that you want to remove and then select **Remove**.
- From the node clipboard, click **Remove Selected** to remove all selected nodes from the clipboard.

> **Note:**
>
> All nodes are removed for the current viewpoint only.

Selecting and Working with Nodes on the Clipboard

Considerations for Working with Nodes on the Clipboard

- When dragging multiple nodes from the clipboard, the mouse cursor displays the number of nodes being dragged. If you attempt to drop your selected nodes in an invalid location, the mouse cursor displays an Unavailable icon.
- When dragging multiple nodes from the clipboard to insert in a viewpoint, if any of the nodes are not able to be inserted (for example, if a node type converter does not exist for one of the selected nodes) then none of the nodes are inserted.
- If you drag and drop a single node and insert it into a different viewpoint (using a different hierarchy set or node type), you are prompted to select whether you want to also insert its related nodes (children, descendants, or bottom nodes).
- If you drag and drop multiple nodes and insert them into a different viewpoint (using a different hierarchy set or node type), only the nodes themselves are inserted. You are not prompted to select related nodes for each of the nodes that you selected.
- To locate a node on the clipboard in a viewpoint, from the clipboard click the **Action Menu** ⋮ next to the node that you want to locate and then select **Locate**.
- After you add, insert, or move nodes from the clipboard, you can change their order in the viewpoint using the **Reorder** menu option or dragging and dropping them in the hierarchy. See Reordering a Node.

> **Tip:**
>
> To insert multiple nodes at the same time using a keyboard only (without dragging and dropping), access the node selector by using the **Insert Here** menu option. See Inserting Nodes and Related Nodes.

Selecting and Deselecting Nodes on the Clipboard

After nodes have been placed on the clipboard, you can select and deselect them in multiple ways. Selected nodes are highlighted on the clipboard, and all selected nodes are dragged when you drag from the clipboard.

- Click on a node to select or deselect it. You can use **Ctrl/Cmd + Click** or **Shift + Click** to select or deselect multiple nodes.
- From the clipboard, click **Select All** to select all nodes, and click **Deselect All** to deselect all nodes.

Working with Nodes on the Clipboard

After you have selected one more nodes from the clipboard, drag and drop the nodes to add, insert, or move them into a viewpoint. The following table describes the behavior when you drag and drop nodes in a viewpoint.

Table 10-1 Drag and Drop Results

Source Viewpoint	Drag Operation	Allow Drop	Drop Result
Same as target viewpoint	A single node from a hierarchy in a viewpoint (not the node clipboard)	You can drop to any node except as the selected node's own descendant	• **Shared Nodes enabled**: You are prompted whether you want to Insert (a new shared instance is inserted) or Move (the node is moved) the node. • **Shared Nodes disabled**: The node is moved. If the target is a sibling and the viewpoint supports a custom sort order, the nodes are reordered.
	A single node from the clipboard	You can drop to any node except a sibling or a descendant of the selected node.	• **Shared Nodes enabled**: You are prompted whether you want to Insert (a new shared instance is inserted) or Move (the node is moved) the node. • **Shared Nodes disabled**: The node is moved.
	Multiple nodes from the clipboard	You can drop to any node except siblings of any of the selected nodes.	• **Shared Nodes enabled**: You are prompted whether you want to Insert (new shared instances are inserted) or Move (the nodes are moved) the node. • **Shared Nodes disabled**: The nodes are moved.

Table 10-1 (Cont.) Drag and Drop Results

Source Viewpoint	Drag Operation	Allow Drop	Drop Result
Different from target viewpoint (different hierarchy set or node type)	A single node from a hierarchy in a viewpoint (not the node clipboard)	You can drop to any node. A node type converter must exist if you are dropping to a different node type.	You are prompted to select the related nodes options (Node only, Node and Children, Node and Descendants, or Bottom Nodes) After you make your selection, new shared instances are inserted if shared nodes are allowed. If shared nodes are not allowed, the node in the target is moved.
	A single node from the clipboard	You can drop to any node. A node type converter must exist if you are dropping to a different node type.	You are prompted to select the related nodes options (Node only, Node and Children, Node and Descendants, or Bottom Nodes) After you make your selection, new shared instances are inserted if shared nodes are allowed. If shared nodes are not allowed, the node in the target is moved.
	Multiple nodes from the clipboard	You can drop to any node. Node type converters must exist for all selected nodes.	New shared instances are inserted if shared nodes are allowed. If shared nodes are not allowed, the nodes in the target is moved. You are not prompted to select related nodes for the nodes that you have selected.

After you drop the selected nodes, the number of request items that were created from the drop action is displayed next to the **Request** tab icon. Navigate to the Request tab to view the request items that were created for the added or inserted nodes.

Making Changes Using a Load File

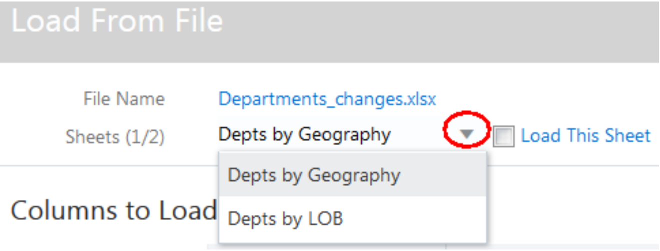
You can use Microsoft Excel spreadsheets to import actions into a request.

For information on setting up a load file, see Request Load File Format.

The permissions of the user uploading the request file are taken into account when processing the file. See Security for Requests.

To load request items from a file:

1. Open the view that you want to make changes to.
2. Open a draft request or create a new request.
3. Click ⋮ to the right of the request title, and then select **Load Request Items**.
4. Browse to select the file, or drop the file into the window.
   The first 10 rows in the first worksheet are displayed. To preview other worksheets, from **Sheets**, select a different worksheet.
5. **Optional**: By default, all worksheets in the file are loaded. For any sheet that you do not want to load, from **Sheets** select the worksheet and then clear **Load This Sheet**.



Load From File

File Name: Departments_changes.xlsx

Sheets (1/2): Depts by Geography ▾ □ Load This Sheet

6. Click **Load**.
   A summary of the results of the file load is displayed. You can see the total number of rows that were loaded, the number of rows that were successfully processed, and the number of rows that were skipped. Click **Close** to return to the viewpoint.

After the file is loaded, it is added as an attachment to the request. Two new columns are added to the file:

- Status—Displays load status for the row, such as Success or Skipped.
- Message—Provides additional information for the row; for example, providing the reason why a row was skipped: "Row skipped because the node type is not specified or not valid."

If the same file name is loaded more than once, the existing attachment is replaced with results from current file.

Request Load File Format

This topic describes the format of Microsoft Excel spreadsheets used to load request items.

Considerations

- Each worksheet in a spreadsheet file corresponds to a viewpoint.
- You can load multiple worksheets to multiple viewpoints.
- Worksheets (and their corresponding viewpoints) must use labels with 30 or fewer characters.

> **Note:**
>
> If a viewpoint name is longer than 30 characters, you'll need to create a label which will be used for the worksheet when loading request items to a viewpoint. See Inspecting a Viewpoint for information on using a label.

- Spreadsheets can contain formulas to derive property values for the standard columns. Spreadsheet formulas can also be used for application-specific properties (FCGL, PLN) and custom properties. Property cells with formula values that equal blank are skipped. Property cells with formula values that equal <clear> or <blank>.
- Worksheets are processed in left to right order.
- Duplicate rows in the import file are not processed during the load.
- Dates and timestamps for supported locales must be in one of these Java date format patterns:
  - Short
  - Default
  - Long
  - Medium

> **Note:**
>
> For information on Java date and time formats, see Using Predefined Formats.

- If you have more than 10,000 nodes, create multiple load files each containing less than 10,000 nodes.
- For nodes that are added or inserted in a hierarchy viewpoint where the parent is not known, use the `<Unknown>` keyword in the Parent column to have the value of the parent calculated and stored for each request item. An administrator must have defined an expression to calculate the parent value. See Calculating and Storing the Parent of a Node.
- To calculate the names of nodes that are being added or inserted, leave the name column blank or use the `<cn>` or `<cn ####>` keywords. The node must be set up to calculate the name value. See Calculating and Storing the Name of a Node.

Considerations for Columns

- The spreadsheet must have column headers that match either the reserved column names or the names of properties, see Reserved Column Names.

> **Note:**
>
> You cannot load a sheet if the Name column is missing.

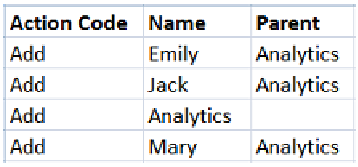
- Column headers for properties must consist of the property name or the property's custom displayed label if one has been set up (this option is set in the Viewpoint inspector Properties tab). The property name cannot include the namespace. For example, `Core.Description` is not valid.

- The columns can be in any order.
- If a column's header does not match any of the reserved column names or property names, the column is ignored. This means that the spreadsheet can include comments that will not be loaded.
- If the load file has a column for the Alternate Name property:
  - If the node type being loaded to contains the Alternate Name property, when loading the file the following operations are performed:
    - If a row contains a node name, the alternate name is loaded in the Alternate Name column.
    - If a row does not contain a node name, the Alternate Name is used to find a matching node in the viewpoint with the same Alternate Name property value.
  - If the node type being loaded to does not contain the Alternate Name property, the column is ignored.

Processing Order of Rows

When loading records from a request file, the load process sorts the file contents so that parent nodes are added to a viewpoint before child nodes. This prevents validation errors if a child node's row precedes its parent node's row in the load file.

For example, this load file contains a parent row, `Analytics`, that is preceded in the file by two child rows, `Emily` and `Jack`, that both have `Analytics` as their parent:



Action Code	Name	Parent
Add	Emily	Analytics
Add	Jack	Analytics
Add	Analytics	
Add	Mary	Analytics

When this file is loaded, the load process sorts the contents so that `Analytics` is added first, then `Emily`, `Jack`, and `Mary`.

<Blank> and <Clear> Processing Versus Empty Fields

Entering the `<blank>` or `<clear>` keywords and leaving cells empty in the request load file are handled differently when processing a request file:

- The `<clear>` keyword clears the existing values from a property.
- The `<blank>` keyword sets a defined value of blank for the property.
- Empty cells in the file are ignored.

The `<blank>` and `<clear>` keywords and empty cells are supported for all data types. However, for list data types, blank entries *within* a list are not supported. That is, the list itself can be empty, but an entry within the list cannot.

Some examples:

- List value: `'<blank>'` is supported. Note that importing a blank for a list will override any existing defined, inherited, or default values in the list.
- List value: `'A,B,C'` is supported.
- List value: `'A,,C'` is not supported.

Performing Copy and Model After Operations in a Request Load File

Use the Copy Node and Copy Action columns to create a new node based on a copy of an existing node or a model after (in a hierarchy viewpoint) of an existing node using a request load file. (See Adding a Node by Modeling After an Existing Node for more details about the model after operation.)

Considerations

- The Copy Node and Copy Action columns are used in Add or Update operations only. If the Action Code column contains any other operation, the Copy Node and Copy Action columns are ignored.
- If the Copy Node column contains a node for which there is already a request item in the load file, the Copy Node and Copy Action columns for that node are ignored.
- If the Core.Name property for the node type of a node being added via a copy or model after operation is configured to be calculated and stored (see Calculated and Stored Properties), the name of the new node is automatically calculated.
- Use the <Ignore> keyword in the parent column to indicate that the parent should be ignored during a model after operation. The new node is inserted only under the parents of the node being copied.

To perform a copy or model after operation in a request load file:

- Use the Copy Node column to specify the node to be copied or modeled after.
- Use the Copy Action column to specify the type of copy operation. Specify one of these values:
  - Properties: Performs a copy operation for the node specified in the Copy Node column.. The copy operation copies defined node and relationship level property values from the specified node and includes them as Update actions for the request item. (This is the default value if you do not specify the type of copy operation.)
  - Model After (hierarchy viewpoints only): Performs a model after operation for the node specified in the Copy Node column. The model after operation does the following:
    - Copies parent relationships from the specified node and includes them as Insert actions for the request item
    - Copies defined relationship level property values for each parent and includes them as Update actions for the request item

> **Note:**
>
> Model After using related viewpoints is not supported during request file uploads. Therefore, you cannot perform model after operations when uploading request files in list viewpoints.

Working with Data Sources in Request Load Files

Best Practice

Although you can add request items from multiple data sources in a single load file, it is a best practice to create a separate load file for each data source. A request item in a load file can only have one data source, so if you load a request file with a node with the same name from multiple data sources only the first data source in the file is retained. Creating separate load files for each data source enables you to run matching for that node for all data sources.

Data Source Column

The request items in the load file are updated with the data source that you specify in the **Data Source** column (see Understanding Data Sources). This enables them to be linked or matched and merged to existing nodes in the viewpoint (see About Node Links and Matching and Merging Request Items).

> **Note:**
>
> The data source must be enabled in order for the request items in the file to be updated.

Considerations

- You can use the data source **Code** or **Name** to identify the data source in the load file. It is a best practice to use the data source code, as the name may change over time.
- You can load nodes from registered data sources only if the load file was originally generated from that registered data source (for example, reloading a generated subscription request file). The rows in the load file must contain the generated Source Node ID and Source Node Type from the registered data sources in order to be loaded.
- If the request item already references a different data source than the one in the row, the row is skipped.

Reserved Column Names

Considerations

- If a property has the same name as that of a reserved column, you can use an alternate column name. For example, if you have a property named `Data Source`, you can use an alternate name such as `Data_Source`, `DataSource`, or `{Data Source}`.
- If the spreadsheet is being loaded into a viewpoint for a list, columns and actions for hierarchies are ignored. For example, Move and Insert actions would be ignored in a list viewpoint.

Table 10-2   Standard Columns

Column Name	Description
`Action Code` **Note**: If the load file does not contain a column with this name or one of the alternative names for the Action Code column, then the action code for all of the request items in the file is set to `Update`.	Specifies how the node will be processed. For example, there are action codes for adding, moving, and deleting nodes. For information on the action codes, see Table 2.

Table 10-2 (Cont.) Standard Columns

Column Name	Description
Copy Action	When the Copy Node column contains a node, this column specifies the type of copy operation to perform. Enter one of the following values: • Properties: Perform a copy operation for the node specified in the Copy Node column.. • Model After (hierarchy viewpoints only): Perform a model after operation for the node specified in the Copy Node column. See Performing Copy and Model After Operations in a Request Load File.
Copy Node	Specifies the name of the node to be copied or modeled after. Use the Copy Action column to specify the type of copy operation. See Performing Copy and Model After Operations in a Request Load File.
Data Source	Specifies the data source for a node. You can use the data source **Code** or **Name**. However, it is a best practice to use the data source code, as the name may change over time.
Description	A description of the node.
Name	The node name. The combination of the node name and node type must be unique. This column is required.
New Name	If the node is being renamed, this cell contains the new node name.
Node Type	The node's node type. If the viewpoint includes only one node type, the column is optional. If the Node Type cell is empty, the load process uses that node type; however, if the viewpoint contains multiple node types, you must specify a node type in each row.
Old Parent Name	If the node is being moved under a different parent, this cell contains the name of the old parent node. If the node being moved exists under only one parent, then this cell can be empty. If the node exists under multiple parent nodes, you must specify the name of the parent node from which the node should be removed.
Old Parent Node Type	If the node is being moved under a different parent, this cell contains the name of the old parent node's node type. If the node being moved exists under only one parent, then this cell can be empty. If the node exists under multiple parent nodes, you must specify the node type of the parent node from which the node should be removed.
Parent	The name of the parent node, if any. If this cell is empty, the node is imported as a top node. **Note**: Use the <Unknown> keyword in the parent column to calculate and store the value for the parent of the request item.
Parent Node Type	The node type of the parent node, if any.
Sibling	The name of a sibling node under the specified parent that the node should be placed after. Or, if the node being ordered is to be the first sibling under the parent, enter <First>. Use with the Reorder, Add, Insert, and Move actions.
Sibling Node Type	The node type of the sibling. Use with the Reorder, Add, Insert, and Move actions.
Source Node ID	The ID of the source node coming from a registered data source. This column should only be used for request file rows generated by a subscription, from compare results, or from copying nodes across viewpoints.

Table 10-2 (Cont.) Standard Columns

Column Name	Description
`Source Node Name`	The name of the source node coming from a registered data source. This column should only be used for request file rows generated by a subscription, from compare results, or from copying nodes across viewpoints.
`Source Node Type`	The ID of the source node type coming from a registered data source. This column should only be used for request file rows generated by a subscription, from compare results, or from copying nodes across viewpoints.

Table 10-3 Action Codes

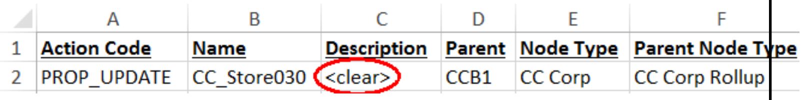
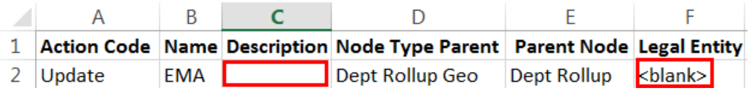
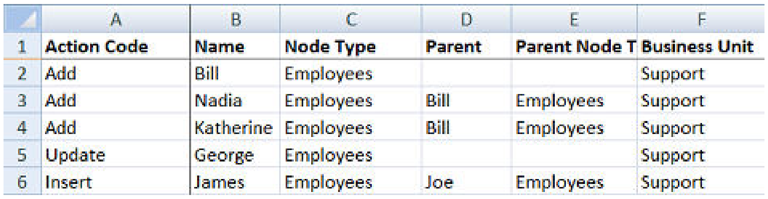
<table>
<tr><th>Action Code</th><th>Description</th></tr>
<tr><td><code>Add</code></td><td>Creates a new node.</td></tr>
<tr><td><code>Delete</code></td><td>Deletes a node.</td></tr>
<tr><td><code><i>&lt;Empty cell&gt;</i></code></td><td>If the <code>Action Code</code> cell is empty, the row is processed using the <code>Update</code> action code.</td></tr>
<tr><td><code>Insert</code></td><td>Inserts an existing node into a hierarchy under the node specified by the <code>Parent</code> and <code>Parent Node Type</code> cells.</td></tr>
<tr><td><code>Move</code></td><td>Moves an existing node to a new position in a hierarchy, removing the node from its existing position.</td></tr>
<tr><td><code>Prop_Update</code></td><td>Updates the property value for an existing node only. To perform other actions, use the <code>Update</code> action code.
You can clear the property value for a defined property. Clearing the property value deletes the stored value and returns the value to the default or inherited value. For example, the Prop_Update action below clears the Description value.

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>1</td><td>Action Code</td><td>Name</td><td>Description</td><td>Parent</td><td>Node Type</td><td>Parent Node Type</td></tr>
<tr><td>2</td><td>PROP_UPDATE</td><td>CC_Store030</td><td>&lt;clear&gt;</td><td>CCB1</td><td>CC Corp</td><td>CC Corp Rollup</td></tr>
</table></td></tr>
<tr><td><code>Rename</code></td><td>Renames an existing node.</td></tr>
<tr><td><code>Reorder</code></td><td>Reorders a node in a hierarchy viewpoint.
You can only reorder nodes in a viewpoint that uses a hierarchy set that allows reordering for that type of node (parent or bottom node). For more information, see Reordering a Node.</td></tr>
<tr><td><code>Remove</code></td><td>Removes the node from the specified parent node.</td></tr>
</table>

Table 10-3 (Cont.) Action Codes

<table>
<tr><th>Action Code</th><th>Description</th></tr>
<tr><td>`Update`</td><td>
Updates an existing node's property values.
The `Update` action also performs additional actions if the cells in a row indicate that such actions are required. The following list describes how this works:
<ul>
<li>If the node does not exist, the node is added. If the viewpoint is hierarchical, the node is inserted in the position of the hierarchy specified by the `Parent` and `Parent Node Type` cells.</li>
<li>If the viewpoint is hierarchical and the node exists but is not under the specified parent, the resulting action depends upon whether the hierarchy set allows shared nodes:
<ul>
<li>If shared nodes are allowed, the node is inserted under the specified parent.</li>
<li>If shared nodes are not allowed, the node is moved under the specified parent.</li>
</ul></li>
</ul>
You can update a property to a blank value if the property allows blanks. You must use the <blank> indicator in the cell; an empty cell does not produce the same results.
For example, the update action below sets the Legal Entity property to a blank value. Even though the cell for description is empty, the existing description for EMA does not change.

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>1</td><td>Action Code</td><td>Name</td><td>Description</td><td>Node Type Parent</td><td>Parent Node</td><td>Legal Entity</td></tr>
<tr><td>2</td><td>Update</td><td>EMA</td><td></td><td>Dept Rollup Geo</td><td>Dept Rollup</td><td>&lt;blank&gt;</td></tr>
</table>
</td></tr>
</table>

Example 10-1 Example

The following example shows a few nodes being added, updated, and inserted into a hierarchical viewpoint:



	A	B	C	D	E	F
1	Action Code	Name	Node Type	Parent	Parent Node T	Business Unit
2	Add	Bill	Employees			Support
3	Add	Nadia	Employees	Bill	Employees	Support
4	Add	Katherine	Employees	Bill	Employees	Support
5	Update	George	Employees			Support
6	Insert	James	Employees	Joe	Employees	Support

The following list describes the example's key points:

- All of the nodes have a node type of Employees.
- The `Business Unit` column indicates that the node type has a property named `Business Unit`.
- The second row adds a node named Bill. The `Parent` and `Parent Node Type` cells are empty, indicating that this is a top node.
- The third and fourth rows add child nodes under the node for Bill.
- The fifth row updates an existing node by setting its `Business Unit` property to a value of Support.

- The sixth row inserts an existing node named James under a parent node named Joe.

Working with Request Items

Request items are groups of change actions for a specific node within a request.

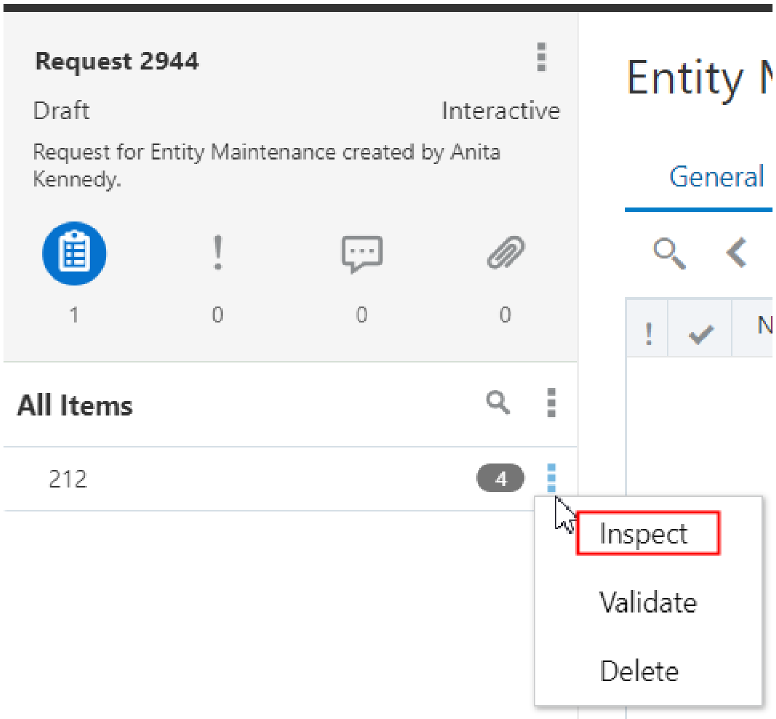
Whenever you make changes to data in a request, those changes to data are organized into request items. You can view these request items when you open the request (see Working with a Request) and when you inspect a request (see Inspecting Requests).

> **Tip:**
>
> From an open request, use the **Filter** ▽ ▾ dropdown to filter the request items to display the following:
>
> - **All Items**: Displays all request items in a request.
> - **Items to Approve** (Requests in the Approve stage only): Displays only the request items that you are invited to approve. See Approving a Request.
> - **Items to Commit** (Requests in the Commit stage only): Displays only the request items that you are invited to commit. See Committing a Request.
> - **Merged Items**: Displays only the request items that were merged with existing nodes as a result of a match and merge operation. See Matching and Merging Request Items.

Inspecting Request Items

From the request pane, you can inspect request items in order to view details about the item.

In the request pane next to a request item, click ⋮ and then click **Inspect** to view details of the request actions for that request item.



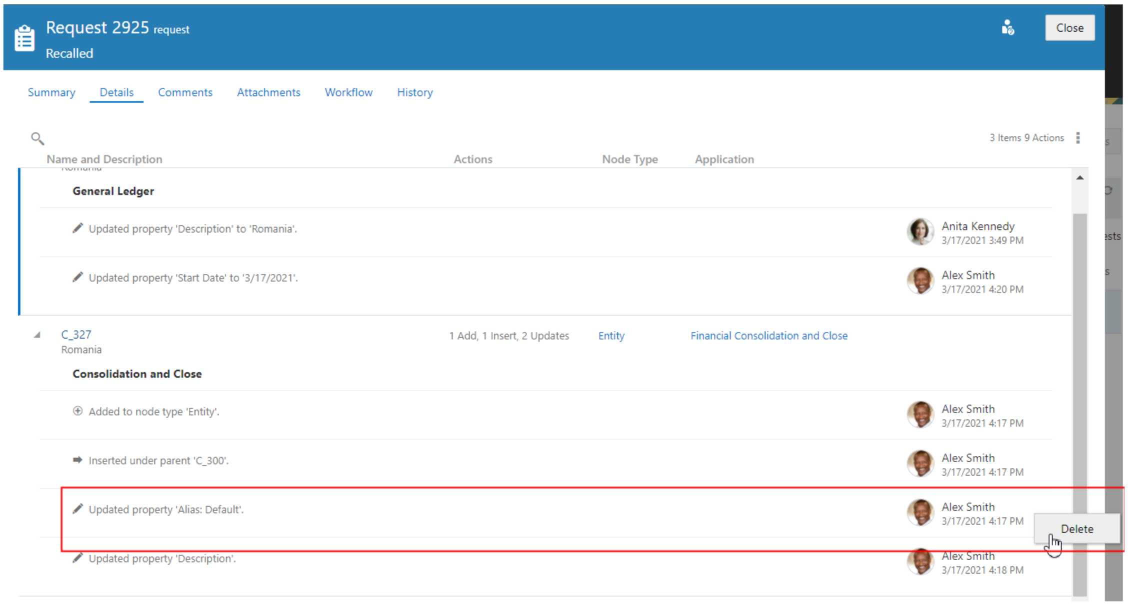
The request is opened in the request inspector, and the Details tab displays the request item that you selected with all of the request actions for that item.

If you meet one of the following criteria, you can delete individual request actions by clicking ⋮, and then selecting **Delete**:

- You are the current assignee of a request in Draft status.
- You are a current collaborator of a request in Draft status, and you have Write access to perform the selected action.
- You are a current invitee of a request in In Flight status, and you have Write access to perform the selected action.

Note:

Request actions that your permissions and data access do not permit you to view are still listed in the request details. You will not be able to see the specifics of the action, but you can still delete it.
For example, in the below image, Anita does not have read access to the **Consolidation and Close** viewpoint where Alex made request changes, and so while she can see that he updated the Alias: Default property, she cannot see the property value. However, because she is the request submitter she can still delete that request action.



Name and Description | Actions | Node Type | Application

General Ledger

Updated property 'Description' to 'Romania'. Anita Kennedy 3/17/2021 3:49 PM

Updated property 'Start Date' to '3/17/2021'. Alex Smith 3/17/2021 4:20 PM

C_327 Romania | 1 Add, 1 Insert, 2 Updates | Entity | Financial Consolidation and Close

Consolidation and Close

Added to node type 'Entity'. Alex Smith 3/17/2021 4:17 PM

Inserted under parent 'C_300'. Alex Smith 3/17/2021 4:17 PM

Updated property 'Alias: Default'. Alex Smith 3/17/2021 4:17 PM

Updated property 'Description'. Alex Smith 3/17/2021 4:18 PM

> **Caution:**
>
> Exercise caution when deleting individual request actions, since actions or property updates may be dependent on a sequence of changes occurring, and that sequence may be disrupted. For example, if you've added parent and child nodes and you delete the action of adding the parent, the child nodes may have validation issues.

When you are finished, click **Close** to close the inspector and return to the request viewpoint.

Attaching a File to a Request

You can attach files to a request in order to justify or provide supporting details about the purpose and intent of the request.

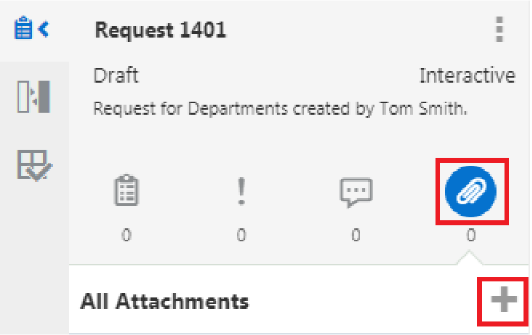
After you attach a file, you can view it on the Attachments tab. Click on an attachment to download it to view its contents. If you attached a file to a request, you can also delete the attachment.

> **Note:**
>
> Attached files that were used to load request items cannot be deleted from a request.

Attaching a File

To attach a file to a request:

1. From an open draft request, click 📎 and then +.



2. Browse to select the file to attach, or drop the file into the window.

> **Note:**
>
> You can replace an existing attachment if you previously attached a file with the same name. If you are trying to attach a file with the same name as an existing attachment that was uploaded by another user, you'll need to specify a different file name.

3. Click **Attach**.

Deleting an Attachment

To delete a request attachment:

1. From an open draft request, click 📎.
2. Locate the attachment that you want to delete, hover the mouse over the attachment and then click ✕ .

> **Note:**
>
> The delete icon is only available for attachments that you added to the request.

3. Click **Yes** to confirm the deletion.

Setting and Viewing Request Priority

Request owners can set the priority of a request to High, Medium, or Low. The request priority can be viewed in request activity, the request inspector, and in an open view.

Considerations

To set the priority of a request, the following must be true:

- You must be the request owner.
- The request must be in the Submit stage. This includes requests that have been pushed back or recalled.

Setting the Request Priority

You set the priority of a request from the request details pane in an open view.

1. Create or open a request that is currently in the Submit stage.
2. In the request details pane, from the request header click **Request Actions** ⋮, and then click **Set Priority**.
3. Select the priority: **Low**, **Medium**, or **High**. You can also select **None** to set the request to unprioritized. The currently set priority is indicated by a check mark.

Viewing the Request Priority

You can view the priority of a request in the following areas:

- **Request details pane in an open view:** The priority (Low, Medium, High) is displayed in the request header. If the priority is not set (None), it is not displayed in the header.
- **Request Activity worklist:** The priority is displayed in the Request Priority column. You can filter on request priority (see Working with Request Activity).
- **Request inspector:** The priority is displayed on the Summary tab.

Working with Future Dated Requests

Future dating a request enables you to assign a time label to the request with a date in the future for that request to be completed and closed.

This enables you to make changes now that won't be applied to the active view until a specified point in the future (for example, you can make requests for changes that won't take effect until the next quarter).

When requests that have been assigned a future time label get submitted they go through the normal approve and commit stages, and then when all of the policies are met they are held until the date and time in the time label is reached (similar to blockout period processing). The requests are then placed in Future Dated status. When the date and time in the time label is reached, the system attempts to complete and close them (see Completing a Future Dated Request, below).

Considerations

- You can assign, modify, or remove a future time label from a request if one of the following is true:
  - The request is in the Submit stage and you are the request owner.
  - The request is in the Approve or Commit stage and you are an invitee for at least one policy that allows a future time label to be assigned. See Creating and Enabling Approval Policies and Creating and Enabling Commit Policies.
- Request with a future time label can only be consolidated with other requests that use the same time label. When you consolidate requests that use the same time label, the consolidation request that gets created is also assigned that time label. You can also assign a future time label to a consolidation request. See Consolidating Requests.
- If the time in a future time label is reached before the request is submitted, you cannot submit the request with that time label. You must remove or change the time label in order to submit it.

- If the time in a future time label is reached after the request is submitted but before it has completed all of its policies, the request is processed as normal upon when the policies have been completed. For example, if you submit a request on a Monday with a future time label of Wednesday but the final policy isn't completed until Friday, when the final policy is completed the request is completed and closed without being put into future dated status. The time label date is still displayed in the request inspector for historical information.

Assigning, Modifying, and Removing a Time Label from a Request

You assign, modify, or remove a future time label from a request from the request details pane in an open view.

1. Create a request or open an existing request for which you are an owner or invitee.
2. In the request details pane, from the request header click **Request Actions** ⋮, and then click **Set Time Label**.
3. Select the time label for the request, or select **None** to remove the current time label from the request. The currently set time label is indicated by a check mark.

> **Note:**
>
> You can select from any valid future time labels that have been defined for the view. If the time in a time label has already been reached, it can no longer be assigned to a request. See Future Time Labels.

Viewing the Time Label for a Request

You can view the time label that has been assigned to a request in the following areas:

- **Requests pane in an open view**: Time labels are displayed under the description of future dated requests. Hover over the time label icon to display the time label name, date, and time. A "Future Dated" status badge is also displayed in the request header.
- **Request details pane in an open view:** When you click on a request in the request pane, the time label is displayed in the request header. Hover over the time label icon to display the time label name, date, and time. A "Future Dated" status badge is also displayed in the request header.
- **Request Activity worklist:** A status of Future Dated and the stage is displayed in the Status and Stage column. The time label name and date is displayed in the Time Label and Date column. You can also filter on Future Dated status in the Status filter (see Working with Request Activity).
- **Request inspector:** On the Summary tab, the request status is displayed as Future Dated, and the time label name, date, and time is displayed in the Time Label field. See Inspecting Requests.

Completing a Future Dated Request

After the time in a future time label on a request is reached, the system attempts to complete and close the request:

1. Requests that have been assigned an expiring time label are processed in the order in which they were submitted.
   - Requests without validation issues are completed and closed.
   - If a request cannot be completed due to validation issues, the owner is notified.

> **Note:**
>
> If the viewpoint is also in a blockout state when the time in the time label has been reached, the request enters Blocked status until the blockout expires. See Requests and Blockout Periods.

2. When the request is completed:
   - The stage and status are updated to Closed and Completed.
   - The time label timestamp is displayed in request activity and on the request inspector.

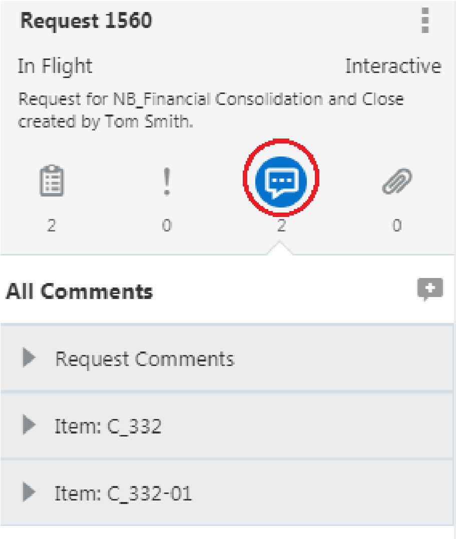
Working with Request Comments

For requests that are in Draft or In Flight status you can add comments to the request and to individual request items. You can reply to comments from other users creating threaded conversations by which you can collaborate on the submission and approval of requests. You can use comments to add summary remarks, explain a specific change, ask questions, and highlight important information about the request or request item.

Users can be mentioned in the request and request item comments providing a way for those who are collaborating on a request to ask questions or make other users aware of what they did to a request. To mention a user, enter the @ symbol into a comment and then select a user to mention. The list of users includes users who have participated in the request or been invited to collaborate on or approve the request. After the comment is saved, the mentioned user is notified of the comment so they can review it and take immediate action on the request.

You can also paste a hyperlink into the comment field or you can use the Link toolbar button to add display text that will make the link easier to read. You can also remove a link.

Your comments can be edited and deleted by you while the request is in Draft status. All comments are displayed on the Comments tab of an open Draft or In Flight request.


Request 1560

In Flight Interactive

Request for NB_Financial Consolidation and Close created by Tom Smith.

For more information, see Adding, Editing, and Deleting Request Comments.

Adding, Editing, and Deleting Request Comments

You can add comments to a request or to individual request items to provide further information. When you enter a comment you have the option to mention a user in the comment and include a hyperlink. You can edit and delete your own comments.

Request Status	Request Comment Actions
Draft	Add and edit comments
In Flight	Add and edit comments
Closed	View comments

Request Comments

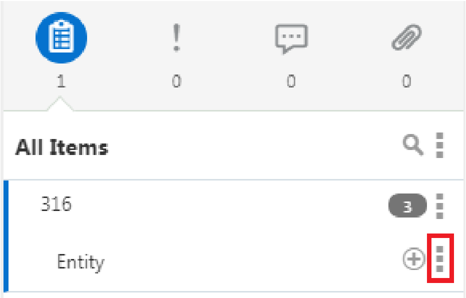
To add, edit, or delete request comments:

1. Open a draft request.
2. Click 💬.
3. Choose an option:
    - **Add a comment** - Click 💬 and enter a comment, and then click **OK**. Click **Save** to add the comment to the request.
        - To mention a specific user in the comment, enter the @ symbol and then select the user's name.
        - To include a hyperlink in the comment, enter or paste the URL directly into the comment field, or click 🔗, enter the text to display for the link, enter or paste the URL.
    - **Edit a comment** - Find your comment, hover the cursor to the right of your name, click ⋮, and then click **Edit**. Make changes, and then click **Save**.
    - **Delete a comment** - Find your comment, hover the cursor to the right of your name, click ⋮, and then click **Delete**.

Request Item Comments

To add a comment to a request item:

1. Click on the request item, click ⋮, and then select **Add a Comment**.



2. Enter your comment and then click **Save**.

> **Note:**
>
> To mention a specific user in the comment, enter the @ symbol and then select the user's name to mention. To include a hyperlink in the comment, enter or paste the URL directly into the comment field, or click 🔗, enter the text to display for the link, enter or paste the URL, and then click **OK**.

To edit or delete a comment in a Draft or In Flight request:

1. Click on the request item.
2. To the right of your name, hover your mouse and click ⋮, and then select **Edit** or **Delete**.

Concurrent Editing of a Request

When multiple users are collaborating on the same request, those users may be working on the request at the same time. Concurrent editing allows users to work simultaneously on the request while ensuring data integrity and alerting users of changes made by others.

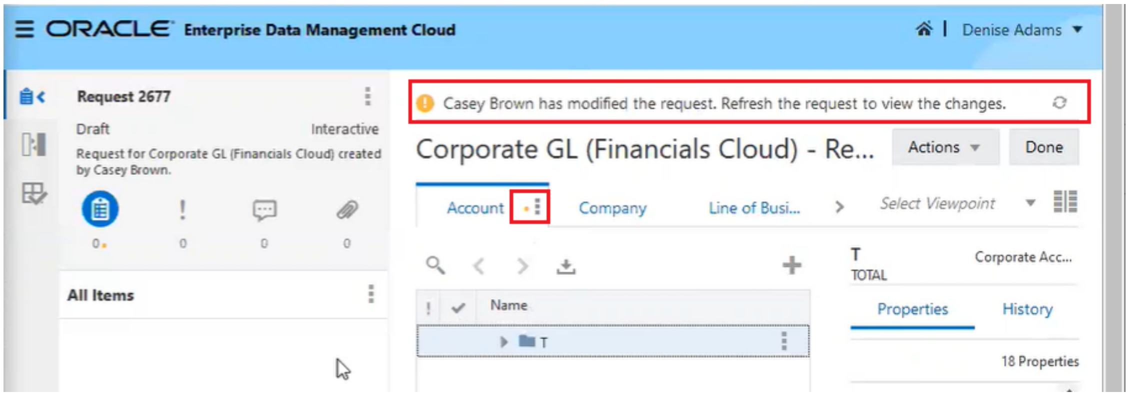
Videos

Your Goal	Watch This Video
Learn about concurrent editing of requests.	▶ Concurrent Request Editing in Enterprise Data Management

Concurrent edits include changes made to request items, comments, and attachments. Concurrent updates are applied to the request in the order in which they happened and are recorded in request history. If there’s a conflicting change being made, the user is notified, and the change will not be applied.

Users who have the request open are alerted of concurrent updates made by other users and can refresh the request to view those modifications. Changes to viewpoints, request items, comments, and attachments are visually indicated.

For example, Casey and Denise working on the same request. After Casey makes a change in the request, Denise receives a message that the request has been modified by Casey. She also sees a visual indicator next to the viewpoint that's been modified.



Casey Brown has modified the request. Refresh the request to view the changes.

Request for Corporate GL (Financials Cloud) created by Casey Brown.

When Denise refreshes the request, she sees the change made by Casey.

Validating Requests

You can validate all items in a request or a single request item.

The system performs the checks necessary to determine the validity of the request items and their request actions compared to other data in the same request and data that already exists in the system. Derived properties for nodes in request items are also validated.

> **Note:**
>
> Validating request items is optional. The system validates the request when you submit it.

Depending on the validation severity, request items that fail a validation check are marked with an error or a warning. Errors must be corrected before the request can be transitioned to the next stage. You can transition requests with warnings to the next stage. For information on resolving validation issues see, Working with Request Validation Issues.

For more information about validations, see Understanding Validations and Constraints.

To validate all items in a request:

1. Open a draft request.
2. Click **Requests**, to the right of **All Items** click ⋮, and then select **Validate**.

To validate a single request item:

1. Open a draft request.
2. Click **Requests**, select a request item, click ⋮ to the right of the item, and then select **Validate**.

Working with Request Validation Issues

In the request panel for a draft request, you can identify request items that have validation issues, and you can take action on the request items to resolve the issues. Validation issues may arise from loading a file or from changes that you are making interactively to the viewpoint.

Depending on the validation severity, request items that fail a validation check are marked with an error or a warning. Errors must be corrected before the request can be transitioned to the next stage. You can transition requests with warnings to the next stage.

Click **Request Issues** ! in an open draft request to view and resolve issues.

Resolving Your Issues

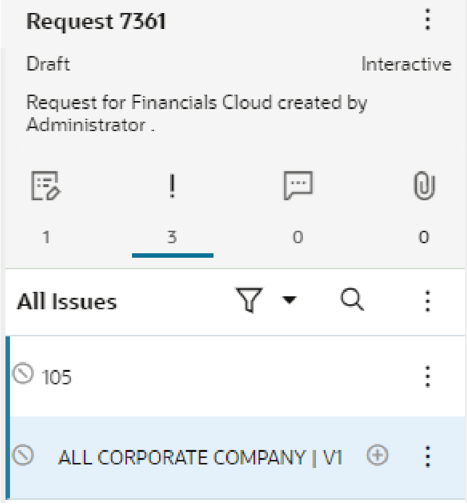
Be default, all request issues are displayed on the issues tab. Click the ▽ button to select a filtering option, as follows:

- **All Issues**: Displays all warnings and errors
- **My Issues**: Displays warnings and errors only on request items that you are responsible for correcting. See **Issue and Error Responsibility**, below.
- **All Errors**: Displays all errors
- **My Errors**: Displays errors only on request items that you are responsible for correcting. See **Issue and Error Responsibility**, below.

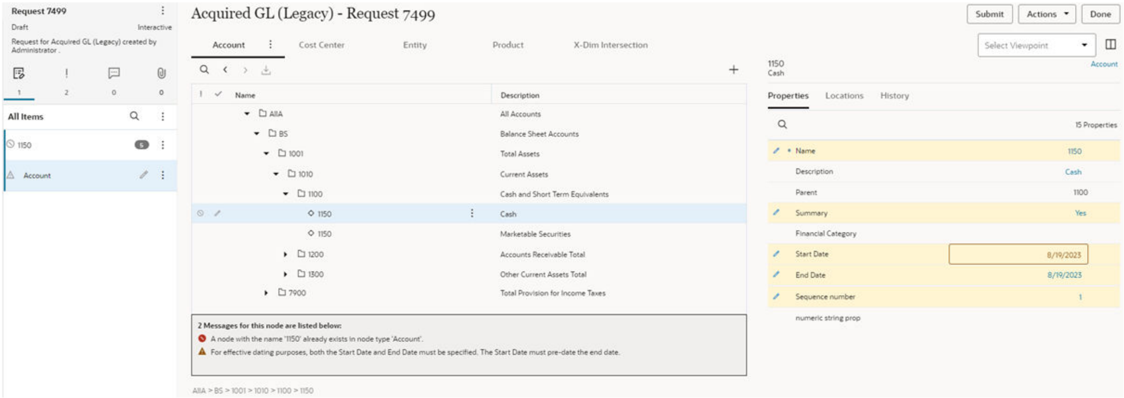
Issue and Error Responsibility

When you select **My Issues** or **My Errors**, the list is filtered to just the warnings or errors that you are responsible for correcting, as follows:

- **If you are a request assignee**, all errors (and warnings, in **My Issues**) are displayed. All validation errors must be corrected before the request can be submitted.
- **If you are a request collaborator or enricher**, these validation warnings and errors are displayed:
  - Validation warnings and errors on request items that you have permission to make changes to
  - Missing required properties that you have access to view


Request 7361
Draft
Interactive
Request for Financials Cloud created by Administrator .
1
3
0
0
All Issues
105
ALL CORPORATE COMPANY | V1

For example, there are two validation issues for this request. The first issue indicates that node 1150 already exists. The second indicates that the End Date must be after the Start Date.



Request for Acquired GL (Legacy) created by Administrator .

Acquired GL (Legacy) - Request 7499

Name	Description
AIIA	All Accounts
BS	Balance Sheet Accounts
1001	Total Assets
1010	Current Assets
1100	Cash and Short Term Equivalents
1150	Cash
1150	Marketable Securities
1200	Accounts Receivable Total
1300	Other Current Assets Total
7900	Total Provision for Income Taxes

2 Messages for this node are listed below:

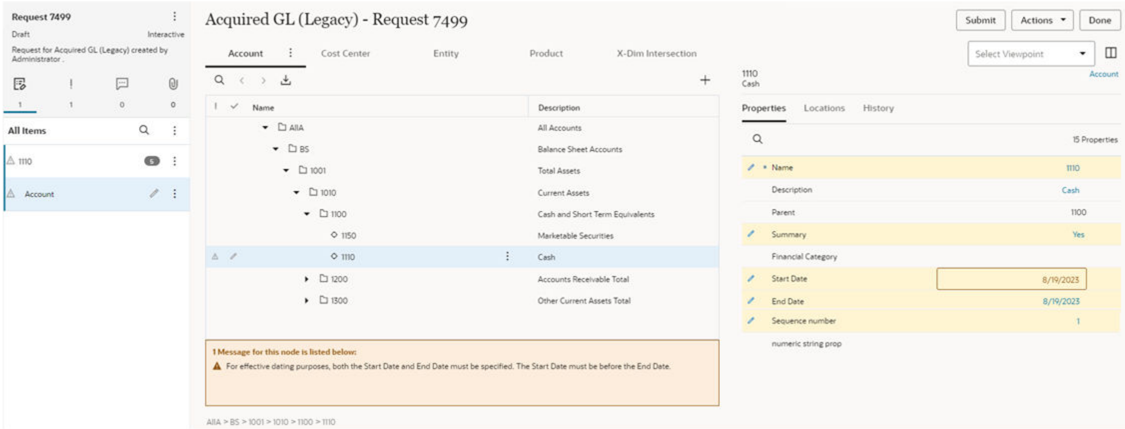
- A node with the name '1150' already exists in node type 'Account'.
- For effective dating purposes, both the Start Date and End Date must be specified. The Start Date must pre-date the end date.

AIIA > BS > 1001 > 1010 > 1100 > 1150

1150
Cash
Account

• Name	1150
Description	Cash
Parent	1100
Summary	Yes
Financial Category	
Start Date	8/19/2023
End Date	8/19/2023
Sequence number	1
numeric string prop	

To correct the first issue, you can change the name of the node in the properties pane. After you do so, the error message is no longer displayed.



Request for Acquired GL (Legacy) created by Administrator .

Acquired GL (Legacy) - Request 7499

Name	Description
AIIA	All Accounts
BS	Balance Sheet Accounts
1001	Total Assets
1010	Current Assets
1100	Cash and Short Term Equivalents
1150	Marketable Securities
1110	Cash
1200	Accounts Receivable Total
1300	Other Current Assets Total

1 Message for this node is listed below:

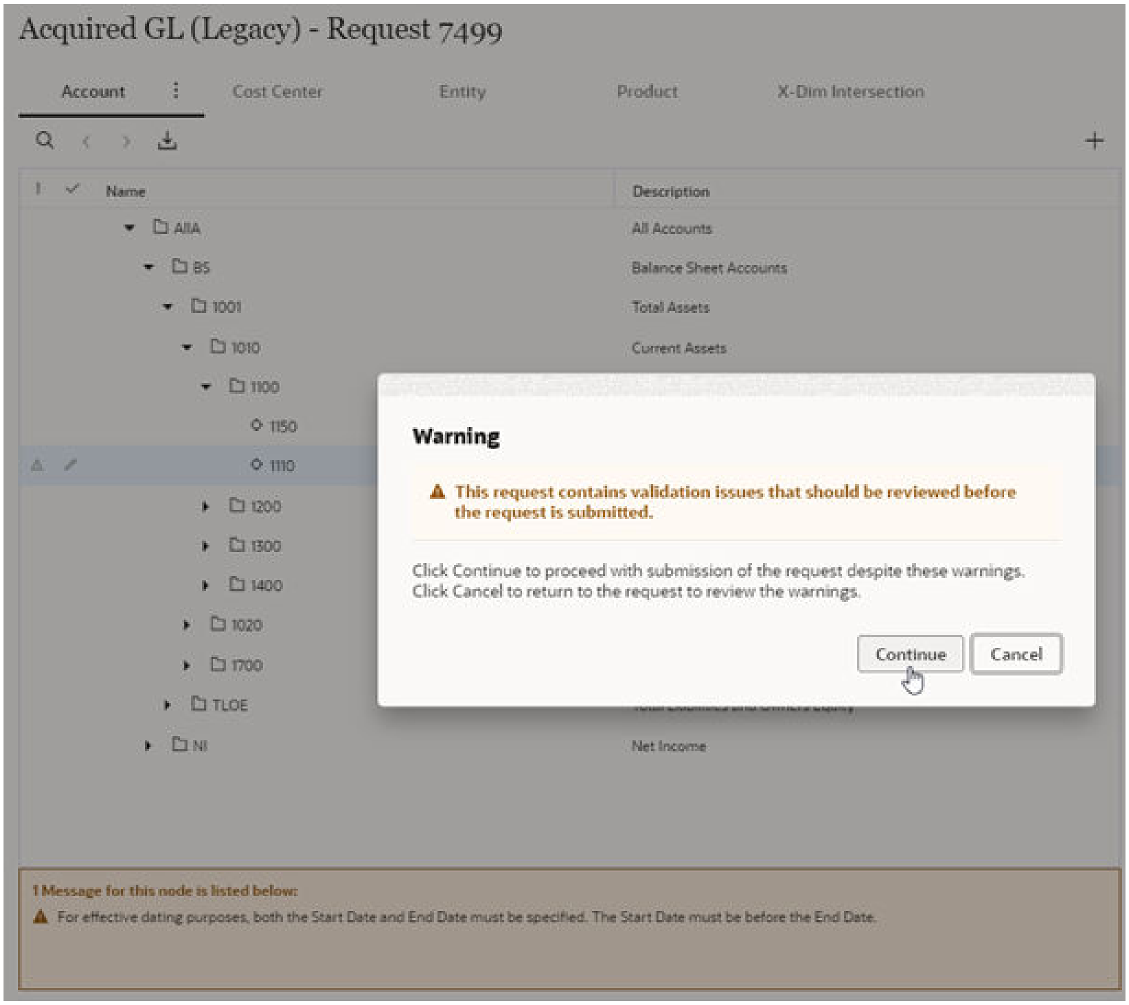
- For effective dating purposes, both the Start Date and End Date must be specified. The Start Date must be before the End Date.

AIIA > BS > 1001 > 1010 > 1100 > 1110

1110
Cash
Account

• Name	1110
Description	Cash
Parent	1100
Summary	Yes
Financial Category	
Start Date	8/19/2023
End Date	8/19/2023
Sequence number	1
numeric string prop	

For the second issue, you can edit either the Start Date or the End Date so that the End Date is after the Start Date. However, in this example the severity of this issue is a Warning in the Submit stage, and an Error in the Approve stage. This means that you can submit the request without correcting the issue at this time. You must confirm that you want to continue submitting the request despite the validation issue.



Acquired GL (Legacy) - Request 7499

Name	Description
AllA	All Accounts
BS	Balance Sheet Accounts
1001	Total Assets
1010	Current Assets
1100	
1150	
1110	
1200	
1300	
1400	
1020	
1700	
TLOE	
NI	Net Income

Warning

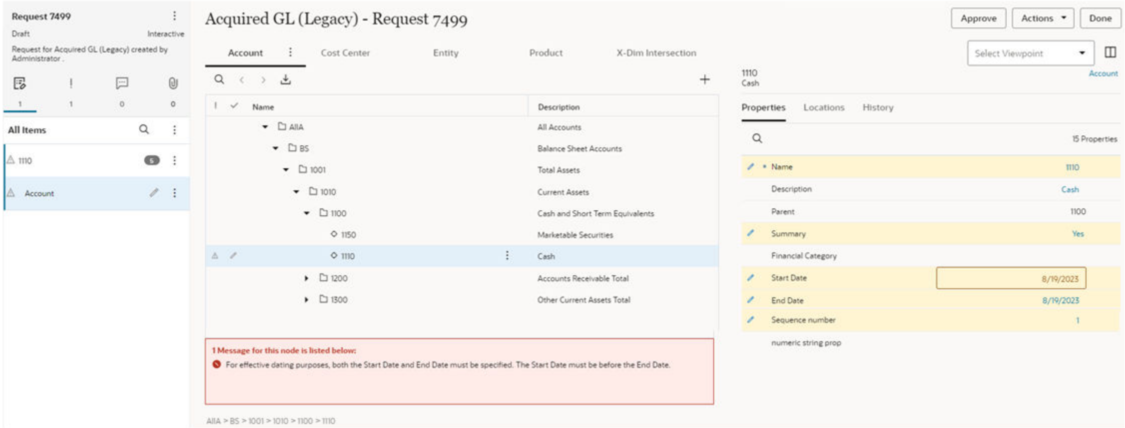
This request contains validation issues that should be reviewed before the request is submitted.

Click Continue to proceed with submission of the request despite these warnings.
Click Cancel to return to the request to review the warnings.

1 Message for this node is listed below:

For effective dating purposes, both the Start Date and End Date must be specified. The Start Date must be before the End Date.

Notice that the validation issue is now displayed as an Error, not a Warning. Because the severity of this issue is set to Error for the Approve stage, you cannot approve the request without correcting the error.



Request 7499
Draft Interactive
Request for Acquired GL (Legacy) created by Administrator .

Acquired GL (Legacy) - Request 7499

Name	Description
AllA	All Accounts
BS	Balance Sheet Accounts
1001	Total Assets
1010	Current Assets
1100	Cash and Short Term Equivalents
1150	Marketable Securities
1110	Cash
1200	Accounts Receivable Total
1300	Other Current Assets Total

1 Message for this node is listed below:

For effective dating purposes, both the Start Date and End Date must be specified. The Start Date must be before the End Date.

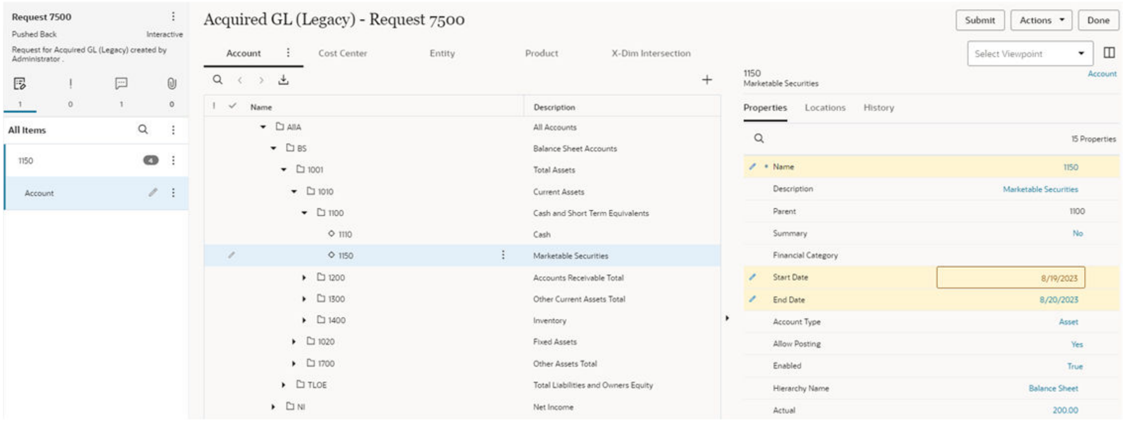
AllA > BS > 1001 > 1010 > 1100 > 1110

1110
Cash
Account

15 Properties

Name	1110
Description	Cash
Parent	1100
Summary	Yes
Financial Category	
Start Date	8/19/2023
End Date	8/19/2023
Sequence number	1
numeric string prop	

When you correct the Start or End Date, the validation issue is no longer displayed:



Resolving Issues from Other Contributors

As a request submitter or collaborator, you may not be able to view or take action on the request items that are causing validation issues from the request pane. This can happen in the following ways:

- You may be assigned or invited to a request, but you don't have access to one or more of the viewpoints where changes were made by the submitter or by a prior enricher.
- You may be assigned or invited to a request in which a property has been updated, but you do not have access to that property.
- A pushed back or recalled request that you submitted may have been enriched with changes that your access does not enable you to perform.

To resolve these issues, perform an action:

- Delete the entire request item or request item instance. In the request pane, click ⋮ next to the request item, and select **Delete**.
- Inspect the request and delete the individual request actions in the request item that are causing the issue: See Inspecting Request Items.

> **Tip:**
>
> Click the ▽ button to switch from **My Issues** to **All Issues** in order to see issues from other contributors.

Resolving Derived Property Validation Issues

Validation issues for derived properties can be resolved in several ways including:

- Override the derived property with a defined value if the property is editable
- Modify a different property which the derived property depends on
- Move or remove the node in the viewpoint
- Delete the request item or item instance

For more information, see Derived Properties.

Resolving Node Already Exists Validation Issues

There are a few scenarios that can result in a "Node already exists" validation error being displayed.

- **A new node is being added with a name that conflicts with the name of an existing node in the same node type:** In this case, change the name of the node being added to make it unique.
- **A request generates multiple subscription requests to add the same node in multiple viewpoints:** When this happens, the first subscription request to be completed adds the node, and then when the other requests are completed a "Node already exists" validation error is displayed. To correct this error, use the **Repair** option. See Repairing Request Items.

Repairing Request Items

Occasionally, a request will generate multiple subscription requests to add the same node in multiple viewpoints. When this happens, the node is added when the first subscription request is completed but when the second subscription request is processed the request fails with a validation message that the node already exists in that node type.

Use the Repair option to correct this issue. When you repair a request, the system converts the Add actions in the request to Insert actions (for hierarchy viewpoints) or Property Update actions (for list viewpoints) and then reprocesses the request.

Considerations

- The Repair option is enabled only for users with the ability to edit a request (for example, request assignees or request enrichers).
- When you repair a request, the only changes that are made are the conversions of Add actions to Insert or Property Update actions for nodes with the "Node already exists" errors. Other validation issues in the request are not affected and must still be addressed.

> **Caution:**
>
> Use the Repair option only to resolve issues where the same node is being added in multiple requests, and the subsequent requests after the first one is completed are displaying validation errors that the node already exists. Do not use the Repair option when you are trying to add a new node that has a name that conflicts with an existing node name. Instead, choose a unique name for the node being added.

To repair a request:

1. Open a request with the request items that you want to correct.
2. At the top of the request item or request pane, click **Actions** ⋮, and then click **Repair**.
3. On the confirmation dialog box, click **OK**.
   The system processes the repair operation and displays the updated request items.
4. Resolve any remaining validation issues, and then click **Submit**.

Reviewing Request File Attachments

The request file attachment contains the inbound request file before request items are created.

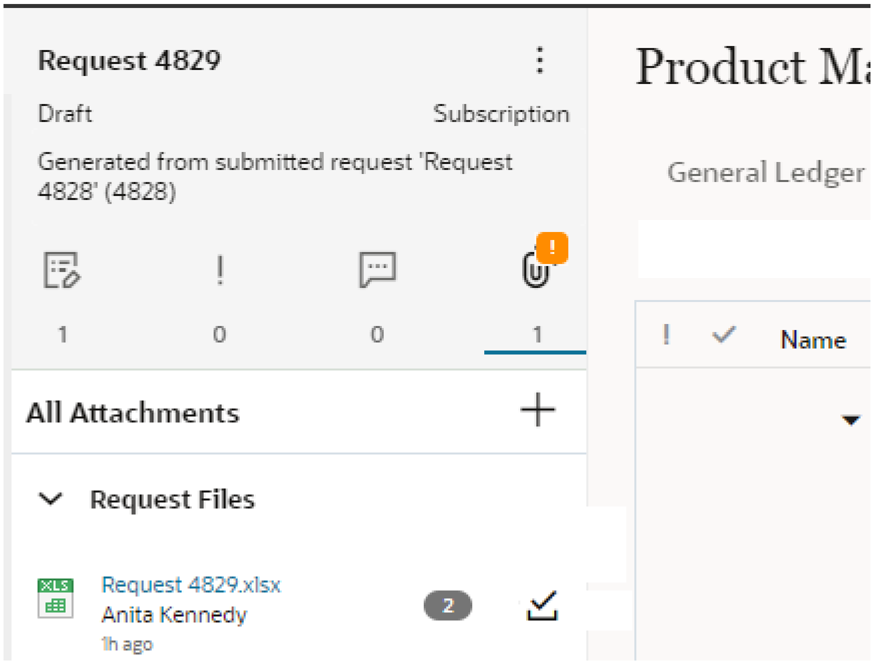
Request items can be loaded in one of three ways:

- A request file that a user has uploaded. See Making Changes Using a Load File.
- A request file that was generated from a subscription. See Subscribing to Viewpoints.
- A request file that was generated as part of a viewpoint comparison. See Creating Request Items from Comparison Results.

The file that contains the actions that will be loaded into the request gets attached to the request so that you can download and review it.

If a file attachment for a request has any skipped rows:

- A warning badge is displayed on the **Attachments** tab.
- The number of skipped rows is displayed next to the link to download the request file.



> **Note:**
>
> If you try to submit a request that has a file attachment with skipped rows, a warning is displayed that the file attachment has skipped rows and you must confirm that you want to continue submitting the request.
> When you download the file and review the skipped rows, the warning is no longer displayed when you submit the request.

To view the inbound request file:

From a draft request, click **Attachments** 📎 and then click the link to download the request file. This file contains the synchronized actions that will be loaded into the request. The `Status` column indicates whether or not a row in the request file was created for the action.

After you download the request file, a **Downloaded** ✓ icon is displayed next to the name of the request file. Hover over the icon to view the name of the user who downloaded the attachment.

> **Tip:**
>
> It can be helpful to compare the inbound file with the file that contains the request items that were created from that file during request processing. To view that file, download the request item details. See Downloading a Single Request to a File.

Downloading a Single Request to a File

You can download a draft or completed request to a file, and then review or modify the file in Microsoft Excel.

For example, for draft requests you can download a request, modify it to add additional items or update existing items, and then load it back into the same view. For completed requests, you can download the request and then load the file into a request in a different view to apply the same changes to that view.

The file format for the request download uses the same format as the request load file. See Request Load File Format.

Considerations

- A separate worksheet is created for each viewpoint that contains an action in the request.

  > **Note:**
  >
  > When you download a request to a file, the viewpoint name must be 30 or fewer characters or a label must be set up. See Inspecting a Viewpoint for information on using a label.

- A download file can contain multiple rows for the same request item or node.
- An item or node can appear in multiple worksheets if there were actions performed in multiple viewpoints for that item.
- If you have validated the request, any validation errors in the request are listed in the download file.

To download a request:

1. Perform an action:

   - From **Requests**, click ⋮ next to the request to download, and then select **Download to File**.

- Inspect a request, click **Details**, click ⋮, and then select **Download to File**.
- From an open view, open the **Requests** panel, click ⋮ next to the request name, and then select **Download to File**.

2. Click **Open** to display the request in Excel, or click **Save File** and select a location to save the request to your local machine.

> **Note:**
>
> By default, the file is saved with the request name. You can rename the file when you save it.

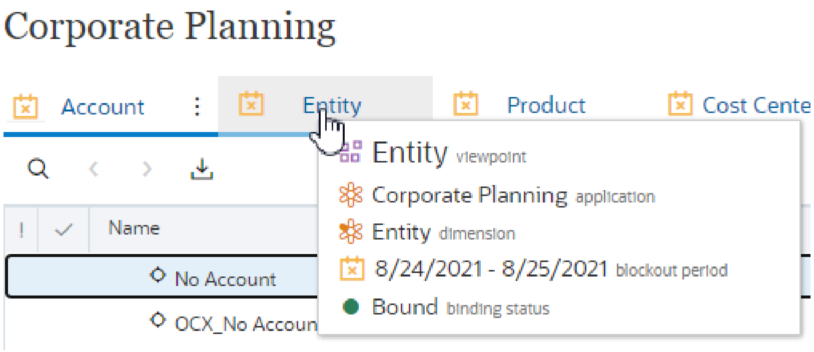
To upload the saved request file, see Making Changes Using a Load File.

Requests and Blockout Periods

An application or dimension *Owner* or *Metadata Manager* may set up a blockout period on an application or a dimension during which requests are held and not applied to the active view until the blockout period is over.

During the blockout period, you can continue to submit, approve, enrich, and commit requests. After the request is submitted (and all of the approve and commit policies are fulfilled, if applicable), the request enters a Blocked state and is held until the blockout period is over.

Viewpoints for dimensions that are currently in a blockout period for you are indicated by a 📅 icon. Any requests that you submit for a viewpoint with the blockout icon will be blocked until the blockout period ends. If you hover over the name of the viewpoint in a view, the blockout period start and end dates are displayed. Current blockout periods are indicated by a yellow blockout icon, and upcoming blockout periods are indicated by a green blockout icon.



> **Note:**
>
> Depending on how the blockout is configured, application and dimension owners and data managers may not be affected by the blockout period. The blockout icon is not displayed if the blockout does not affect you. For more information, see Understanding Blockout Periods.

For requests that contain request items from different applications or dimensions, if any application or dimension in the request is currently in a blockout period the entire request is blocked. All changes from all request items are applied after the blockout period has ended for all associated applications or dimensions.

11

Matching and Deduplicating

The process of matching and deduplicating enables you to analyze incoming request items or nodes in a viewpoint and identify the nodes that might be duplicates in order to merge them.

The following matching operations are available:

- Matching and Merging Request Items: Match incoming request items for adding new nodes to nodes that already exist nodes in a node type in order to avoid duplicate nodes, and then merging the changes from the incoming request items into the nodes in the viewpoint.
- Deduplicating Nodes in a Viewpoint: Evaluate similar nodes in a viewpoint and then merge them into a single node if they are duplicates of one another.

Matching and Merging Request Items

Matching and merging refers to the process of matching incoming request items for adding new nodes to nodes that already exist nodes in a node type in order to avoid duplicate nodes, and then merging the changes from the incoming request items into the nodes in the viewpoint.

Data managers perform matching on request items from different data sources by using the matching workbench. The matching and merging process follows this general sequence:

1. The request items are brought in (manually, through a subscription, or through a load file).
2. The data manager creates and runs a match for a particular node type and data source. See Creating and Running New Matches and Understanding Data Sources.
3. The matching workbench displays the existing nodes and all of the potential matches as determined by the matching rules that were configured for each data source. See Understanding Match Results and Creating, Editing, and Deleting Matching Rules.

> **Note:**
>
> Match results with match scores that meet or exceed the **Auto Accept Threshold** on the matching rule are displayed as "Accepted" in the matching workbench. Match results with match scores that fall below the **Auto Exclude Threshold** on the matching rule are not displayed in the workbench.

4. The data manager reviews the matches and accepts or rejects each match, and then applies the changes. See Reviewing Match Results and Applying Changes.
5. The accepted matches are applied to the request and the request items are merged with the existing nodes. The property values of the request items are updated as defined by the survivorship rules. See Creating, Editing, and Deleting Survivorship Rules.
6. The accepted matches are retained for future matching operations. The system stores accepted matches by data source and node type so that the next time a request item comes in for the same data source and node type, the match to an existing node is performed automatically. See Understanding Automatic Matching and Merging Using Prior Match Results.

For more information about the matching and merging process, see Working with Matching and Deduplication.

Creating and Running New Matches

Matching enables you to compare incoming request items for added nodes to existing nodes in a list or hierarchy viewpoint. Therefore, the first step in a matching operation is to create a request with the request items that you want to match. You can create the request manually (see Making Changes Interactively), using a request load file (see Making Changes Using a Load File), or via a subscription (see Subscribing to Viewpoints).

Best Practice

When matching and merging request items in a hierarchy viewpoint, it is a best practice to merge the parent nodes and apply the changes before merging the child nodes. For example, if your request contains geographical information for City, State, and Country, merge and apply the changes for Country first, then State, and then City.

Considerations

You must have *Data Manager* permission or greater on the dimension that contains the node type in order to run matching for that node type.

To create and run a new match:

After you have created or loaded the request items that you want to run matching for, perform the following actions to create and run a match:

1. From the viewpoint with the open request, click the **Match and Deduplicate** tab on the left side of the viewpoint window.

2. In the Match Pane, click **New** +, and then select **Match Request Items**.

   > **Note:**
   >
   > The following conditions must be true in order to create a new match:
   >
   > - You must be in an open request in the view.
   > - The request must have at least one request item with an Add action.

3. On the **New Match** screen, perform these actions:

   a. Select the application for the match. Only the applications that contain request items with Add actions in the request are available for selection.

   b. Select the node type for the match. Only node types for request items in the request are displayed.

   c. Select the data source for the match. Only data sources for the request items with the selected node type in the request are displayed.

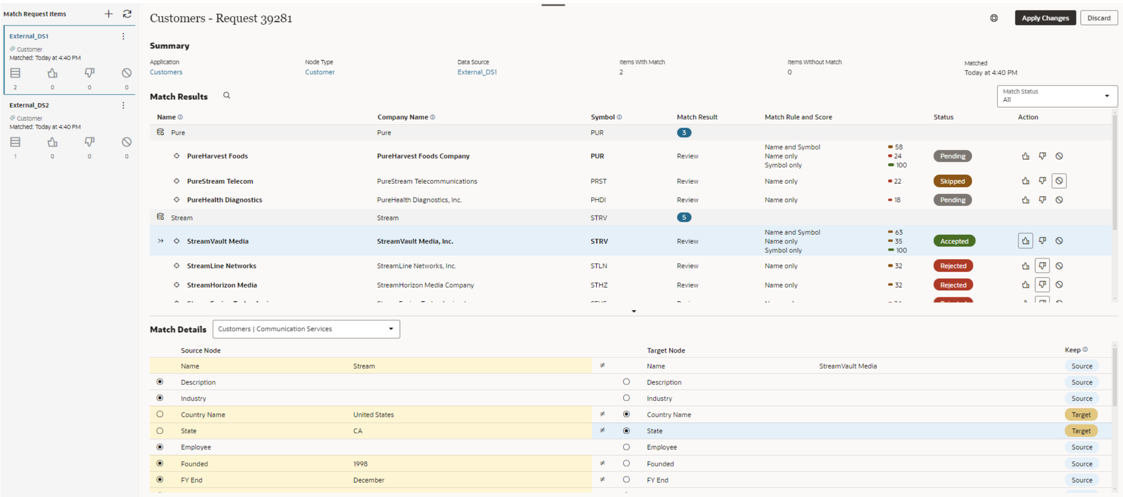
   d. Click **Run Match**.

   The matching operation is started using the defined match rules for the node type and data source, and a progress bar is displayed. When the matching operation has finished, the match result set is displayed. See Understanding Match Results.

Understanding Match Results

The Match Results screen displays the match candidates and enables you to accept, reject, or skip them.

After you run a match, the screen displays the match candidates:



Customers - Request 39281

Summary

Application	Node Type	Data Source	Items With Match	Items Without Match	Matched
Customers	Customer	External_DS1	2	0	Today at 4:40 PM

Match Results

Match Status: All

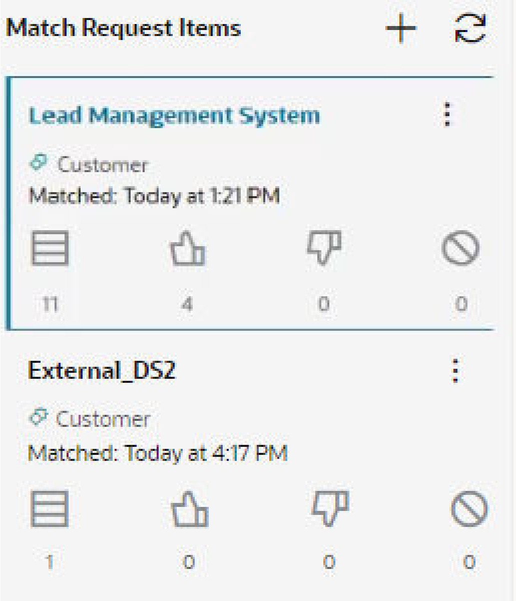
Name	Company Name	Symbol	Match Result	Match Rule and Score		Status
Pure	Pure	PUR	3			
PureHarvest Foods	PureHarvest Foods Company	PUR	Review	Name and Symbol Name only Symbol only	58 24 100	Pending
PureStream Telecom	PureStream Telecommunications	PRST	Review	Name only	22	Skipped
PureHealth Diagnostics	PureHealth Diagnostics, Inc.	PHDI	Review	Name only	18	Pending
Stream	Stream	STRV	5			
StreamVault Media	StreamVault Media, Inc.	STRV	Review	Name and Symbol Name only Symbol only	63 35 100	Accepted
StreamLine Networks	StreamLine Networks, Inc.	STLN	Review	Name only	32	Rejected
StreamHorizon Media	StreamHorizon Media Company	STHZ	Review	Name only	32	Rejected

Match Details: Customers | Communication Services

	Source Node				Target Node		Keep
	Name	Stream	≠		Name	StreamVault Media	Source
●	Description			○	Description		Source
●	Industry			○	Industry		Source
○	Country Name	United States	≠	●	Country Name		Target
○	State	CA	≠	●	State		Target
●	Employee			○	Employee		Source
●	Founded	1998	≠	○	Founded		Source
●	FY End	December	≠	○	FY End		Source

The screen contains the following sections:

Match Request Items Panel



Lead Management System
Customer
Matched: Today at 1:21 PM
11 4 0 0

External_DS2
Customer
Matched: Today at 4:17 PM
1 0 0 0

The Match Request Items panel displays the match result sets for the matches that you have run for this request by data source. If your request contains request items with different data sources, you can run matches for each data source and the results are displayed in the match request items panel. Click a match result set to switch your view to the matches from that data source.

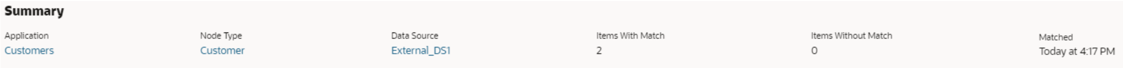
> **Note:**
>
> Match results sets persist if you navigate away from the matching workbench (for example, by returning to the request view or closing the viewpoint).

Each match result set displays the total number of request items that have match candidates, as well as the number of request items with match candidates that were accepted, rejected, and skipped.

From this panel, you can perform the following actions:

- Create and run a new match. See Creating and Running New Matches.
- Rerun a current match. See Rerunning Matches.
- Apply the changes from a match. See Applying Changes from Match Results.
- Discard the match results from a match. See Discarding Match Results.

Summary Section



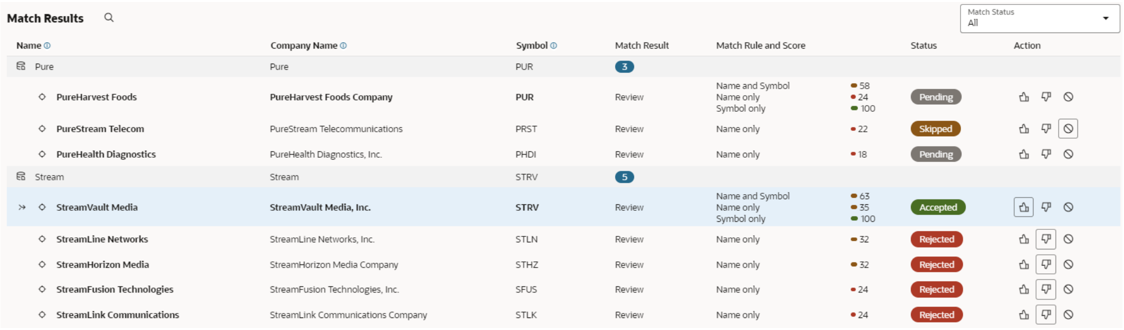
Summary

Application	Node Type	Data Source	Items With Match	Items Without Match	Matched
Customers	Customer	External_DS1	2	0	Today at 4:17 PM

The Summary section displays the application, node type, and data source that were used for the match, the number of items in the request with and without a match, and the time the match was run. You can click the application, node type, or data source to inspect that object.

> **Note:**
>
> Each match result set is specific to an application, node type, and data source. So, if your request contains request items from multiple data sources, you must run separate matches for each data source in order to see the match candidates for each source.

Match Results Section



Match Results

Match Status: All

Name	Company Name	Symbol	Match Result	Match Rule and Score		Status	Action
Pure	Pure	PUR	3				
PureHarvest Foods	**PureHarvest Foods Company**	**PUR**	Review	Name and Symbol Name only Symbol only	58 24 100	Pending	
PureStream Telecom	PureStream Telecommunications	PRST	Review	Name only	22	Skipped	
PureHealth Diagnostics	PureHealth Diagnostics, Inc.	PHDI	Review	Name only	18	Pending	
Stream	Stream	STRV	5				
StreamVault Media	**StreamVault Media, Inc.**	**STRV**	Review	Name and Symbol Name only Symbol only	63 35 100	Accepted	
StreamLine Networks	StreamLine Networks, Inc.	STLN	Review	Name only	32	Rejected	
StreamHorizon Media	StreamHorizon Media Company	STHZ	Review	Name only	32	Rejected	
StreamFusion Technologies	StreamFusion Technologies, Inc.	SFUS	Review	Name only	24	Rejected	
StreamLink Communications	StreamLink Communications Company	STLK	Review	Name only	24	Rejected	

The Match Results section displays the match candidates in the selected match result set, and enables you to accept, reject, or skip matches. The properties that were used in the match rule are highlighted in bold.

If a candidate meets the criteria for multiple matching rules, a single result is returned for that candidate with the associated matching rules displayed. The matching score for each rule is also displayed. For example, in the screenshot above the StreamVault Media candidate met the criteria of three matching rules: Name and Symbol, Name only, and Symbol only.

> **Note:**
>
> If a matching rule meets or exceeds the **Auto Accept Threshold** (see Creating, Editing, and Deleting Matching Rules), the candidate is automatically marked as Accepted and the other candidates are marked as Rejected. You can review the match candidate to modify this status or to select which properties to keep (see Selecting the Properties to Keep During a Merge) before applying your changes.
>
> If multiple matching rules meet or exceed the **Auto Accept Threshold**, the rule with the highest matching score is accepted. If the rules have the same matching score, then the rule with the highest priority (that is, the lowest **Rule Order**) is accepted. See Creating, Editing, and Deleting Matching Rules.
>
> If a matching rule meets or falls below the **Auto Exclude Threshold**, the match candidate is not displayed in the match results.

The Match Results section contains the following information:

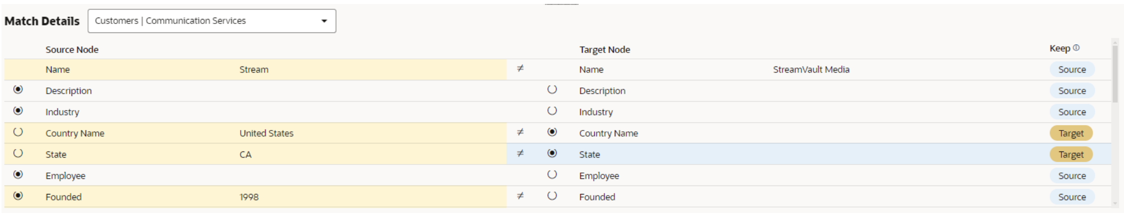
- **Request Items**: The result set is grouped by the incoming request items. Request items are displayed with a data source icon, and highlighted with a grey background. The Match Result column for each request item displays the number of match results that were found.
  In the above example, the first row with the name "Pure" and the fifth row with the name "Stream" are the incoming request items.
- **Match Candidates**: Indented under each request item are the match candidates. The match candidates are existing nodes there were determined to be match candidates based on the matching rules, and they are displayed with a node icon. The following details about each candidate are displayed. (The **Example** column shows the corresponding values from the example screen shot, above.)

Column	Description	Example
Name	The Name property of the match candidate. **Note:** If the Name property was used for matching, an information icon is displayed in the column header.	For the Pure request item, the match candidates include PureHarvest Foods, PureStream Telecomm, and PureHealth Diagnostics. For the Stream request item, the match candidates include StreamVault Media, StreamLine Networks, and other names that match the rule criteria.

Column	Description	Example
Additional properties used for matching	If other properties besides Name are used in the match rules, they are displayed as columns with an information icon in the column header.	The other properties that were used in the match rules in the above example are: – Company Name – Symbol
Match Result	For incoming request items, displays the number of matches. For match candidates, displays **Review**, indicating that a node in the node type was identified as a possible match for the request item.	The request item rows display the number of matches (3 and 5), and the match candidate rows all display **Review** in the Match Result column, indicating that they have been identified as possible match candidates.
Match Rule and Score	Displays the following information: – The name of the match rule for the node type and data source – The confidence level of the match candidate, as indicated by a numerical score and a colored meter bar. For more information about the match scoring, see the FAQ How are match scores calculated and how do I use them?	For both incoming request items, the **Symbol Only** match rules are displayed with 100 confidence levels, indicating that the match candidates match the matching rules criteria completely. The candidates display lower numbers for the **Name and Symbol** and **Name Only** rules, indicating a lower level of confidence for those candidates because they match only some of the matching rules criteria.

Column	Description	Example
Status	Displays the following information: – **Pending**: The match candidate has not yet had an action performed on it. – **Accepted**: The match candidate was accepted as a match. **Note:** Accepting a match candidate automatically updates all other Pending match candidates for that request item to Rejected. – **Rejected**: The match candidate was rejected, either because you accepted another match candidate or because you rejected it. – **Skipped**: The match candidate was reviewed and marked as skipped. When you mark a candidate as Skipped, the next time you run matching for this node type and data source it will be presented again for your review so that you can take action on it.	Two of the Pure candidates have a status of **Pending**, indicating that they haven't been accepted or rejected yet, while the PureStream Telecomm match candidate was **Skipped**. For the Stream candidates, the first candidate has a status of **Accepted**. This automatically updated the status of all other Pending match candidates to **Rejected**.
Action	Enables you to accept, reject, or skip a match candidate. See Reviewing Match Results and Applying Changes.	The Accept, Reject, and Skip buttons are displayed for each match candidate.

Match Details



Match Details Customers | Communication Services

	Source Node				Target Node		Keep
	Name	Stream	≠		Name	StreamVault Media	Source
●	Description			○	Description		Source
●	Industry			○	Industry		Source
○	Country Name	United States	≠	●	Country Name		Target
○	State	CA	≠	●	State		Target
●	Employee			○	Employee		Source
●	Founded	1998	≠	○	Founded		Source

The Match Details section displays the properties for both the request item and the match candidate in a side-by-side layout so that you can compare them.

Use the drop down menu to select the viewpoint (and parent, for hierarchy viewpoints) for the source node.

- The **Source Node** section displays the properties and values for the incoming request item instance, including the parent and relationship properties.
- The **Target Node** section displays the properties and values of the existing match candidate in the viewpoint.

Properties that are being updated in the request item are highlighted in yellow, and differences in property values between the incoming request item and the existing target node are indicated with a Not Equal (≠) sign.

You can choose to keep the source or target property value for a match candidate by selecting the radio button in the Source Node or Target Node column. The **Keep** column indicates whether the value for a property comes from the source or target node. See Selecting the Properties to Keep During a Merge.

> **Note:**
>
> The properties that are displayed and the default keep settings are based on the survivorship rule that was set up for the node type and data source. See Creating, Editing, and Deleting Survivorship Rules.

Reviewing Match Results and Applying Changes

After you run a match, you review the match results and accept, reject, or skip the matches for the request items. Then, you apply the changes to merge the accepted matches from the request items into the existing nodes in a viewpoint.

Searching and Filtering Match Results

Searching Match Results

Click the **Search** icon (🔍) and enter text to search for a match result.

> **Note:**
>
> The system searches the current page of match results only. For very large result sets, you may have to page down to load more results in order to search them.

Filtering Match Results

Use the Match Status drop down menu to filter the match results by match status:

- **All**
- **Accepted**
- **Rejected**
- **Skipped**

- **Pending**
- **No Matches**

> **Tip:**
>
> Click **Refresh Results** 🔄 to refresh the filtered match status list.

Accepting, Rejecting, or Skipping Match Results

Select a row in the match results table, and perform an action:

Action	Results
Accept (👍)	The request item is merged into the matching candidate node when you apply your changes. It is not added as a separate node in the node type. The match results table is updated as follows when you accept a match: • The Status column for the match item that you selected is updated to **Accepted**. • The Status column for all other pending match candidates for this request item is updated to **Rejected**. The status for skipped items does not get updated. • A **Merge** icon (») is displayed next to the node icon for the match candidate. This indicates that the request item will be merged with this node when you apply the changes.
Reject (👎)	The request item is kept as a separate node that is added to the node type when you apply your changes. The Status column for the match item that you selected is updated to **Rejected**.

Action	Results
Skip (⦸)	The request item is skipped for the current request so that you can re-evaluate it at a later time. The Status column for the match item that you selected is updated to **Skipped**. **Tip:** Use **Skip** to override a previous Accept or Reject action before applying changes. For example, if you initially reject a match candidate and then decide that you want keep that candidate for later evaluation, clicking Skip will undo the reject action. The next time matching is run for the current request, that candidate is presented as a potential match.

Selecting the Properties to Keep During a Merge

The Match Details section of the matching workbench displays the properties for both the request item and the match candidate (see Match Details). By default, the properties and relationships that are merged into a match candidate are defined by the survivorship rule . See Creating, Editing, and Deleting Survivorship Rules.

You can select whether to keep the source or target values for these properties when the nodes are merged by selecting the radio buttons in either the Source Node or the Target Node column for each property. The **Keep** column indicates whether the value for a property comes from the source or target node.

Caution:

After a request with a matched and merged item has been completed and closed, you cannot undo the selection of which properties to keep. You must delete and re-add the existing target node to delete the stored selection setting.

Applying Changes from Match Results

After you have accepted at least one match result, click the **Apply Changes** button to merge the request items for all Accepted match results. Applying the changes merges the request items for the matches that you accepted with the existing nodes in the viewpoint, as follows:

- The node ID and name in the request items are changed to the existing target nodes.
- Properties from the request items are merged into the existing target nodes.
- Nodes in hierarchies are either inserted (when shared nodes are enabled) or moved (when shared node are not enabled) to their respective parents that are specified in the request items.

After you click **Apply Changes**, the match candidates that have been accepted or rejected are removed from the match results table.

> **Tip:**
>
> To view the candidates in a request that were merged, inspect the request, navigate to the **Details** tab, and change the request filter to **Merged**. See Inspecting Requests.

Best Practice

When matching and merging nodes in a hierarchy, it is a best practice to complete all matching and merging operations for the parent nodes, apply the changes, and then match and merge the child nodes. Applying changes to child nodes and then applying additional changes to their parent nodes afterwards can lead to unexpected results.

Considerations

- Applying changes from the match process updates the request items in a request, but no changes are committed until the request itself is completed and closed. You can continue to modify the request items (see Inspecting Request Items) before submitting the request.
- The status of the accepted matches are retained for the node type and data source after the request is completed and closed. The next time you bring in a node that has the same name from the same node type and data source, the node is automatically matched and merged with the existing node during request processing.

> **Caution:**
>
> After a request with a matched and merged item has been completed and closed, you cannot undo that match and merge operation to the existing node. Before the request is competed and closed, you can undo the match to an existing node in the following ways:
>
> – Before applying changes (by clicking Reject or Skip in the matching workbench)
>
> – After applying changes but before the request is completed (by deleting the request item and recreating it separately)
>
> However, after the request has been completed and closed you can no longer undo that match. You must delete and re-add the existing target node to delete the stored match information.

Discarding Match Results

Click the **Discard** button on the match results screen to discard all match results that haven't been applied yet. This action removes the match result set from the Match Request Items page and deletes all accepted, rejected, and pending matches for the selected node type and data source. However, match results that have already been applied are not discarded. You can review and delete those merged request items from the request itself (see Inspecting Request Items).

Rerunning Matches

In the match result set, click **Actions** ⋮ and select **Rerun Match** to rerun the matching process for this node type and data source. This can be helpful if you have updated your matching rules or you have made changes to the request items that were already matched.

Understanding Automatic Matching and Merging Using Prior Match Results

After you have accepted and applied matches in a request and the request has been completed and closed, the status of those matches is retained for the node type and data source.

When you bring in nodes with the same name for the same node type from the same data source, the nodes that were previously accepted as matches are merged automatically with the existing target nodes as part of request processing during the following operations:

- For registered data sources:
  - Creating request items from viewpoint compare
  - Copying nodes across viewpoints
  - Consolidating requests
  - Generating subscription request items
- For unregistered data sources:
  - Request file upload
  - Loading a viewpoint in merge mode

See Reviewing Match Results and Applying Changes.

> **Note:**
>
> Automatic matching and merging takes place only after the request with the applied matches has been completed and closed. If you have an open request with accepted matches and a second request is submitted with the same node name, node type, and data source, that request items in the second request will not be automatically matched and merged because the first request hasn't been completed and closed.

Automatic Matching and Merging Process

Automatic matching and merging follows this general process:

1. The system checks the node type and data source in the request item to see if there is a prior applied match result.
2. During request processing, request items are created:
   - **If a match is found**: A request item is created to update the existing matched node instead of adding a new node. The name of the node in the request item comes from the existing matched node, and the merged data source and node columns are populated.
   - **If a match is not found**: A request item is created to add the new node. The name of the node in the request item comes from the load file.

> **Caution:**
>
> For subscriptions, request actions that take place in a registered data source are automatically synchronized to target viewpoints. That means that if you delete a source node that was previously merged into a target node, the subscription engine will automatically match and merge the request action and the Delete action will be applied to the target node.
>
> To avoid inadvertently deleting nodes in your target viewpoints, consider excluding Deletes (and Removes, if necessary) in the Included Actions filter in your subscription. See Filtering by Actions.

Frequently Asked Questions about Matching and Merging

- How are match scores calculated and how do I use them?
- Should I use a code or a data source name in a load file?
- Can I create a survivorship rule for a registered data source?
- When creating match rules, is it better to add multiple criteria to a rule or to create separate rules?
- I accidentally accepted a match that I did not mean to. Can I review my previously accepted matches and undo them?
- When are node links established between nodes?

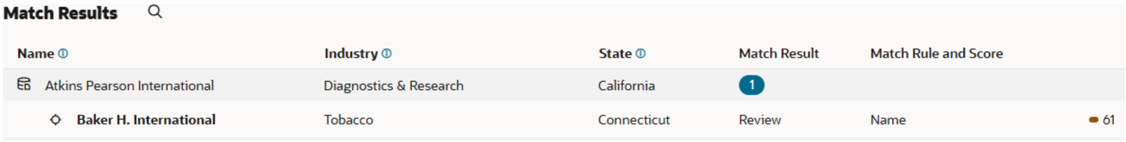
How are match scores calculated and how do I use them?

The exact mechanism for calculating a match score depends on several factors such as the data type (string, integer, date, etc.), the match type (contains, similar to, between), and whether or not the match operations are combined. But in general a higher match score indicates that more matching criteria have been met for a candidate.

Let's look at some examples to better understand how a match score is calculated.

Example 1

In this example, the match rule is matching on the Name property, which is a string:



Match Results

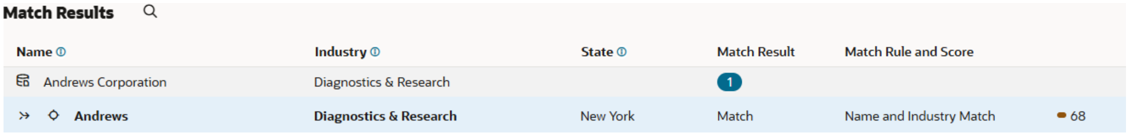
Name	Industry	State	Match Result	Match Rule and Score	
Atkins Pearson International	Diagnostics & Research	California	1		
Baker H. International	Tobacco	Connecticut	Review	Name	61

The target name that we are matching to is "Atkins Pearson International", and the source name that we are trying to match is "Baker H. International".

In this example, there are 28 characters in the target name, and the source name matches 17 of them ("a", "k", two spaces, and all of "International"). Therefore, approximately 61% (17 of 28) of the characters match, giving a match score of 61.

Example 2

In the second example, we are matching on two string properties, Name and Industry:



Match Results

Name	Industry	State	Match Result	Match Rule and Score	
Andrews Corporation	Diagnostics & Research		1		
Andrews	Diagnostics & Research	New York	Match	Name and Industry Match	68

The target name we are matching is "Andrews Corporation", and the target industry is "Diagnostics & Research". The source name is "Andrews", and the source industry is "Diagnostics & Research".

In this case, the source Name matches 37% (7 of 19) of the characters in the target Name, and the source Industry matches 100% of the target Industry. Since this is a combined match, the average is taken (37+100)/2, giving a match score of 68.

Other data types and match operators perform similar calculations to determine the match score.

Should I use a code or a data source name in a load file?

When a request file that includes data source information is processed, the data source for each node is identified in two ways:

- If a code is configured for the data source, the code is used. See Creating, Editing, and Deleting Data Sources.
- If a code is not configured for the data source, the data source name is used.

Because data source names can change over time, it is a best practice to always configure a code for your data sources and to use that code in your request files instead of the data source name.

Can I create a survivorship rule for a registered data source?

Survivorship rules determine which properties and relationships from an unregistered data source get merged from an accepted match candidate into a matching node in a node type. For registered data sources, you use a node type converter to determine how the properties and relationships from a match candidate are merged into a matching target nodes. See Working with Node Type Converters. You do not need to create survivorship rules for registered data sources.

> **Tip:**
>
> While node type converters for registered data sources determine which properties are available to be merged from an accepted match candidate into a matching node in a node type, you can still decide which of those properties gets merged. Use the Source Node and Target Node radio buttons in the Match Results panel to determine which values to keep. See Selecting the Properties to Keep During a Merge.

When creating match rules, is it better to add multiple criteria to a rule or to create separate rules?

The decision on whether to use separate match rules for specific identifying properties or as multiple criteria within a single rule is one best determined through experimentation by the implementing organization. When tuning rules in a test environment, stewards may evaluate which rule is better at generating fewer false positives.

In principle, combining identifying properties in a single match rule performs an "AND" operation while using separate rules for a specific identifying property would evaluate its value as a match determinant on a singular basis, thus serving as a potential "OR" operation across multiple rules as they are evaluated for a single combination of node type and data source.

One scenario where it might make sense to create multiple separate match rules instead of adding multiple criteria to a combined single rule is if you are auto-accepting match rules above a certain match score threshold and you expect that some criteria will meet that threshold while others may not.

For example, consider a scenario where you automatically accept matches above 90%, and you have two criteria for matching, with one matching at 100% and the other at 50%:

- If you have two separate match rules, the match rule with 100% match will automatically be accepted.
- If you have one match rule that contains both criteria, the average match score is 75%, which is below your threshold of 90% for automatically accepting the match. The match will not be auto-accepted.

So, in this example the decision to combine the criteria or to create separate match rules would depend on whether or not you wanted to automatically accept some matches above a certain threshold.

I accidentally accepted a match that I did not mean to. Can I review my previously accepted matches and undo them?

After a request with a matched and merged item has been completed and closed, you cannot undo that match and merge operation to the existing node. Before the request is completed and closed, you can undo the match to an existing node in the following ways:

- Before applying changes (by clicking Reject or Skip in the matching workbench)
- After applying changes but before the request is completed (by deleting the request item and recreating it separately)

However, after the request has been completed and closed you can no longer undo that match. You must delete and re-add the existing target node to delete the stored match information.

When are node links established between nodes?

Node links are established between a source and a target node when an existing target node is updated by an incoming source node that has a defined data source. For details, see Understanding Node Links and Data Sources.

Deduplicating Nodes in a Viewpoint

Deduplicating nodes enables you to evaluate similar nodes in a viewpoint and then merge them into a single node if they are duplicates of one another.

Deduplication Process Overview

Deduplication operations on a viewpoint follows this general process:

1. A data manager creates a request for a view that contains the viewpoint to be deduplicated.

   > **Tip:**
   >
   > A request is necessary for deduplicating a viewpoint because the outcome of the deduplication process results in processing changes to nodes in a viewpoint.

2. The data manager creates and runs a match for a specific node type in a particular viewpoint in order to deduplicate that viewpoint. See Running a Deduplication Operation for a Viewpoint.
3. The matching workbench displays the potential matches as determined by the matching rules that were configured for each data source. See Understanding Deduplication Results and Creating, Editing, and Deleting Matching Rules.

   > **Note:**
   >
   > Only the match results with match scores that exceed the **Auto Exclude Threshold** on the matching rules are displayed.

4. The data manager reviews the deduplication matches and accepts or rejects each match, and then applies the changes. See Reviewing Deduplication Results and Applying Changes.

5. The accepted matches are applied as follows:
   - The matched (source) node is deleted from the viewpoint (because it is a duplicate)
   - The properties and relationships from the duplicate node are merged into the match candidate (target) node that will remain as determined by the survivorship rules. See Creating, Editing, and Deleting Survivorship Rules.
6. The system uses the applied changes to create request items in the request. Delete actions are added for duplicate nodes, and property insert, update, and move actions are added based on the survivorship rules.

Understanding Matched Nodes and Match Candidates in Deduplication

Because the viewpoints that you are deduplicating contain both the matched nodes and the match candidates, it is important to understand the difference between the two:

- **Matched Nodes** are the nodes from the data source that you are evaluating during the matching process. When merging nodes, they become the source nodes that get deleted after the merge operation.
- **Match Candidates** are the nodes that you are matching against during the matching process. When merging nodes they become the target nodes that survive after a merge, and the properties and relationship values from the source nodes get merged into them as determined by the survivorship rules.

> **Note:**
>
> When you run deduplication using a cluster key, the cluster key is applied to the matched nodes only. It is not used to limit the nodes that are being matched against.
>
> For example, if you deduplicate a customer viewpoint using a cluster key of State and a clustering property value of Texas, only customers in Texas (matched node) will be evaluated , but they could be matched with a customer in California (match candidate) with the same name. When you merge the records, the node from Texas is deleted and its information is merged into the node from California.

Running a Deduplication Operation for a Viewpoint

Deduplicating nodes enables you to match similar existing nodes in a viewpoint and combine them into a single node.

Considerations

- You must have *Data Manager* permission or greater on the viewpoint in order to deduplicate that viewpoint.
- You cannot deduplicate time-labeled or archived viewpoints.
- You can deduplicate a viewpoint in one active request at a time. If a viewpoint is in the process of being deduplicated in another active request, you cannot select that viewpoint in a new deduplication operation.
- Each request supports one deduplication mode only. You cannot run a cluster key and a time based deduplication in the same request.
- When deduplication in a viewpoint is run:

  - Nodes in the viewpoint are matched against all nodes in the node type, even if some of those nodes are not in the existing viewpoint.
  - If a viewpoint contains shared nodes, match rules are run only for one instance of the node.
  - A limit of 20 maximum match results are displayed for each matched node.
- Requests have a limit of 10,000 request items. Because each merge operation results in two request items (a delete of the source node and a property update of the target node), when the number of matched nodes reaches 5000 the deduplication process for that request is stopped and you are prompted to create a new request to continue deduplicating nodes. The request maximum may be reached earlier if your request already contains other request items.
- Because you are deduplicating a set of nodes in a viewpoint instead of incoming request items, two different nodes can often be match candidates for each other. For example, when deduplicating a viewpoint that contains the nodes "Oracle" and "Oracle Inc", each node can be a match candidate for the other. The node that you accept as the duplicate will control which node gets deleted and which will be the surviving node. Remember, the matched nodes are the nodes that will be deleted, and the match candidate nodes are the surviving node. See Understanding Matched Nodes and Match Candidates in Deduplication.

  > **Tip:**
  >
  > When you accept a match as a duplicate, that duplicate node is marked as **Duplicate** in the Deduplication Results screen (see Understanding Deduplication Results). The marked node is the one that will be deleted.

- If three or more nodes are matched during deduplication, you cannot merge the first into the second and then merge the second into the third. You can, however, merge both the first and second into the third.
  For example, suppose you have nodes "Oracle", "Oracle Inc", and "Oracle Incorporated", and you want to keep "Oracle Incorporated" and merge information from the other two nodes into it. You can't merge "Oracle" into "Oracle Inc" and then merge "Oracle Inc" into "Oracle Incorporated". Instead, locate the matched node "Oracle" and mark it as a duplicate of "Oracle Incorporated", and then locate "Oracle Inc" and mark it as a duplicate of "Oracle Incorporated".

To deduplicate a viewpoint:

1. Create a request for the viewpoint that you want to deduplicate.
2. Click the **Match and Deduplicate** ○△△○ tab on the left side of the viewpoint window.
3. In the Match Pane, click **New** +, and then select **Deduplicate Viewpoint**.
4. On the Deduplicate Viewpoint dialog box, perform these actions:
   a. Select the **Viewpoint** that you want to deduplicate.
   b. Select a **Node Type** in that viewpoint. The node type must be configured for deduplication (see Understanding Deduplication).
   c. Select the deduplication **Mode**:
      - **Cluster Key**: Deduplicate the viewpoint using a clustering property. Select the clustering property value from the drop down menu. The clustering property values

in the drop down menu are based on the allowed values for the property that you defined as the cluster key. See Deduplicating Using a Cluster Key.

> **Note:**
>
> If a deduplication operation has already been run for the clustering property, the node creation date of the last node processed is displayed.

- **Time Based**: Deduplicate the viewpoint based on the date that the nodes were created. Enter the node creation date. See Time-Based Deduplication.

5. **Optional**: Enter a **Batch Size** to specify the number of nodes to be checked for duplicates.

> **Tip:**
>
> This can be helpful, for example, if you've made changes to a matching rule that you want to test. You can run a smaller batch and evaluate the results before deduplicating the entire viewpoint.

6. Click **Run Deduplicate**.

Deduplication is run on the viewpoint using the defined match rules for the node type and the registered data source for the viewpoint.

Deduplication Operations

Because viewpoints can contain thousands of nodes, you generally deduplicate them in batches. The batches can be defined in the following ways:

- The cluster key (see Deduplicating Using a Cluster Key)
- The node creation start date (see Time-Based Deduplication)
- The batch size that you specify when creating the match
- The request item limit in a request

Batches can also be defined by a combination of some of the above, such as a cluster key and a specified batch size.

You have several options for how you process the nodes in these batches. The following terms can help you understand these options:

Table 11-1 Batch Processing Options

Option	Definition	How to Perform
Run	Perform the initial deduplication of the first batch of nodes for a specified cluster or node creation start date.	Click **Run Deduplicate** in the Deduplicate Viewpoint dialog box.

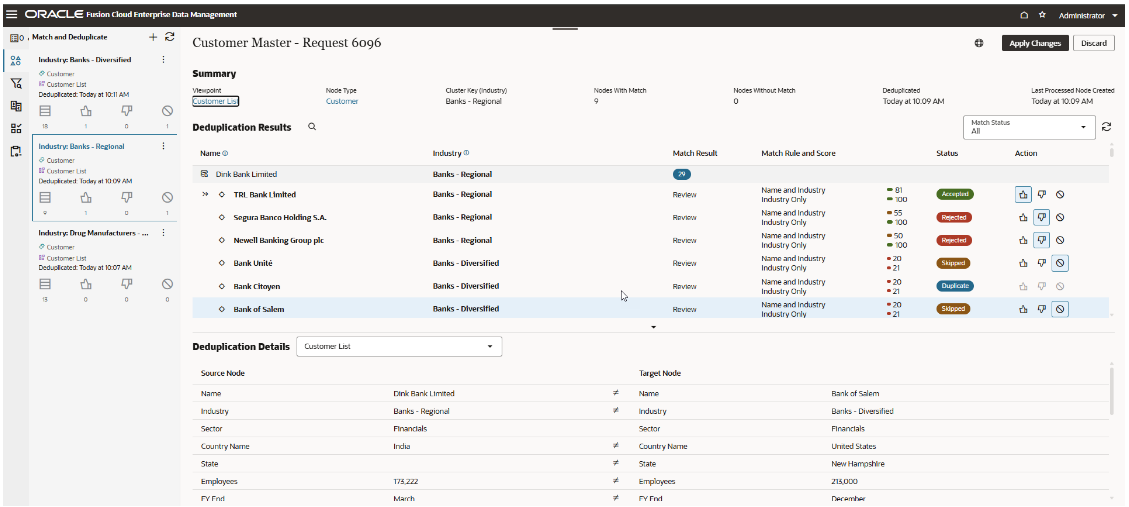
Table 11-1 (Cont.) Batch Processing Options

Option	Definition	How to Perform
Continue	Perform a subsequent duplication of the next batch of nodes for a specified cluster or node creation start date. The system tracks the nodes that have already been processed so that you can pick up where you left off.	Click **Run Deduplicate** in the Deduplicate Viewpoint dialog box after performing an initial Run operation.
Rerun	Reprocess an existing result set in a request. This may include one or more batches. **Note**: Rerun reprocesses unaccepted match results only.	In the Deduplicate Result Set panel, click **Actions** ⋮ next to the result set that you want to rerun, and then select **Rerun**.
Restart	Reprocess a cluster that was already processed by starting from the beginning of it. **Note**: Restart is available for cluster key deduplication only. **Tip**: The difference between **Rerun** and **Restart** is that Rerun reprocesses one or more *batches*, while Restart reprocesses a *cluster*.	Click **Restart** ↺ next to the Cluster Key in the Deduplicate Viewpoint dialog box.
Discard	Delete an existing result set for a given request. The last node that was processed is retained so that you can Continue the next time you run deduplication. **Note**: Deleting the request will also discard the result set.	In the Deduplicate Result Set panel, click **Actions** ⋮ next to the result set that you want to rerun, and then select **Discard**.
Discard and Rerun	Delete an existing result set for a given request and reprocess the same nodes in the result set. This may include one or more batches.	In the Deduplicate Result Set panel, click **Actions** ⋮ next to the result set that you want to rerun, and then select **Discard and Rerun**.

Understanding Deduplication Results

The Deduplication Results screen displays the match candidates and enables you to accept, reject, or skip them.

After you run a match, the screen displays the match candidates:



Industry: Banks - Diversified
Customer
Customer List
Deduplicated: Today at 10:11 AM
18 1 0 1

Industry: Banks - Regional
Customer
Customer List
Deduplicated: Today at 10:09 AM
9 1 0 1

Industry: Drug Manufacturers - ...
Customer
Customer List
Deduplicated: Today at 10:07 AM
13 0 0 0

Customer Master - Request 6096

Summary

Viewpoint	Node Type	Cluster Key (Industry)	Nodes With Match	Nodes Without Match	Deduplicated	Last Processed Node Created
Customer List	Customer	Banks - Regional	9	0	Today at 10:09 AM	Today at 10:09 AM

Deduplication Results

Match Status: All

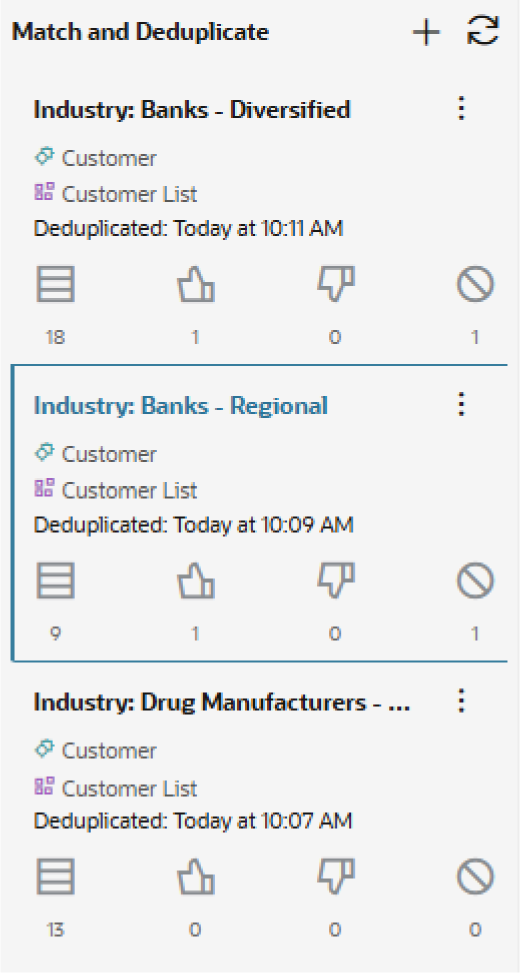
Name	Industry	Match Result	Match Rule and Score		Status	Action
Dink Bank Limited	Banks - Regional	29				
TRL Bank Limited	Banks - Regional	Review	Name and Industry Industry Only	81 100	Accepted	
Segura Banco Holding S.A.	Banks - Regional	Review	Name and Industry Industry Only	55 100	Rejected	
Newell Banking Group plc	Banks - Regional	Review	Name and Industry Industry Only	50 100	Rejected	
Bank Unité	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Skipped	
Bank Citoyen	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Duplicate	
Bank of Salem	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Skipped	

Deduplication Details Customer List

Source Node			Target Node	
Name	Dink Bank Limited	≠	Name	Bank of Salem
Industry	Banks - Regional	≠	Industry	Banks - Diversified
Sector	Financials		Sector	Financials
Country Name	India	≠	Country Name	United States
State		≠	State	New Hampshire
Employees	173,222	≠	Employees	213,000
FY End	March	≠	FY End	December

The screen contains the following sections:

Deduplicate Result Set Panel



Industry: Banks - Diversified
Customer
Customer List
Deduplicated: Today at 10:11 AM
18 1 0 1

Industry: Banks - Regional
Customer
Customer List
Deduplicated: Today at 10:09 AM
9 1 0 1

Industry: Drug Manufacturers - ...
Customer
Customer List
Deduplicated: Today at 10:07 AM
13 0 0 0

The Deduplicate Result Set panel displays the deduplication result sets for the matches that you have run for this request by clustering property or node creation date (if you are deduplicating multiple different viewpoints or node types). Click a deduplicate result set to switch your view to the matches from that clustering property or creation date.

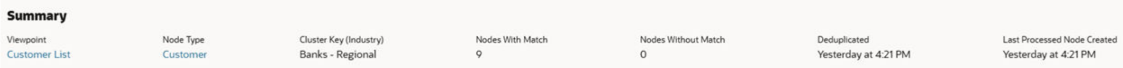
> **Note:**
>
> Deduplication result sets persist if you navigate away from the matching workbench (for example, by returning to the request view or closing the viewpoint).

Each deduplication result set displays the total number of matched nodes that have match candidates, as well as the number of matched nodes with match candidates that were accepted, rejected, and skipped.

From this panel, you can perform the following actions:

- Create and run a new match. See Running a Deduplication Operation for a Viewpoint.
- Rerun a current match. See Rerunning Matches.
- Apply the changes from a match. See Applying Changes from Deduplication Results.
- Discard the match results from a match. See Discarding Deduplication Results.

Summary Section



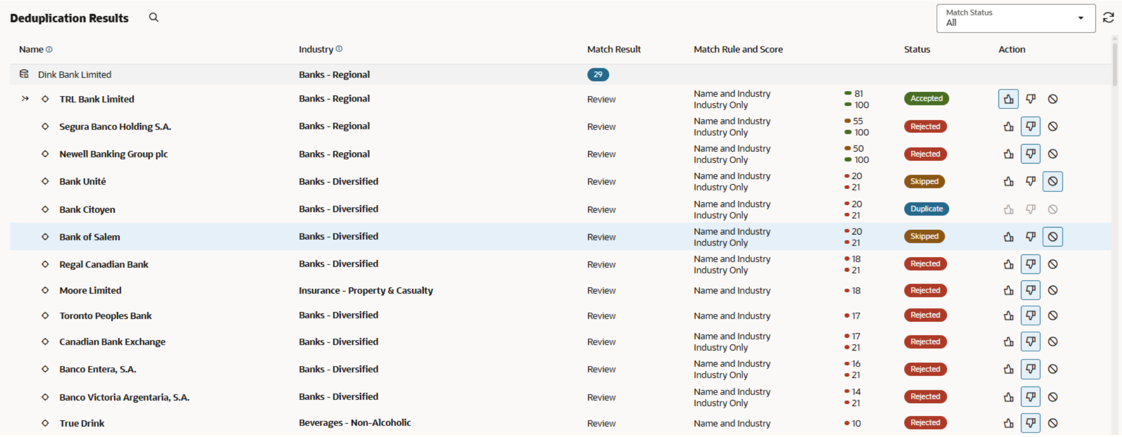
Summary

Viewpoint	Node Type	Cluster Key (Industry)	Nodes With Match	Nodes Without Match	Deduplicated	Last Processed Node Created
Customer List	Customer	Banks - Regional	9	0	Yesterday at 4:21 PM	Yesterday at 4:21 PM

The Summary section displays the viewpoint, node type, and cluster key (for cluster key-based deduplication) or the node creation date (for time-based deduplication) that was used for the match, the number of matched nodes in the viewpoint with and without a match, the date and time that the deduplication was run, and the creation date of the last node to be processed. You can click the viewpoint or node type to inspect that object.

> **Note:**
>
> Each deduplication result set is specific to an viewpoint, node type, and cluster key value or node creation date. So, if your viewpoint contains nodes with different cluster key values, you must run separate matches for each cluster key value in order to see the match candidates for each value.

Deduplication Results Section



Deduplication Results

Match Status: All

Name	Industry	Match Result	Match Rule and Score		Status	Action
Dink Bank Limited	Banks - Regional	29				
TRL Bank Limited	Banks - Regional	Review	Name and Industry Industry Only	81 100	Accepted	
Segura Banco Holding S.A.	Banks - Regional	Review	Name and Industry Industry Only	55 100	Rejected	
Newell Banking Group plc	Banks - Regional	Review	Name and Industry Industry Only	50 100	Rejected	
Bank Unité	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Skipped	
Bank Citoyen	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Duplicate	
Bank of Salem	Banks - Diversified	Review	Name and Industry Industry Only	20 21	Skipped	
Regal Canadian Bank	Banks - Diversified	Review	Name and Industry Industry Only	18 21	Rejected	
Moore Limited	Insurance - Property & Casualty	Review	Name and Industry	18	Rejected	
Toronto Peoples Bank	Banks - Diversified	Review	Name and Industry	17	Rejected	
Canadian Bank Exchange	Banks - Diversified	Review	Name and Industry Industry Only	17 21	Rejected	
Banco Entera, S.A.	Banks - Diversified	Review	Name and Industry Industry Only	16 21	Rejected	
Banco Victoria Argentaria, S.A.	Banks - Diversified	Review	Name and Industry Industry Only	14 21	Rejected	
True Drink	Beverages - Non-Alcoholic	Review	Name and Industry	10	Rejected	

The Deduplication Results section displays the match candidates in the selected deduplication result set, and enables you to accept, reject, or skip matches. For cluster key deduplication operations, the cluster key is displayed. Other properties that were used in the match rule are highlighted in bold.

If a candidate meets the criteria for multiple matching rules, a single result is returned for that candidate with the associated matching rules displayed. The matching score for each rule is also displayed. For example, in the screenshot above the TRL Bank Limited candidate met the criteria of two matching rules: Name and Industry and Industry Only.

> **Note:**
>
> If a matching rule meets or falls below the **Auto Exclude Threshold**, the match candidate is not displayed in the deduplication results.

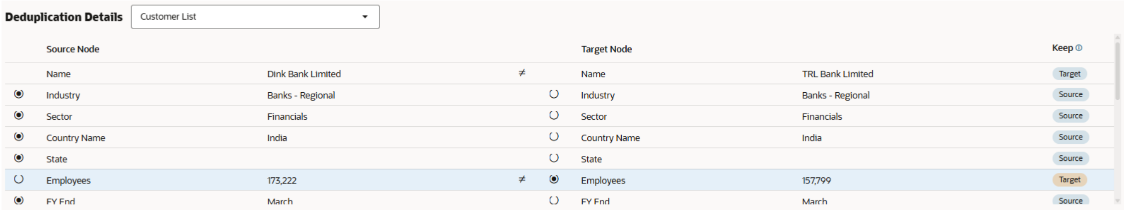
The Deduplication Results section contains the following information:

- **Matched Node**: The result set is grouped by the matched nodes (that is, the nodes from the data source that you are evaluating). Matched nodes are displayed with a data source icon, and highlighted with a grey background. The Match Result column for each matched node displays the number of match results that were found.
  In the above example, the first row with the name "Dink Bank Limited" is the matched node.
- **Match Candidates**: Indented under each matched node are the match candidates. The match candidates are the nodes that you are matching against based on the matching rules, and they are displayed with a node icon. The following details about each candidate are displayed. (The **Example** column shows the corresponding values from the example screen shot, above.)

Column	Description	Example
Name	The Name property of the match candidate. **Note:** If the Name property was used for matching, an information icon is displayed in the column header.	Match candidates include TRL Bank Limited, Bank of Salem, and Moore Limited, among others.
Clustering Property (cluster key-based deduplication only)	The clustering property value that was used for the deduplication.	Industry
Additional properties used for matching	If other properties besides Name are used in the match rules, they are displayed as columns with an information icon in the column header.	Industry
Match Result	For each matched node, displays the number of matches. For the match candidates, displays **Review**, indicating that a node in the node type was identified as a possible match for the matched node.	The matched node row displays 29, and the match candidate rows all display **Review** in the Match Result column, indicating that they have been identified as possible match candidates.
Match Rule and Score	Displays the following information: – The name of the match rule for the node type and data source – The confidence level of the match candidate, as indicated by a numerical score and a colored meter bar. For more information about the match scoring, see the FAQ How are match scores calculated and how do I use them?	The first three **Industry Only** match rules are displayed with 100 confidence levels, indicating that the match candidates match the matching rules criteria completely. The other candidates have lower numbers, indicating a lower level of confidence for those candidates because they match only some of the matching rules criteria. The candidates display various numbers for the **Name and Industry** rule, depending on how many characters are shared in the names between the matched node and the match candidates.

Column	Description	Example
Status	Displays the following information: – **Pending**: The match candidate has not yet had an action performed on it. – **Accepted**: The match candidate was accepted as a match. **Note:** Accepting a match candidate automatically updates all other Pending match candidates for that request item to Rejected. – **Rejected**: The match candidate was rejected, either because you accepted another match candidate or because you rejected it. – **Skipped**: The match candidate was reviewed and marked as skipped. When you mark a candidate as Skipped, the next time you run matching for this node type and data source it will be presented again for your review so that you can take action on it. – **Duplicate**: The match candidate is a matched node that was already accepted as a duplicate of another node in the viewpoint within this request.	TRL Bank Limited was **Accepted**, while Bank Unite and Bank Salem were **Skipped**. Bank Citoyen is marked as a **Duplicate**, meaning that it was already accepted as a duplicate node in this request.
Action	Enables you to accept, reject, or skip a match candidate. See Reviewing Deduplication Results and Applying Changes.	The Accept, Reject, and Skip buttons are displayed for each match candidate.

Deduplication Details



Deduplication Details [Customer List ▾]

	Source Node				Target Node		Keep ⓘ
	Name	Dink Bank Limited	≠		Name	TRL Bank Limited	Target
●	Industry	Banks - Regional		○	Industry	Banks - Regional	Source
●	Sector	Financials		○	Sector	Financials	Source
●	Country Name	India		○	Country Name	India	Source
●	State			○	State		Source
○	Employees	173,222	≠	●	Employees	157,799	Target
●	FY End	March		○	FY End	March	Source

The Deduplication Details section displays the properties for both the matched (source) node and the match candidate (target) node in a side-by-side layout so that you can compare them.

Use the drop down menu to select the viewpoint (and parent, for hierarchy viewpoints) for the source node.

- The **Source Node** section displays the properties and values for the matched node instance, including the parent and relationship properties.
- The **Target Node** section displays the properties and values of the match candidate in the viewpoint.

Differences in property values between the source node and the target node are indicated with a Not Equal (≠) sign.

You can choose to keep the source or target property value for a surviving node by selecting the radio button in the Source Node or Target Node column. The **Keep** column indicates whether the value for a property comes from the source or target node (see Selecting the Properties to Keep During a Merge). The properties that are displayed and the default keep settings are based on the survivorship rules that were set up for the node type and data source. See Creating, Editing, and Deleting Survivorship Rules.

Reviewing Deduplication Results and Applying Changes

After you run deduplication, you review the results and accept, reject, or skip the matches for each matched node. Then, you apply the changes to merge the nodes in the viewpoint.

Searching and Filtering Deduplication Results

Searching Deduplication Results

Click the **Search** icon (🔍) and enter text to search for a deduplication result.

> **Note:**
>
> The system searches the current page of deduplication results only. For very large result sets, you may have to page down to load more results in order to search them.

Filtering Deduplication Results

Use the Deduplication Status drop down menu to filter the deduplication results by match status:

- **All**
- **Accepted**
- **Rejected**
- **Pending**

- **Skipped**
- **Duplicate**

> **Tip:**
>
> Click **Refresh Results** 🔄 to refresh the filtered match status list.

Accepting, Rejecting, or Skipping Deduplication Results

Select a row in the deduplication results table, and perform an action:

Action	Results
Accept (👍)	The matched node is deleted and the property and relationship values are merged into the matching candidate node when you apply your changes. The deduplication results table is updated as follows when you accept a match: • The Status column for the match candidate that you selected is updated to **Accepted**. • The Status column for all other pending match candidates for this source node is updated to **Rejected**. The status for skipped items does not get updated. • A **Merge** icon (») is displayed next to the node icon for the match candidate. This indicates that the matched node will be merged with this node when you apply the changes. • If the matched node is also a match candidate for other matched nodes, those match candidates are marked as **Duplicate** in the rest of the deduplication results table.
Reject (👎)	The matched node is kept as a separate node when you apply your changes. The Status column for the match item that you selected is updated to **Rejected**.

Action	Results
Skip (⃠)	The match candidate is skipped for the current request so that you can re-evaluate it at a later time. The Status column for the match item that you selected is updated to **Skipped**. **Tip:** Use **Skip** to override a previous Accept or Reject action before applying changes. For example, if you initially reject a match candidate and then decide that you want keep that candidate for later evaluation, clicking Skip will undo the reject action. The next time matching is run, that candidate is presented as a potential match.

Selecting the Properties to Keep During a Merge

The Deduplication Details section of the matching workbench displays the properties for both the matched (source) node and the match candidate (target) nodes (see Deduplication Details). By default, the properties and relationships that are merged into a surviving node are defined by the survivorship rule. See Creating, Editing, and Deleting Survivorship Rules.

You can select whether to keep the source or target values for these properties when the nodes are merged by selecting the radio buttons in either the Source Node or the Target Node column for each property. The **Keep** column indicates whether the value for a property comes from the source or target node.

Applying Changes from Deduplication Results

After you have accepted at least one deduplication result, click the **Apply Changes** button to merge the source nodes for all Accepted match results. Applying the changes merges the source nodes for the matches that you accepted with the target nodes in the viewpoint, as follows:

- For the source nodes that were accepted as duplicates, request items are created to delete the source node and remove the node if the deduplicated viewpoint is a hierarchy.
- Properties from the source nodes are merged into the existing target nodes for properties where the Keep column was Source.

- Nodes in hierarchies are either inserted (when shared nodes are enabled) or moved (when shared node are not enabled) to their respective parents that are specified in the source nodes if the Parent property has a Keep column of Source.
- Node links on the surviving nodes are updated as follows:
  - Node links are created on the surviving node for the source nodes that were merged into it.
  - If a duplicate node that is merged already has a node link from another data source, the node link from the original source is updated to point to the surviving node instead.

After you click **Apply Changes**, the match candidates that have been accepted or rejected are removed from the match results table.

> **Tip:**
>
> To view the candidates in a request that were merged, inspect the request, navigate to the **Details** tab, and change the request filter to **Merged**. See Inspecting Requests.

Applying changes from the match process updates the request items in a request, but no changes are committed until the request itself is completed and closed. You can continue to modify the request items (see Inspecting Request Items) before submitting the request.

Discarding Deduplication Results

Click the **Discard** button on the deduplication results screen to discard all deduplication results that haven't been applied yet. This action removes the deduplication result set from the Deduplication Request Items page and deletes all accepted, rejected, and pending matches for the selected node type and data source. See Deduplication Operations.

Deduplication results that have already been applied are not discarded. You can review and delete those merged request items from the request itself (see Inspecting Request Items).

Rerunning Matches

In the deduplication result set, click **Actions** ⋮ and select **Rerun** to rerun the matching process for this node type and data source. This can be helpful if you have updated your matching rules or you have made changes to the source nodes that were already matched. See Deduplication Operations.

12
Working with Request Activity

You can browse requests in the Request Activity list.

In the Request Activity list, you can quickly see information about a request including the associated view, number of items in the request, and age of the request. The Request Issues column displays the number of errors and warnings in the request (including validation warnings in completed requests).

Filtering the Request Activity List

You can filter the request list by:

- **Request ID**: Enter a request ID.
- **View**: Select one or more views, up to a maximum of 15. If your drop down list contains fewer than 15 views, you can click **Select All** ☑ to quickly select them all.

> **Note:**
>
> If the view that you select has time labels associated with it, use the Time Label drop down to filter the request worklist to the selected time labels only. The Time Label filter is available when a single view has been selected.

- **Request Type**
  - All: All interactive and subscription requests
  - Interactive: Requests created manually by users.
  - Subscription: Requests generated from a subscription.
  - Import: Requests generated from an import.
  - Consolidation: Requests generated by request consolidation
  - Load: Requests generated by viewpoint loads in merge mode
- **Status**
  - All: Requests of all status types
  - Draft: Draft requests that you created. You cannot see or modify draft requests that you did not create.
  - In Flight: Requests submitted and awaiting approval
  - Pushed Back: Requests modify and resubmit
  - Future Dated: Requests that have been assigned a future dated time label.
  - Completed: Requests that have been committed and can no longer be edited.
  - Rejected: Requests that were rejected and closed.
  - Blocked: Requests that are currently being held pending the end of an application blockout period.

- **Stage**
  - All: Requests that are in the Submit, Approve, and Closed stages
  - Submit: Requests that have not been submitted (Draft status)
  - Approve: Requests that have been submitted and have not yet been approved (In Flight status)
  - Closed: Requests that have been approved and committed or requests that have been rejected (Completed status)
  - Commit: Requests that have been approved and have not yet been committed (In Flight status)
- **Priority**
  - All: Requests that have None, Low, Medium, or High priority
  - None: Requests that do not have a priority set on them
  - Low: Requests that have a Low priority
  - Medium: Requests that have a Medium priority
  - High: Requests that have a High priority
- **Owner**: Specify one or more request owners (either the submitter or assignee), up to a maximum of 15, to display only the requests owned by those users. If your drop down list contains fewer than 15 owners, you can click **Select All** ☑ to quickly select them all.
- **My Activity**
  - All: All requests that you submitted, managed, were assigned to, were invited to, or participated in.
  - Assigned: Requests assigned to you.
  - Collaborated: Requests for you to collaborate on.
  - Submitted: Requests you've submitted.
  - Invited: Requests where you are currently invited to approve or commit.
  - Contributed: Requests that you have either made changes to (changed data, performed a workflow action, or edited a comment or an attachment), or were previously invited to approve or commit (even if you did not take any action on the request).
  - Managed: Requests containing data objects for which you have *Data Manager* permission or higher
- **Time Frame**: Select a period of time, such as Last 90 Days, or enter a date range.

Saving Request Activity Filters

You can save your filter selections so that you can quickly view specific request activity on a repeated basis. Saved filters are only available to the user who saved them.

To save your filters:

1. In the filter header, click the **Saved Filters** menu ⋮, and then click **Save**.
2. Enter a name and click **Save**.

After you save your filters, the name of the saved filters is displayed in the **Saved Filters** panel and in the Request Activity header.

From a saved filter, use the **Saved Filters** menu ⋮ to perform one of the following tasks:

- **Save**: Saves the filter
- **Save As**: Saves the filter with a different name.
- **Add Favorite**: Adds the saved filter to your Favorites list. Saved filters in your Favorites list are indicated by a ★ icon. You can remove a saved filter from your Favorites list by selecting **Undo Favorite**. See Working with Favorites.
- **Delete**: Deletes the saved filter.

To clear the selection of a saved filter, click **Reset All Filters** ⊘.

Request Activity List Actions

From the request activity list, you can:

- **Inspect** a request at any stage to display the request item details of the proposed or the completed changes. You cannot edit anything in the request when you are inspecting it.

  To inspect a request, in the **Actions** column for the request, click ⋮, and then click **Inspect**.
- **Open** open a draft, in flight, and pushed back requests to make additional changes, correct validation issues, and submit the request.

  To open a request, click the request that you want to work with, or in the **Actions** column click ⋮, and then click **Open**.

  The request and the associated view open.
- **Delete** a draft or pushed back request if you no longer need it.

  To delete a request, in the **Actions** column for the draft request, click ⋮, and then click **Delete**.
- **Download** request information to a file. You can download an individual request, summary information for multiple requests, or detailed information for multiple requests. See Downloading Requests to Files.

Selecting a Column Set for Request Activity

Click **Select Column Set** ▥ and select one of the following options to change the set of columns that are displayed in the Request Activity table.

- **General**: Basic information about the request, such as the number of request issues, the request age, and the request owner.
- **Workflow**: Workflow information about the requests, such as request invitees, approvals, and commits.
- **All**: Both general and workflow information.

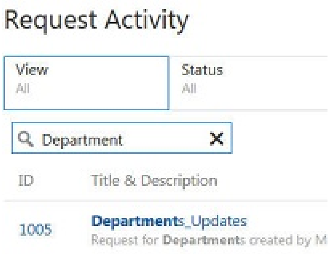
Searching for Requests

You can search for specific requests in the Requests list.

You can search for these attributes of a request:

- **Request ID**—Generated by the system when you create a draft request. The request ID; for example, "Request 1006", is included with the default title for the request.
- **Request Title**—The default title is generated by the system and includes the request ID. You can change the title of a request. Note that the request ID does not change when you edit the request title.
- **Request Description**—Generated by the system using the name of the view and the user’s name. For example, if the view for the request is Departments, the description may be "Request for Departments created by Maria Kennedy".

Your search can be for part of the ID, title, or description. For example, you know that the request you're looking for made changes to the Departments view. You can search for the word Department and narrow down the list of requests to only those that have the word Department in the ID, title, or description.



Request Activity

View: All
Status: All

🔍 Department ✕

ID	Title & Description
1005	Departments_Updates Request for Departments created by M

> **Note:**
>
> Your search is restricted to the requests that are currently displayed in the Requests list. The list displays up to 200 requests.

To search for requests:

1. From **Requests**, click 🔍.
2. Enter text to search for, and then press **Enter** or click 🔍.

> **Note:**
>
> To return to the unfiltered list of requests, delete the text from the search box, and then press **Enter** or click X in the search box.

Opening and Editing Draft Requests

You can open a draft request to add or edit request items, make comments, fix validations issues, and submit it.

You can inspect draft and completed requests to view more details about the request, see Inspecting Requests. You can also delete draft requests if you no longer need them, see Deleting Requests.

From a View

To open a request:

1. Open the view associated with the request.
2. Click **Requests** ☰.
3. Click the name of the draft request that you want to open.

From Request Activity

To open a request, click the draft request that you want to work with. The draft request and the view associated with the request open.

Deleting Requests

You can delete draft, pushed back, and recalled requests that you are the request owner for. When you delete a request, any comments or attachments associated with the request are also deleted. You can also delete multiple requests at one time.

> **Note:**
>
> Service administrators can specify that requests that were already submitted cannot be deleted (see Request Settings). You are prompted to close instead of deleting these requests.

From a View

To delete a request:

1. Open the view associated with the request.
2. Click **Requests** ☰.
3. To delete, click ⋮ to the right of the request name, and then select **Delete**.

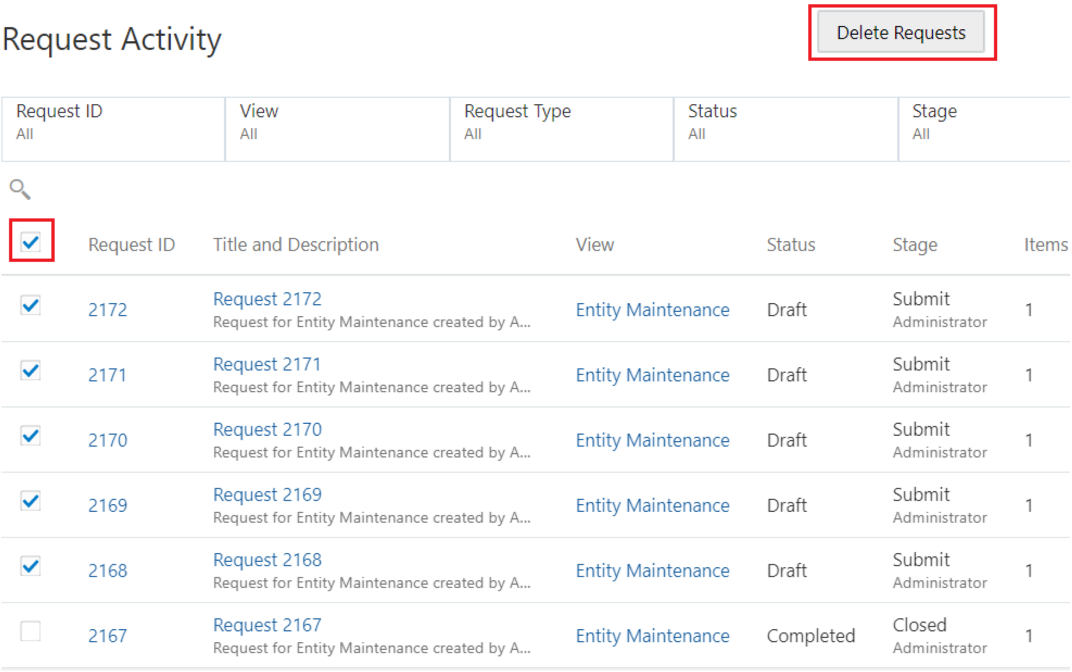
From Request Activity

From the Request Activity list, you can delete individual or multiple requests. You can delete requests in Draft, Pushed Back, or Recalled status that you are the request owner for.

- To delete an individual request:
  1. In the **Actions** column for the request, click ⋮ and then click **Delete**.
  2. Click **Yes** to confirm the deletion.
- To delete multiple requests:
  1. Select the check box next to individual requests that you want to delete or click the check box in the header to select all requests. You must be the request owner in order to delete a request.

> **Note:**
>
> When you select all requests, only requests that you own are selected.

2. Click **Delete Requests**.
3. Click **Yes** to confirm the deletion.



Request ID	View	Request Type	Status	Stage
All	All	All	All	All

☑	Request ID	Title and Description	View	Status	Stage	Items
☑	2172	Request 2172 Request for Entity Maintenance created by A...	Entity Maintenance	Draft	Submit Administrator	1
☑	2171	Request 2171 Request for Entity Maintenance created by A...	Entity Maintenance	Draft	Submit Administrator	1
☑	2170	Request 2170 Request for Entity Maintenance created by A...	Entity Maintenance	Draft	Submit Administrator	1
☑	2169	Request 2169 Request for Entity Maintenance created by A...	Entity Maintenance	Draft	Submit Administrator	1
☑	2168	Request 2168 Request for Entity Maintenance created by A...	Entity Maintenance	Draft	Submit Administrator	1
□	2167	Request 2167 Request for Entity Maintenance created by A...	Entity Maintenance	Completed	Closed Administrator	1

Downloading Requests to Files

There are several ways to download one or more requests to a file:

- You can download request items for a single request to a file. See Downloading a Single Request to a File.
- You can download request items for multiple requests to a file. See Downloading Multiple Requests to a File.
- You can download summary information for multiple requests to a file. See Downloading Request Summary Information to a File.

Downloading Request Summary Information to a File

You can download request summary information for the filtered list of requests on the Request Activity page.

For example, you can download a filtered list of requests on the Request Activity page to a file in order to review those requests offline.

To download request summary information to a file:

1. On the Request Activity page, set your filters to select the requests that you want to download summary information for. See Working with Request Activity.

> **Note:**
>
> The downloaded file will display the filtered request level summary information from the Request Activity page, even if that list exceeds the 500 requests that are displayed on the screen.

2. Click **Download** ⤓ and then select **Download Request Summary** to download the summary information for the requests to an Excel file. Click the linked Request ID in the Excel file to navigate to that request in Oracle Fusion Cloud Enterprise Data Management.

> **Note:**
>
> You cannot download summary information for requests that have been purged.

Downloading Multiple Requests to a File

From the Request Activity page, you can download multiple requests to a single file.

Considerations

- The requests must be in the same view.
- The user performing the download must have at least view access to all of the requests that are being downloaded.

To download requests to a file:

1. On the Request Activity page select the check box next to the requests that you want to download.
2. Click **Download** ⤓ and then select **Download Selected Request Details**.

> **Note:**
>
> The requests must be in the same view in order to be downloaded to a file.

The requests are downloaded to an Excel file. Each viewpoint in the requests is displayed on a separate tab, and on each tab the request actions are grouped by request in order by request number. The Excel file has a maximum of 12,000 request items.

> **Tip:**
>
> Click the linked request number in the worksheet to view the request in the request inspector.

Understanding Request Consolidation

From the request activity page, you can select multiple requests that are still in flight and consolidate them into a single combined request.

Consolidating requests enables you to periodically aggregate all of the changes across multiple requests into a single request in order to review and approve them in bulk.

> **Tip:**
>
> - A *consolidated* request is a source request that was combined into a consolidation request.
> - A *consolidation* request is a combined request made up of multiple consolidated source requests.

Consolidation Process Flow

The request consolidation process follows this general flow:

1. A user selects the requests to be consolidated on the Request Activity page. Requests can be selected for consolidation if they are in the same view, they have a status of In Flight, and they are not already a Consolidation type. The user must have at least *Data Manager* permission on all of the request items in all of the requests.
2. After the user clicks **Consolidate Requests**, the request items in the source requests are combined into a new consolidation request. Note the following about the consolidation process:
   - Requests are processed according to submit date. If two or more operations conflict with one another (that is, there are multiple changes to the same node across different original requests), the consolidation request will contain only the most recent change by submit date of the requests. For example, if a user submits a request to update a node description to "South", and then at a later time another user submits a request to update the description to "Southeast" to that same node, the "Southeast" update is contained in the consolidation request.
   - Request comments and attachments in the original requests are not carried forward into the consolidation request. You can navigate to the original consolidated requests to view the comments and attachments by inspecting the consolidation request.
   - The request file attachment contains a record of all request actions in all of the consolidated requests, including the users who performed them. Conflicting operations are highlighted with a note that a record with the same key was already processed. The attachment contains tabs for each viewpoint in the requests, and each tab contains links to navigate to the original source requests.
   - When consolidating requests, you can add the owners of the original requests as collaborators on the consolidation request. This enables the original request owners to contribute to the consolidation request, track its progress, and reference its details.
3. The source requests are updated with a status of Consolidated, and you cannot make any changes or perform any workflow actions (such as Approve) on them. However, you can continue to add comments and attachments and you can inspect the consolidated request to view workflow actions that were taken up to the time that it was consolidated. The request inspector also contains a link to the consolidation request on the Summary tab.
4. The consolidation request status is set to Draft and the stage is set to Submit. In the request inspector, the **Consolidation** tab displays the source requests that were combined to create the consolidation request.
5. If you discard the consolidation request, the original source requests are restored and placed back in the workflow where they were when they were consolidated. See Discarding a Consolidation Request

Consolidating Requests

Request consolidation enables you to combine multiple in flight requests into a single consolidated request. You consolidate requests from the request activity page.

Considerations

- All of the requests to be consolidated must be in the same view.
- All of the requests to be consolidated must have a status of In Flight.
- None of the requests to be consolidated can have a request type of Consolidation.
- Requests that have been assigned a future time label can only be consolidated with other requests that use the same time label. See Working with Future Dated Requests. When you consolidate requests that use the same time label, the consolidation request that gets created is also assigned that time label. You can also assign a future time label to a consolidation request.
- You must have at least *Data Manager* permission for all of the actions in all of the requests to be consolidated.

To consolidate requests:

1. From Request Activity, select multiple requests that meet the criteria in the Considerations section, above.

   > **Tip:**
   >
   > Ensure that you have selected all of the requests that you want to review in a consolidation request. You cannot add more requests to an existing consolidation request. To add another source request to a consolidation request, discard the consolidation request and create a new one with all of the source requests included. See Discarding a Consolidation Request.

2. Click **Consolidate Requests**, and then optionally select the check box to include the original request owners as collaborators on the consolidation request. Then, click **Yes** to confirm.
   The source consolidated requests are marked as Consolidated, and the new consolidation request is available in Request Activity.

After you consolidate requests, the following actions take place:

- The source requests are marked as Consolidated, locked for changes, and workflow on them is stopped.
- A request history item is added to each of the consolidated requests indicating the consolidate action, as well as the user and timestamp.
- The source requests are combined into a single consolidation request. The new consolidation request status is set to Draft and the stage is set to Submit.

> **Note:**
>
> A consolidation request has a limit of 12,000 request items.

Request Consolidation and Workflow

When source requests are consolidated into a consolidation request, the following workflow actions take place:

For consolidated requests:

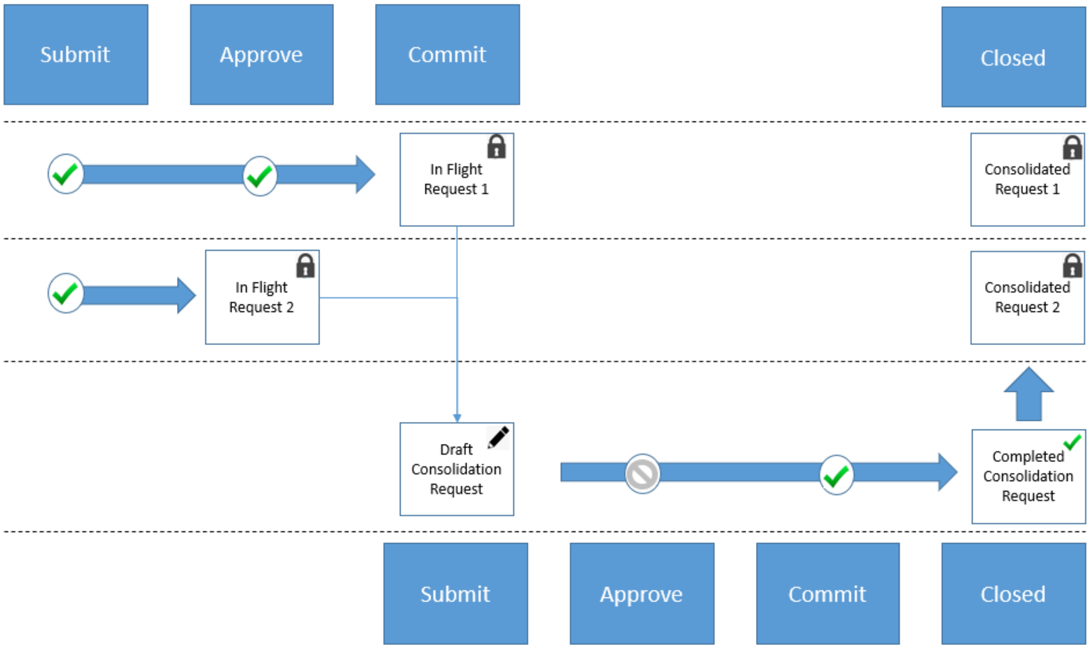
- Source request owners are notified that their request was consolidated. They are also notified if the consolidation request is completed, rejected, or discarded (see Discarding a Consolidation Request.
- The status of the source requests is updated to Consolidated. The request items in the original requests become read-only and cannot be edited.
- No additional request actions can be added to the original requests.
- Any existing validation errors are retained in the original requests until they are closed.
- Source requests are no longer evaluated for unfulfilled policies. Existing workflow history is preserved.

For the consolidation request:

- When the consolidation request is created, the request status is Draft and the stage is Submit. You can perform any actions on the consolidation request that you can perform on any other draft request, except Delete (use Discard, instead, see Discarding a Consolidation Request). For example, you can:
  - Add, edit, and delete request items.
  - Push back, recall, or reject the request. If you reject the consolidation request, the stage of all of the original requests is updated to Closed and the request owners are notified. The request status of the original requests remains Consolidated.
  - Submit the request. The request then goes into the Approval or Commit phase based on the policies associated with the request items in the request. Only approval and commit policies with Consolidation in the Request Type are evaluated for consolidation requests. See Creating and Enabling Approval Policies or Creating and Enabling Commit Policies.
- In addition to the normal draft request actions listed above, you can also discard a consolidation request. This deletes the consolidation request and returns all of the original requests to their previous state. See Discarding a Consolidation Request.

The following diagram provides an example of a consolidation process workflow:

1. Two in flight requests from the same view are displayed; one in the Commit stage and one in the Approve stage.
2. The requests are consolidated, and workflow on each source request is stopped. The consolidated requests are no longer able to be edited (lock icon).
3. The consolidation request is created in Draft status and a new governance workflow is started. Polices that are configured to include the Consolidation request type (see Creating and Enabling Approval Policies or Creating and Enabling Commit Policies) are included in the workflow.
   In this example, the consolidation request policy does not have an Approve stage, so when the request is submitted it goes to the Commit stage.
4. When the commit policy is fulfilled, the consolidation request is completed and closed, and both consolidated requests are closed as well.



Discarding a Consolidation Request

Consolidation request owners can discard a consolidation request in order to revert the consolidation and return the original requests to their previous state.

You can discard a consolidation request for a number of reasons. For example, you may want to include additional source requests in the consolidation request, or remove one or more source requests from the consolidation request, or you may want the source requests to go through some additional workflow (to get an additional approval as a source request before consolidating, for instance).

> **Note:**
>
> You can discard a consolidation request if you are the request owner and the request is in the Submit stage.

When you discard a consolidation request, the following actions take place:

- The original source requests are returned to their previous status of In Flight and the request owners are notified.
- Current policies for the source requests are evaluated and workflow proceeds as normal.
- The consolidation request is deleted.
- A request history item is added to each of the source requests.

To discard a consolidation request, perform an action:

- From Request Activity, select a consolidation request, click **Actions** ⋮, select **Discard**, and then select **Yes** to confirm.

- From the consolidation request, click **Actions** ⋮ next to the request title, select **Discard**, and then select **Yes** to confirm.

Inspecting Requests

You can inspect a request to view more details about it. If you are an approver or a committer for a request, you can perform approval or commit actions while inspecting the request.

Inspecting a Request From a View

> **Note:**
>
> Draft and In Flight requests are available from within a view. Completed requests are available in the Requests list.

To inspect a request from a view:

1. Open the view associated with the request.
2. Click **Requests** ▤.
3. Click ⋮ to the right of the request name, and then select **Inspect**.

Inspecting a Request From the Requests List
To inspect a request from the Requests list:

1. Click **Requests**.
2. In the **Actions** column for the request, click ⋮, and then click **Inspect**.

When you inspect a request you see:

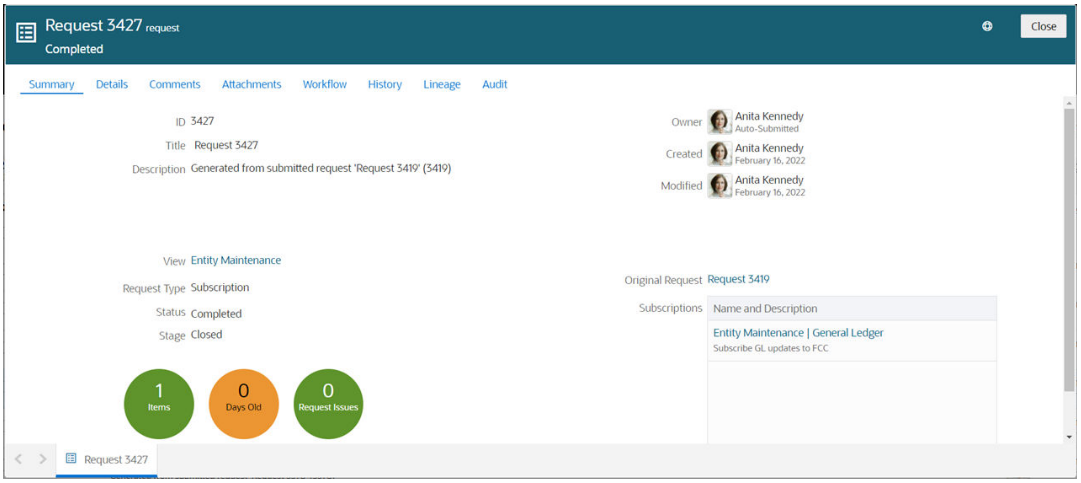
- **Summary**: Displays information about the request including the type of request, the number of request items, number of days since the request was created, and the number of issues with the request.

  > **Note:**
  >
  > The number of issues includes both warnings and errors. The color of the icon is based on the highest severity of the validation issues.

  If the request has been assigned a future time label, the status is updated to "Future Dated" and the time label name, date, and time is displayed in the **Time Label** field. After a future dated request has been completed and closed, the time label information is displayed in **Time Label Timestamp** field. See Working with Future Dated Requests.

  If the request was generated from a subscription, the original request title and the subscriptions are displayed as well. Click the request title or a subscription title to open them in the inspector to get more details about how the subscription request originated.

  If the request was automatically submitted as part of a subscription or an import, the Owner field displays "Auto-Submitted".



ID 3427
Title Request 3427
Description Generated from submitted request 'Request 3419' (3419)

Owner Anita Kennedy Auto-Submitted
Created Anita Kennedy February 16, 2022
Modified Anita Kennedy February 16, 2022

View Entity Maintenance
Request Type Subscription
Status Completed
Stage Closed

Original Request Request 3419
Subscriptions Name and Description
Entity Maintenance | General Ledger
Subscribe GL updates to FCC

1 Items
0 Days Old
0 Request Issues

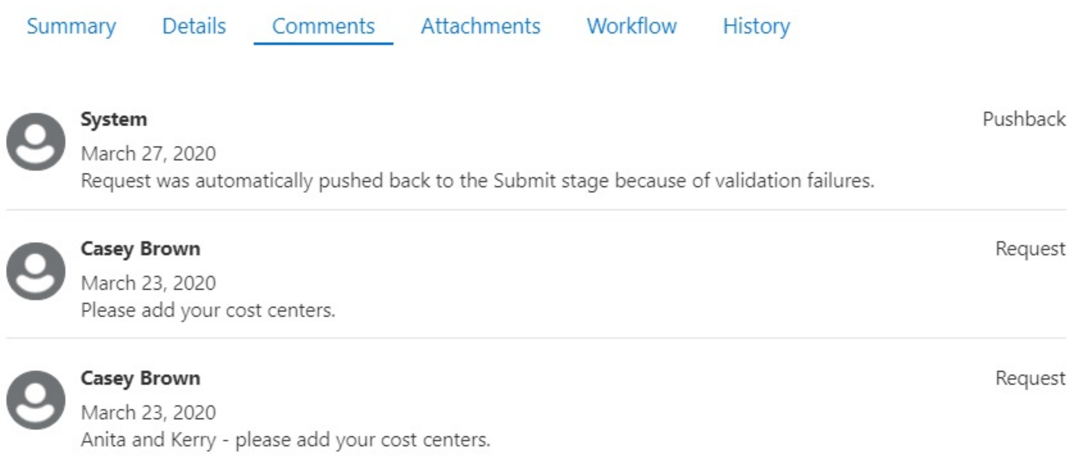
- **Details**: Displays a collapsed list of request items showing the affected node and the actions that were taken on the node, as well as the action type, node type, application, data source and source node name (for merged request items), and the person who performed the action. Expanding a request item shows what specifically was done to the node, including any validation errors or warnings on each action. You can delete individual request actions on a request item or request item instance. See Inspecting Request Items.

  To view request items that were merged as a result of a match operation (see Matching and Merging Request Items), click the **Filter** icon (▽) and select **Merged**.

  Click the **Action** ⋮ icon and select **Download to file** to download the request details to an Excel file.

  You cannot view or download request actions for requests that have been purged.

- **Comments**: Displays any comments for the request including hyperlinks if provided.



System
March 27, 2020
Request was automatically pushed back to the Submit stage because of validation failures.
Pushback

Casey Brown
March 23, 2020
Please add your cost centers.
Request

Casey Brown
March 23, 2020
Anita and Kerry - please add your cost centers.
Request

- **Attachments**: Displays load files if files were used to create the request actions or any attachments that were added to the request. You can open a file directly from here by clicking on it.

- **Policies**: The Policies tab displays the policy execution plan, which enables you to see all of the policies that affect a request, including their current fulfillment status and the order of processing for each. See Policy Execution Plan
- **Workflow**: Displays the workflow path and approval history for the request. Click the **Submit** icon to view the submitter, the date the request was submitted, and the users and groups asked to collaborate on the request. Click the **Approve** icon to view invitees and approvals for the Approve stage. Click the **Commit** icon to view invitees and commits for the Commit stage.

  > **Tip:**
  >
  > For groups, you can click the group icon to display the names of the users in the group.

  > **Note:**
  >
  > For the Approve and Commit stages, users that were invited because of an escalation on one or more workflow policies are indicated as such.

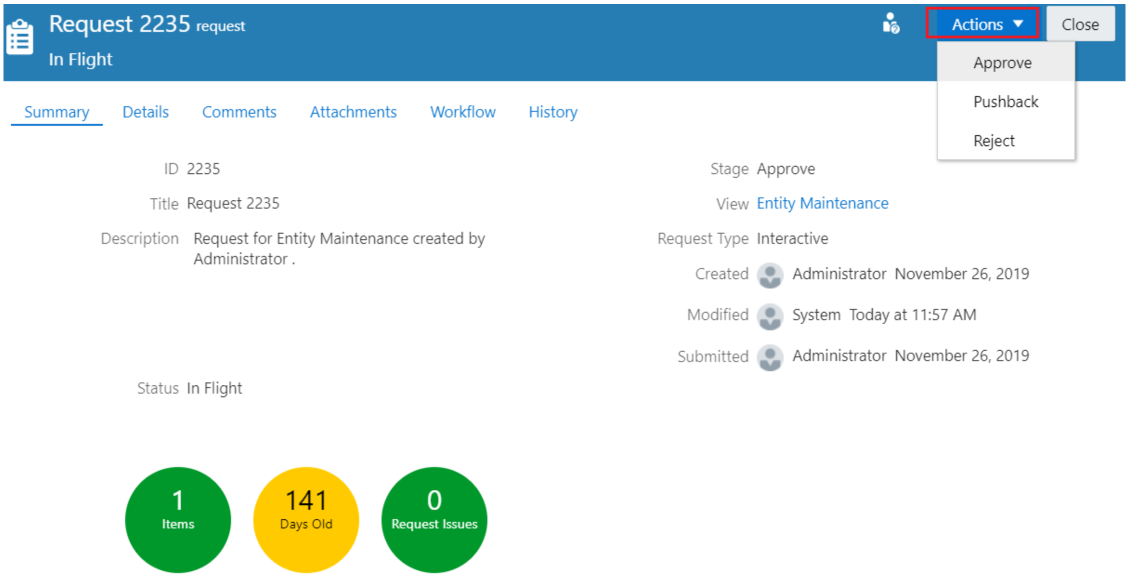
- **Consolidation**: If the request is a consolidation request, displays the source requests that were combined to create the consolidation request. See Consolidating Requests.
- **History**: Displays the history of a request including who participated, what actions were taken, and when. If there were validation warnings on any of the request items, those are also displayed.

  You can search for specific information, for example you may want to search for approval information for a request. The search looks for the text in the fields displayed on the tab.
- **Lineage**: Displays a lineage diagram for the request that shows related requests that precede and follow it. See Viewing Request Lineage.
- **Audit**: Displays the transaction history associated with a completed request. This tab is displayed for completed requests only. See Viewing the Transaction History for a Request.

Request Inspector Actions

If you are an approver or a committer of a request, you'll see an Actions drop-down giving you the ability to approve (or commit), push back, or reject the request from the request inspector. This allows you to act on a request directly from Request Activity instead of opening the request to visualize the changes in the view.

This is useful when a request is being approved or committed from a summary point of view, when an approver or committer’s permissions do not allow access to the view where it was submitted, or when an approver or committer is responsible for approving the node type but not the hierarchy.



ID 2235

Title Request 2235

Description Request for Entity Maintenance created by Administrator .

Status In Flight

Stage Approve

View Entity Maintenance

Request Type Interactive

Created Administrator November 26, 2019

Modified System Today at 11:57 AM

Submitted Administrator November 26, 2019

From the **Actions** menu, select an action:

- **Approve** Optionally enter a comment and then click **OK** to approve the request.
- **Commit** Optionally enter a comment and then click **OK** to commit the request.
- **Pushback** Enter a comment, press Tab, and then click **OK** to push the request back to the submitter for changes.
- **Reject** Enter a comment, press Tab, and then click **Reject** to reject the request.

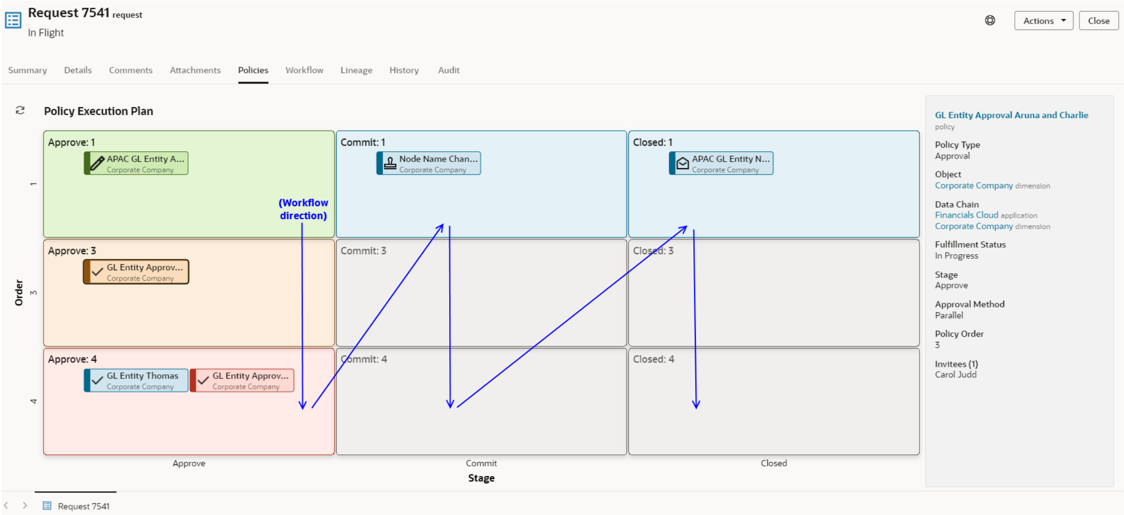
For more information, see Approval Process or Commit Process.

Policy Execution Plan

The Policy Execution Plan enables you to see all of the policies that affect a request, including their current fulfillment status and the order of processing for each.

This information can be helpful when visualizing complex governance scenarios for requests. For example, data managers may use the policy plan to determine where a request is in the workflow process and which policies are remaining. Owners can use the plan to appropriately configure workflows for particular types of requests.

You access the Policy Execution Plan from the Policies tab on the Request Inspector.



GL Entity Approval Aruna and Charlie
policy

Policy Type
Approval

Object
Corporate Company dimension

Data Chain
Financials Cloud application
Corporate Company dimension

Fulfillment Status
In Progress

Stage
Approve

Approval Method
Parallel

Policy Order
3

Invitees (1)
Carol Judd

Workflow Direction

As indicated by the blue arrows in the image above, the workflow in the Policy Execution Plan starts at the top of the first column (in this case, with Approve in policy order 1) and continues down the first column. When the last policy in the first column has been fulfilled, the first policy in the second column (in this case, Commit in policy order 1) is processed, and when the policies in the second column have all been fulfilled, the first policy in the third column is processed, and so on.

Policy Types and Fulfillment Statuses

The icons and colors in the Policy Execution Plan show the type and fulfillment status for each policy, as follows:

Policy Types

- **Approve**: ✓
- **Enrich**: ✎
- **Commit**:
- **Notify**: ✉

Fulfillment Statuses

- **Green**: Fulfilled
- **Brown**: In Progress
- **Blue**: Future
- **Red**: Error
- **Grey**: Incomplete

> **Note:**
>
> An incomplete fulfillment status is displayed for policies where the request was rejected or consolidated before the policy could be fulfilled. (for example, a Commit policy that was not fulfilled because the request was rejected in the Approve stage).

Policy Details Pane

Click the name of a policy to view the details in the Policy Details pane. This pane enables you to click the name of the policy or the data chain objects to view the policy or data chain objects in the inspector.

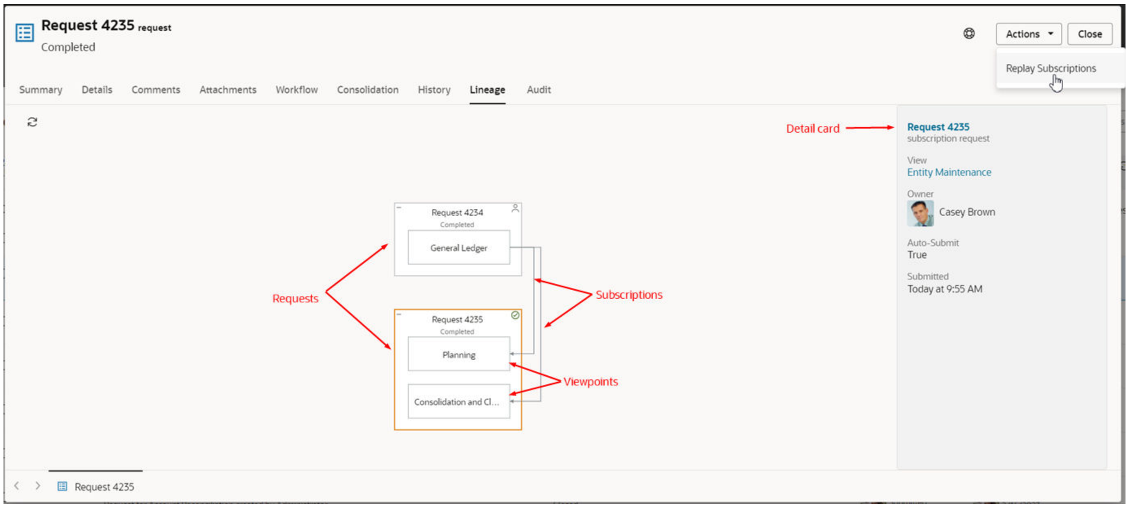
The details pane also displays information about the policy such as the status, the approvals or commits that have already been granted, current invitees, and the users and groups who will be invited in the future. For approval and commit policies, users that were invited because of an escalation on one or more workflow policies are indicated as such.

If there are errors on a policy (for example, a group with no active users assigned to it), that information is displayed in the details pane as well so that you can take corrective action.

Viewing Request Lineage

The Lineage tab of the request inspector displays a diagram for the request that shows related requests that precede and follow it.

The lineage diagram includes the originating request along with subscription requests which were generated from it. You can expand each request to view details such as its title, status, owner, the viewpoints that were affected, and the subscriptions (including pending subscriptions, for subscriptions that have not yet been processed) that were used to create it. You can also click the request title in the detail card to inspect any request in the lineage and access more information such as the request items, comments, attachments, request workflow, and history.



Request 4235 request
Completed

Request 4235
subscription request

View
Entity Maintenance

Owner
Casey Brown

Auto-Submit
True

Submitted
Today at 9:55 AM

Each request in the lineage tab is represented by a box, with the original interactive request at the top of the diagram. You can expand each box to display the viewpoint within the request. Subscriptions are represented by dashed arrows when the requests are collapsed, and by solid arrows when the requests are expanded and the viewpoints can be seen.

Click a request, viewpoint, or subscription to display details about them in the detail card, such as the date a completed request was submitted or the current assignee of draft requests. The detail card also contains links to get more information about that object. For example, when a request is displayed in the detail card, you can use the links in the detail card to inspect the request or the view. When a subscription is displayed in the detail card, you can use the links in the detail card to inspect the subscription, as well as the source and target views and viewpoints.

> **Note:**
>
> If a request that is part of the request lineage is deleted, a placeholder for the deleted request is displayed in the diagram and subsequent subscriptions or requests are not displayed. Click on a deleted subscription request to view additional information about the deleted request.

You can reposition and zoom in and out of the digram by using your mouse. Click **Refresh** ⟳ to restore the original diagram or to update the status of a subscription currently in progress.

The diagram contains icons that display the origin or outcome of the requests. The following table describes the icons in the lineage diagram:

Table 12-1 Request Lineage Icons

Icon	Meaning
+	Expand request
–	Collapse request
👤	Request was created interactively
▦	Request was created via subscription
↙	Request was created via import
✓	Subscription request was auto-submitted
🗑	Request was deleted

Table 12-1   (Cont.) Request Lineage Icons

Icon	Meaning
ⓘ	A subscription request failed or was skipped

Failed or Skipped Subscriptions

For subscription requests that have failed or were skipped, the lineage diagram displays a ⓘ icon in the source request for the failed or skipped subscription. The detail card provides details about the failure, such as:

- The subscription does not have an assignee available.
- The subscription was skipped because no items would have been generated.

From a completed request, Service Administrators and users with *Owner* permission on the view can replay a subscription that failed due to a configuration issue, such as a missing subscription assignee. This enables you to reprocess a subscription that did not previously run for a completed request in order to ensure data consistency between source and target viewpoints.

Considerations for Replaying Subscriptions

- When you replay a subscription, the following subscriptions are reprocessed:
  - Subscriptions that previously failed to run
  - Subscriptions that were never run, including new subscriptions
- You cannot replay a *skipped* subscription because no items would have been generated.
- It is a best practice to replay a failed subscription as close to the time that it failed as possible. Replaying stale subscriptions may cause unexpected results if the underlying data in the viewpoint has since been updated by other requests.
- When replaying a failed subscription, you should disable **Auto Submit** in order to review the changes before committing them. See Creating, Editing, and Validating Subscriptions.

To replay a subscription:

From the request inspector of a completed request, click **Actions**, and then **Replay Subscription**.

> **Note:**
>
> The **Replay Subscription** menu item is displayed on all completed requests. However, only requests that have failed subscriptions or subscriptions that were never run will be reprocessed when you select this action.

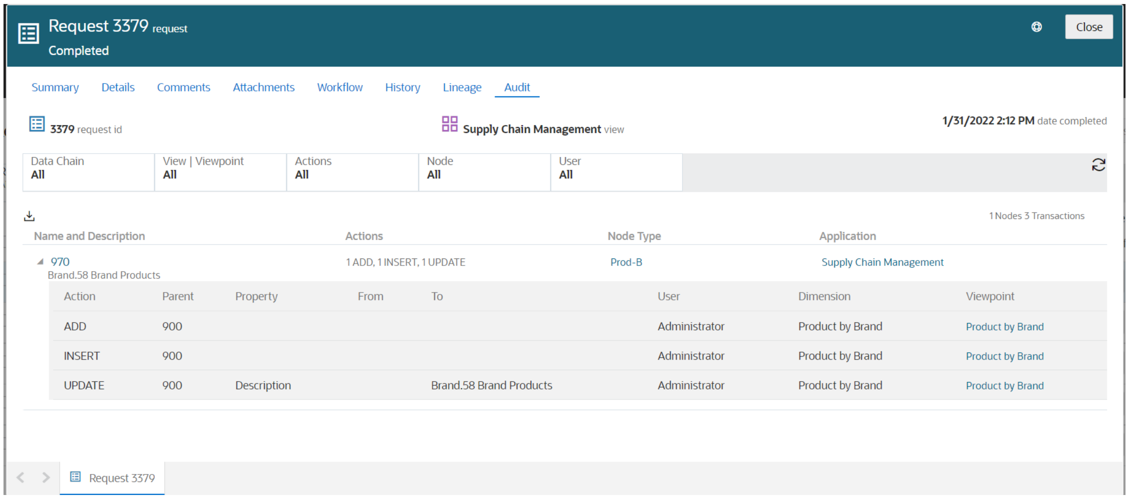
Viewing the Transaction History for a Request

You can view the transaction history for completed requests on the Audit tab of the request inspector. You can filter the results to display specific nodes, and you can download the transaction history to a file.

Considerations

- You view transaction history for a request from the Audit tab, which is displayed for completed requests only.
- If you do not have permission to view a property that was updated in the request, the From and To values of the property are not displayed on screen or downloaded to the transaction history file.

To view transaction history for a request, inspect the request and then click the Audit tab.



3379 request id
Supply Chain Management view
1/31/2022 2:12 PM date completed

Data Chain	View | Viewpoint	Actions	Node	User
All	All	All	All	All

1 Nodes 3 Transactions

Name and Description	Actions	Node Type	Application
970 Brand.58 Brand Products	1 ADD, 1 INSERT, 1 UPDATE	Prod-B	Supply Chain Management

Action	Parent	Property	From	To	User	Dimension	Viewpoint
ADD	900				Administrator	Product by Brand	Product by Brand
INSERT	900				Administrator	Product by Brand	Product by Brand
UPDATE	900	Description		Brand.58 Brand Products	Administrator	Product by Brand	Product by Brand

You can filter the transactions contained in the request that you want to view. The following filters are available:

- **Data Chain:** Select an application, dimension, node type, or hierarchy set to view the transactions in the request that were performed on that object.
- **Viewpoint:** If the request contains transactions for more than one viewpoint, select a viewpoint to display those specific transactions.
- **Actions:** Select the specific action that you want to display. The following actions are available to select. You can select more than one action to filter on.
  - Add
  - Insert
  - Move
  - Remove
  - Delete
  - Update
  - Rename
  - Reorder

- **Node:** Enter an individual node to see the transactions on that node. Only exact matches are displayed, and wildcard characters are not supported.
- **User:** Select a user name to see the transactions in the request made by that user.

When using the transaction history filters for a request, click **Reset all Filters** ⊘ to reset all filters to their default values, and click **Refresh Data** ⟳ to refresh the data returned by the current filter settings.

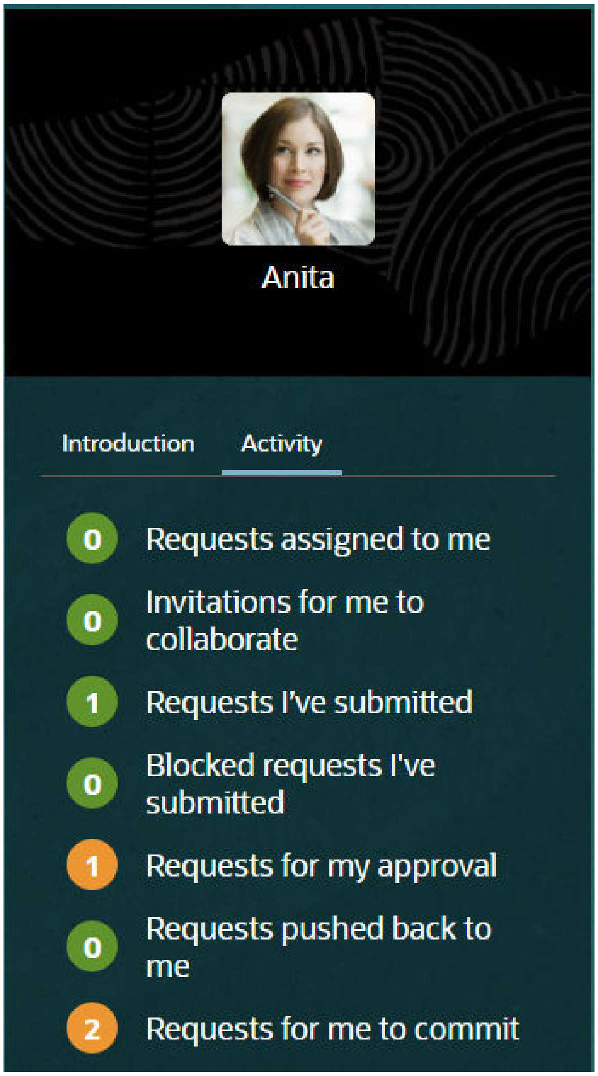
You can download the results to an Excel file by clicking **Download Request Transactions** ⤓. The download file contains all of the transaction history based on the current filtered data for a request, not just the ones expanded or displayed on screen.

Accessing Requests from the Home Page

The Welcome Panel on the Home page displays request activity specific to the current user.

You can see the number of requests in these categories:

- Requests assigned to me
- Invitations for me to collaborate
- Requests I've submitted
- Blocked requests I've submitted
- Requests for my approval
- Requests pushed back to me
- Requests for me to commit



0 Requests assigned to me

0 Invitations for me to collaborate

1 Requests I've submitted

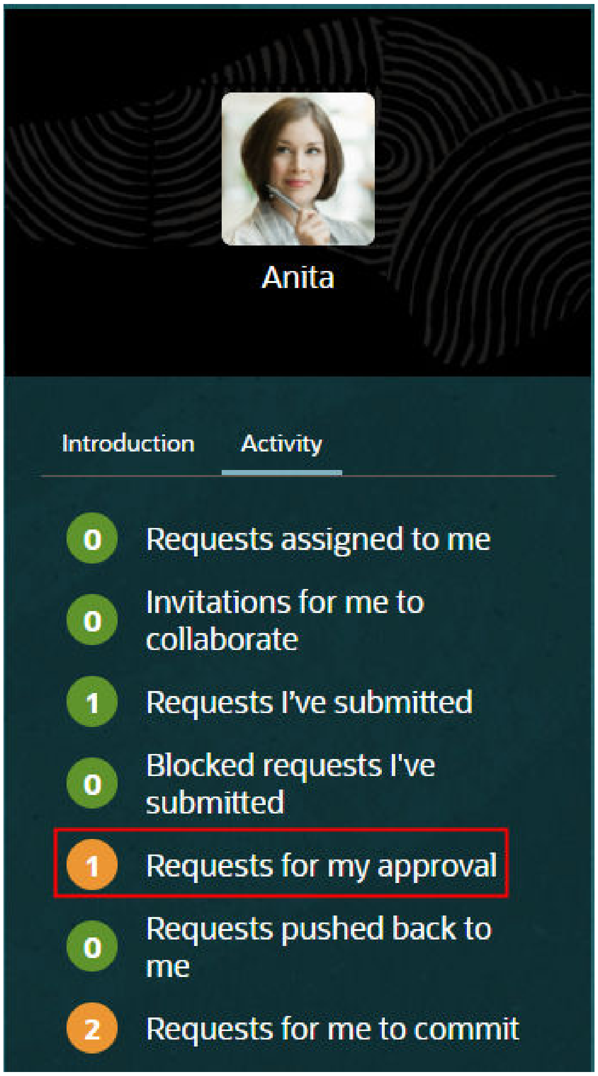
0 Blocked requests I've submitted

1 Requests for my approval

0 Requests pushed back to me

2 Requests for me to commit

1. Click on one of the categories in the Activity list to go to the Request Activity page where you can work with a specific request.

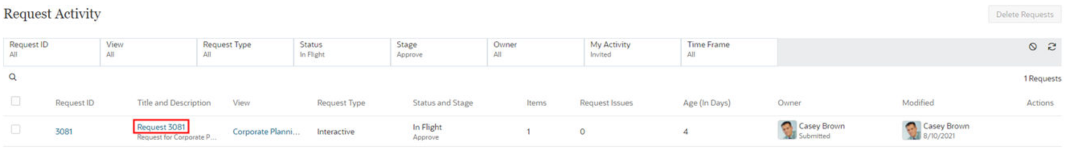


Anita

Introduction Activity

- 0 Requests assigned to me
- 0 Invitations for me to collaborate
- 1 Requests I've submitted
- 0 Blocked requests I've submitted
- 1 Requests for my approval
- 0 Requests pushed back to me
- 2 Requests for me to commit

2. Open the request by clicking on it.



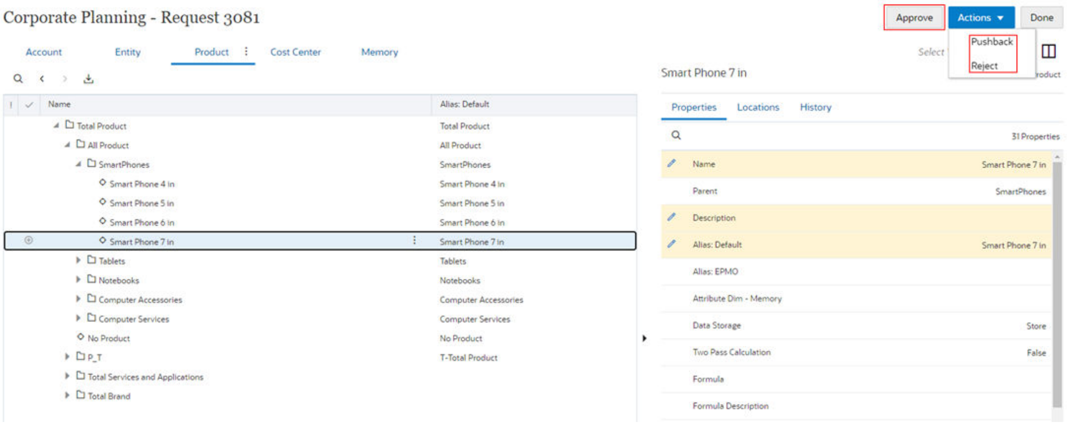
Request Activity

Request ID	View	Request Type	Status	Stage	Owner	My Activity	Time Frame
All	All	All	In Flight	Approve	All	Invited	All

1 Requests

□	Request ID	Title and Description	View	Request Type	Status and Stage	Items	Request Issues	Age (In Days)	Owner	Modified	Actions
□	3081	Request 3081 Request for Corporate P...	Corporate Planni...	Interactive	In Flight Approve	1	0	4	Casey Brown Submitted	Casey Brown 8/10/2021	

3. Review the change and decide to approve, reject, or push back the request.



Name	Alias: Default
Total Product	Total Product
All Product	All Product
SmartPhones	SmartPhones
Smart Phone 4 in	Smart Phone 4 in
Smart Phone 5 in	Smart Phone 5 in
Smart Phone 6 in	Smart Phone 6 in
Smart Phone 7 in	Smart Phone 7 in
Tablets	Tablets
Notebooks	Notebooks
Computer Accessories	Computer Accessories
Computer Services	Computer Services
No Product	No Product
P_T	T-Total Product
Total Services and Applications	
Total Brand	

Name	Smart Phone 7 in
Parent	SmartPhones
Description	
Alias: Default	Smart Phone 7 in
Alias: EPMO	
Attribute Dim - Memory	
Data Storage	Store
Two Pass Calculation	False
Formula	
Formula Description	

13 Governance Workflows and Approvals

Data governance represents the convergence of data quality, data management, data policies, business process management, and risk management surrounding the handling of information as an asset within the enterprise. In Oracle Fusion Cloud Enterprise Data Management, organizations use request workflows to exercise positive control over the processes and methods used by their data stewards and data custodians to create and maintain high quality enterprise data assets.

Request workflows address the governance challenge in a number of critical ways:

- Requests model changes based upon a set of items and actions. Items within a request are validated, approved, and committed together. This creates integrity in change management and enables change control.
- Workflow approvals are inference-based. Configure one or more approval policies at the application, dimension, hierarchy set, or node type level. Workflow orchestrates the invitation of approvers while executing approval policies concurrently to achieve high quality outcomes.
- Workflows adhere to the "four eyes" principle by default. Approval policies are configured by default to exclude submitters as approvers unless otherwise specified.
- Workflows can be implemented across multiple business contexts.
  - Configured within an application view to ensure application, dimension, or node/hierarchy level approvals.
  - Implemented in a maintenance view to secure approval for related changes across application contexts.
  - Triggered as part of subscription requests to simulate application dimension-level enrichment and approval stages across multiple application contexts.

> **Note:**
>
> If you subscribe to changes in another viewpoint which uses an approval policy, the changes are applied to the subscribing viewpoint when the request is approved. If the subscribing viewpoint is configured for enrichment, then the subscription assignee can modify the request. See Subscribing to Viewpoints.

Videos

Your Goal	Watch This Video
Get familiar with workflow approvals	▶ Overview: Workflow Approvals in Enterprise Data Management Cloud

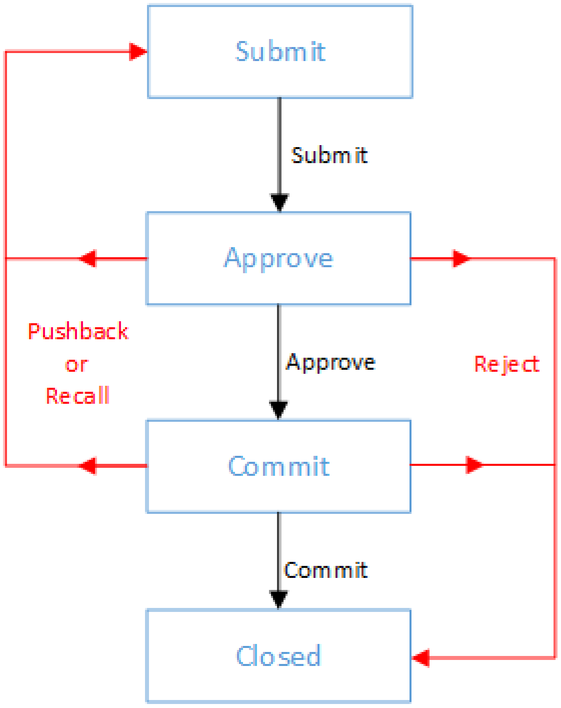
Your Goal	Watch This Video
Understand how to use subscriptions in workflows	Incorporating Subscriptions Into Approval Workflows in Oracle Enterprise Data Management Cloud

Complete this tutorial for a hands-on example of the workflow approval process.

Workflow Approvals in Oracle Enterprise Data Management Cloud

Workflow Stages

Request workflow stages, along with the actions that move a request to a different stage, are shown in the following diagram.



Requests move through four stages: Submit, Approve, Commit, and Closed. The Approve and Commit stages are optional stages that can be enabled by creating approval and commit policies. See Configuring Policies.

The following table describes the tasks that can be performed in each workflow stage.

Request Stage	Task	See
Submit	Create requests Add, update, delete request items	Making Changes Using Requests
	Add comments and attachments	Working with Request Comments Attaching a File to a Request

Request Stage	Task	See
	Assign, collaborate, submit, and delete requests	Assigning a Request Collaborating on a Request Submitting a Request Opening and Deleting Draft Requests
Approve	Approvers are invited to approve submitted requests	Workflow Notifications
	Approve, enrich, push back, recall (request submitter only) or reject requests	Approving a Request Enriching a Request Pushing Back a Request Recalling a Request Rejecting a Request
	Add comments and attachments (whether approving, pushing back, or rejecting a request)	Working with Request Comments Attaching a File to a Request
Commit	Committers are invited to commit approved requests	Workflow Notifications
	Commit, reject, recall (request submitter only) or push back requests	Committing a Request Rejecting a Request Recalling a Request Pushing Back a Request
Closed	View request history	Inspecting Requests

About Application Blockout Periods

An application *Owner* or *Metadata Manager* may set up a blockout period on an application during which requests are held and not applied to the active view until the blockout period is over. This blockout period occurs just before the Closed stage, and all requests are held in "Blocked" status until the blockout period ends. When the blockout period is over, all blocked requests are validated and either completed and closed if there are no validation errors, or returned to the workflow if there are validation errors.

Submit Process

Users can collaborate on the submission of requests. A request can be assigned to another user who can work on and submit the request. Multiple users can also be asked to collaborate on a request.

The submit process includes:

- Creating requests
- Adding, updating, deleting request items
- Adding comments to the request or specific request items
- Attaching supporting documentation
- Assigning a request to another user to work on
- Deleting draft requests

- Submitting requests

Assigning a Request

A draft request can be assigned to another user who can continue working on the request.

For example, a request may need to be reassigned to a different user if the creator of the request is unavailable or unable to complete the submit stage for the request. The new assignee is responsible for submitting or deleting the request.

> **Note:**
>
> To be reassigned to a different user, the request must be in the Submit stage.

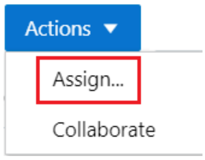
Considerations

- When assigning a request, the permissions and data access of both the assigner and the assignees of the request are taken into consideration. See Security for Requests.
- When validating a request, if the request contains an action that the new assignee does not have permission for, or updates a property (including hidden properties) that the new assignee does not have Edit permission on, that request item will fail validation due to user permissions. Any validation errors must be resolved before the request can be submitted.
- When correcting validation errors, the request assignee can delete any request items in a request, including items that contain actions that their permission does not enable them to perform or properties that they do not have permission to update.

When a request is reassigned, both the previous and new assignee are notified of the change in responsibility. The previous assignee receives an email notification informing that the request is no longer assigned to them. The new assignee receives an email notification informing that the request is now assigned to them as the person responsible for submitting the request.

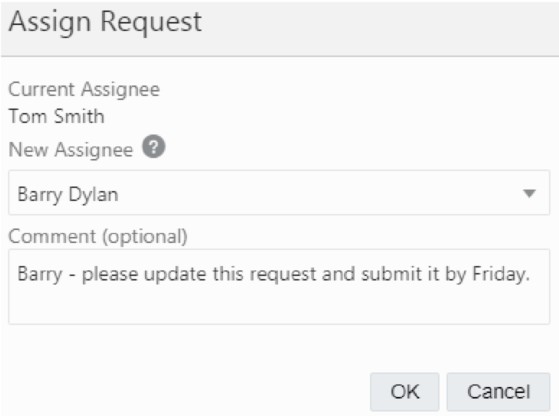
To assign a request:

1. Open the request.
2. Click **Actions** and then select **Assign**.



> **Note:**
>
> The Assign menu item is only available if the current user is able to assign the request (that is, the user must be the current request assignee, a user with *View Owner* permission to the view, or a Service Administrator).

3. Select the user to assign the request to, add an optional comment, and then click **OK**.



Assign Request

Current Assignee
Tom Smith

New Assignee

Barry Dylan

Comment (optional)

Barry - please update this request and submit it by Friday.

Collaborating on a Request

To facilitate multiple users participating on a request, the request assignee can ask additional users and groups to collaborate on Draft and Pushed Back requests in the Submit stage. Collaborators can be added or removed from a request. Collaborators are notified of requests in which they are asked to collaborate.

Users can access collaboration requests from the Activity list on the Home page and request activity. The Collaborated item in the My Activity filter can be used to find requests in which the user is a collaborator.

Collaborating users can take these actions on request items:

- Browse
- Add
- Edit
- Load
- Validate
- Delete

They can also browse, add, edit, and delete request comments as well as view, add, and delete attachments.

> **Note:**
>
> The user assigned to the request retains control over the request. Collaborators cannot submit, re-assign, or delete a request in the Submit stage.

Considerations

When inviting users to collaborate on a request, the following considerations apply:

- When collaborating on a request, the permissions and data access of both the request assignee and the collaborator are taken into consideration. See Security for Requests.
- When a collaborator validates a request, if the request contains an action that either the collaborator does not have permission for, or updates a property (including hidden

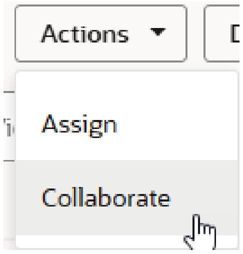
properties) that the collaborator does not have Edit permission on, that request item will fail validation due to user permissions. Any validation errors must be resolved before the request can be submitted.

- When correcting validation errors, the *assignee* can delete any request items in a request, including items that contain actions that their permission does not enable them to perform or properties that they do not have permission to update. However, the *collaborator* can delete only the request items that they have sufficient data access to perform all actions and property updates for.

After collaborators have participated, the assignee can submit the request to be approved and committed.

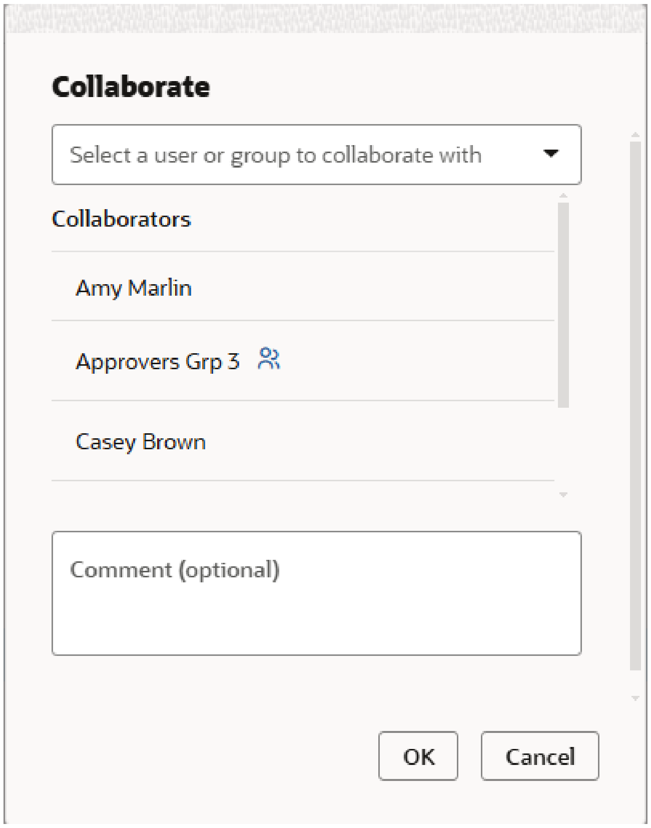
To add users to collaborate on a request:

1. Open the request.
2. Click **Actions** and then select **Collaborate**.



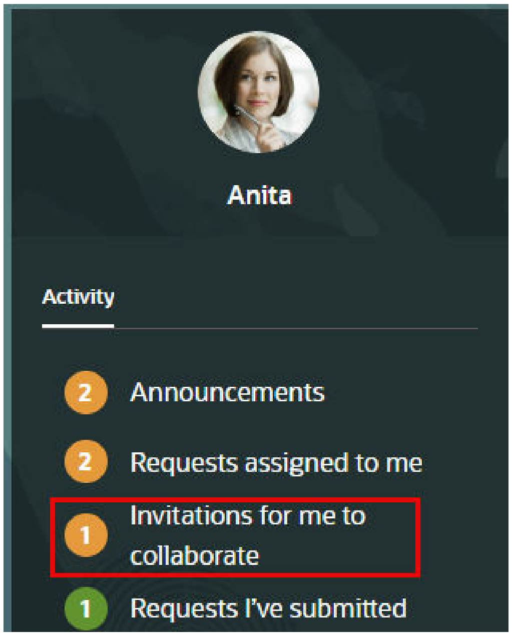
3. From the drop down menu, select users and groups, add an optional comment, and then click **OK**.

> **Note:**
>
> Only users and groups who have access to one or more viewpoints in the view will be available in the drop-down list. Groups are indicated by a group icon. Click the icon to display the members of the group.



Collaborate

Users asked to collaborate on a request receive an email notification and their activity list is updated.



Anita
Activity
2 Announcements
2 Requests assigned to me
1 Invitations for me to collaborate
1 Requests I've submitted

To remove a collaborator from a request:

1. Open the request
2. Click **Actions** and then select **Collaborate**.

3. Click the X to the right of the user's name that you want to remove and then click **OK**.

Submitting a Request

When a request is submitted, validation checks run, the changes are committed, and the viewpoint is updated with the changes.

When you submit a request, it is automatically validated:

- If there are validation errors, you must correct the errors before you can submit the request.
- If there are validation warnings only (with no errors), you must confirm that you want to continue submitting the request. Click **Continue** to submit the request despite the warnings, or click **Cancel** to review the request.

To submit a request:

1. Open a draft request.
2. Click **Submit**.

You can view completed requests in the Requests list, see Working with Request Activity.

Resubmitting a Request

Requests that have been pushed back or that you have recalled must be resubmitted for approval. If the request has been enriched, the new request items must be validated before you can submit the request.

Considerations

As the request submitter, your permissions and data access must enable you to perform every action in a request. If a request has been enriched with an action that your access does not enable you to perform, take one of the following steps to allow the request to be validated and submitted:

- Delete the request item or request item instance.
- From the request inspector, delete the individual request action that you are not able to perform.

See Working with Request Validation Issues.

To resubmit a request:

1. Open a pushed back or recalled request.
2. If the request has been enriched, review the request items and resolve any issues with validations or actions that you cannot perform by deleting request items, request item instances, or request actions.
3. Click **Submit**.

The request is moved to the Approval stage, and invitations to approve are sent out again.

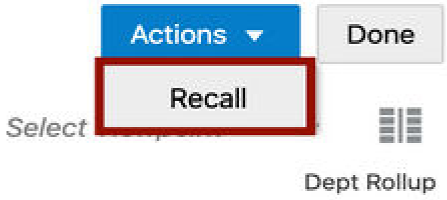
Recalling a Request

If you are the submitter of the request, you can recall an In Flight request after you submit it in order to make additional changes.

When you recall a request, any request approvals and commits are cleared and the request is returned to the Submit stage. You can modify the request items and resubmit the request (see Resubmitting a Request) or delete the request if it is no longer needed.

Existing collaborators are retained and can make changes to the request, and all other request participants continue to have access to the request. If the request has been enriched, all changes in the enrichment process are preserved.

1. On an In Flight request that you submitted, click **Actions**, and then click **Recall**.



2. Add a comment, and then click **OK**.

Approval Process

Users can collaborate on the approval of requests.

Approvers are invited by email notification to approve submitted requests. Who gets invited is based on the configuration of approval policies. For more information, see Configuring Policies.

The approval process includes:

- Adding comments to a request
- Attaching supporting documentation
- Enriching a request by adding or editing request items (if enrichment is enabled)
- Pushing back a request so that it can be updated and resubmitted
- Rejecting a request
- Approving a request
- Withdrawing an approval from a request

Approving a Request

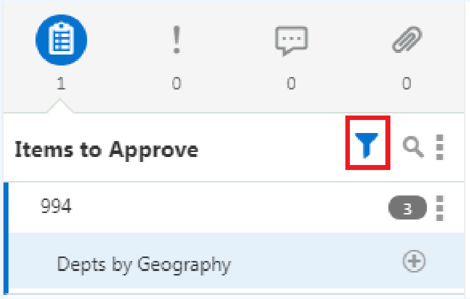
You'll receive an email notification if you've been invited to approve a request.

When you open a request to approve, you can:

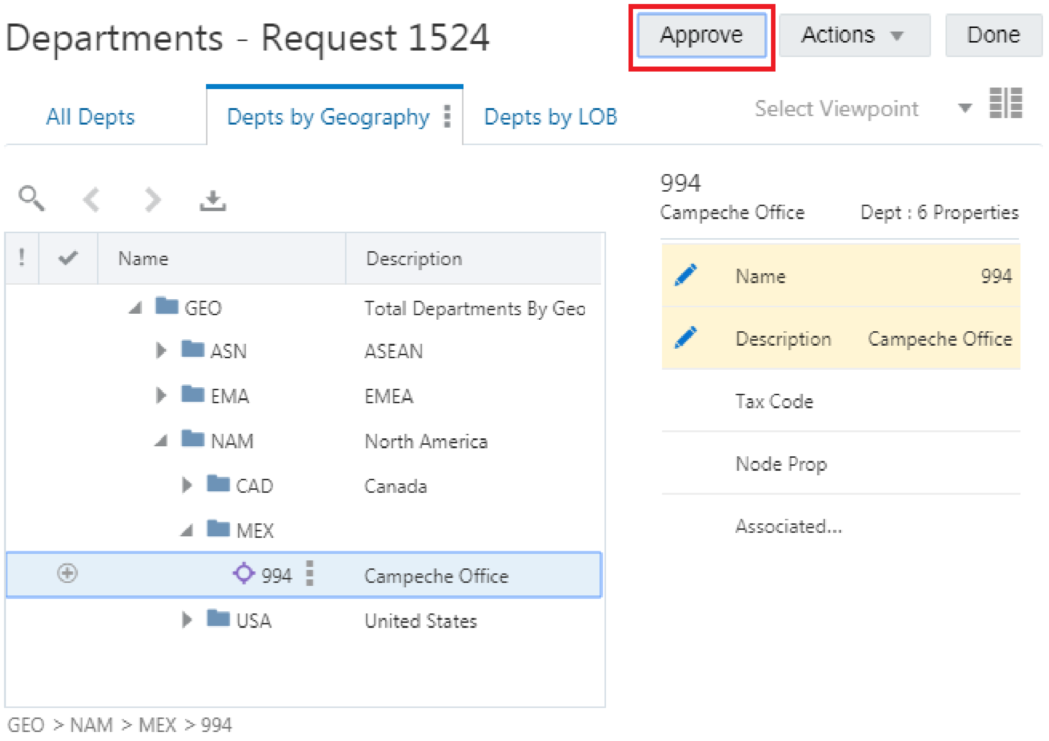
- Review the request items on the Items tab and visualize the changes in the viewpoint.
- View and add comments for specific request items on the Items tab.
- View and add comment for the request on the Comments tab.
- View and add attachments for the request on the Attachments tab.

> **Note:**
>
> If the approval policy has enrichment enabled, approvers can also enrich the request by adding, removing, and editing request items. See Enriching a Request.

1. From the email notification, click the link to open request activity and then open the request.

   > **Note:**
   >
   > If you open a request where you are only approving a portion of the changes, the **Items to Approve** filter is enabled by default and displays only the request items that you're invited to approve. You can show all request items by clicking the filter icon.
   >
   > 
   >

2. Review the changes in the request. View and add comments or attachments, and then click **Approve**.



!	✓	Name	Description
		GEO	Total Departments By Geo
		ASN	ASEAN
		EMA	EMEA
		NAM	North America
		CAD	Canada
		MEX	
		994	Campeche Office
		USA	United States

GEO > NAM > MEX > 994

994
Campeche Office Dept : 6 Properties

Name	994
Description	Campeche Office
Tax Code	
Node Prop	
Associated...	

3. You can add an approval comment and then click **OK**.

> **Note:**
>
> Validation errors must be corrected in order to approve the request. If there are validation warnings only, you must confirm that you want to continue approving the request. Click **Continue** to approve the request despite the warnings, or click **Cancel** to review the request.

Enriching a Request

If enrichment is enabled on an approval policy, you can enrich requests by adding, removing, and editing request items before you approve the request.

The request actions that you can perform depend on your permissions and data access. See Security for Requests

When you enrich a request, you can:

- Perform all of the actions that an approver can perform. See Approving a Request.
- Add new items to the request, including adding new nodes or making changes in viewpoints not previously included in the request.
- Make changes to existing request items (as allowed by your permissions), including deleting items or making changes that offset or nullify a previous action.

When you have finished enriching the request by making changes, you approve the request. The request is validated based on your permissions and data access.

Note:

If enrichment is enabled on an approval policy, you cannot approve a request that has validation issues that you can take action on. When a request has one of the following types of validation issues and you are an enricher on the approval policy, an error is displayed if you try to approve it:

- You made the change that caused the validation issue
- You have the appropriate permissions to correct a validation issue that someone else's change caused

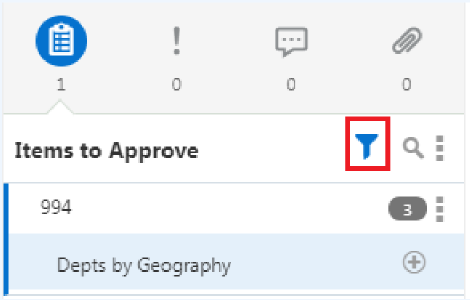
You must correct the validation issue before you can approve the request. See Working with Request Validation Issues.

If a request has a validation issue that you cannot take action on, you can approve the request.

1. From the email notification, click the link to open request activity and then open the request.

   Note:

   If you open a request where you are only approving a portion of the changes, the **Items to Approve** filter is enabled by default and displays only the request items that you're invited to approve. You can show all request items by clicking the filter icon.

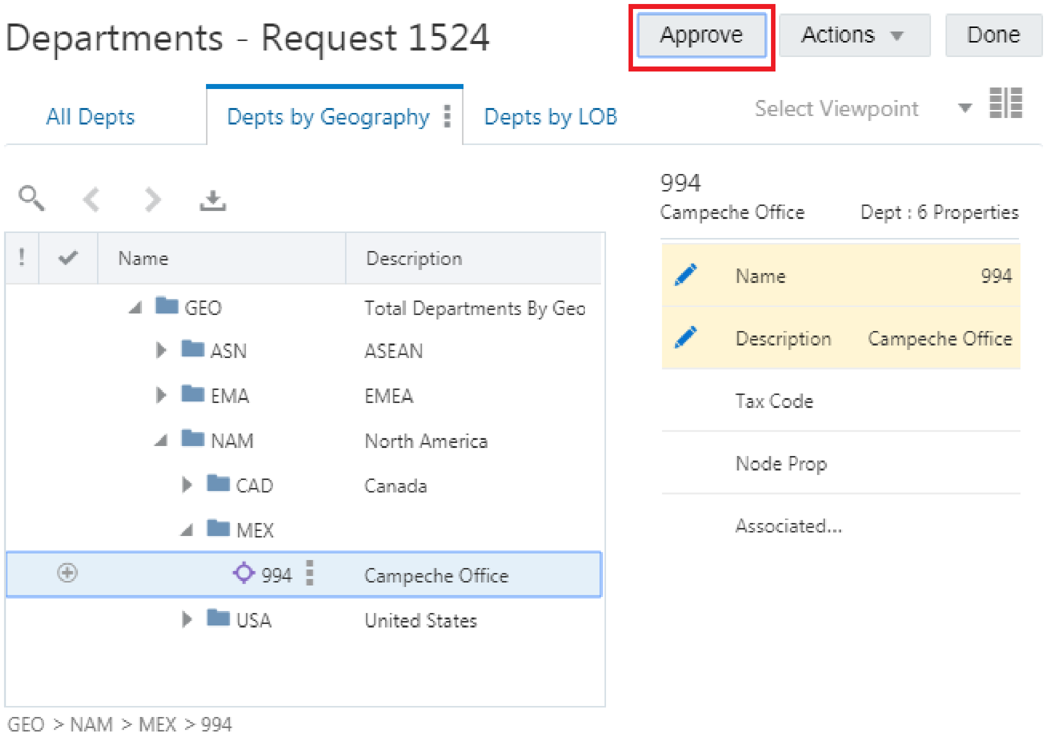
   

2. Perform request actions such as adding and removing nodes and editing properties as dictated by your permission and data access. See Making Changes Using Requests.

   Note:

   Because the **Items to Approve** filter displays only the items that you have been invited to approve, if you add request items for data chain objects that you are not an Approver on during enrichment, you must click the filter icon to disable the filter in order to see those items in the request pane.

3. Review the changes in the request. View and add comments or attachments, and then click **Approve**.



Name	Description
GEO	Total Departments By Geo
ASN	ASEAN
EMA	EMEA
NAM	North America
CAD	Canada
MEX	
994	Campeche Office
USA	United States

GEO > NAM > MEX > 994

994
Campeche Office Dept : 6 Properties

Name	994
Description	Campeche Office
Tax Code	
Node Prop	
Associated...	

4. You can add an approval comment and then click **OK**.

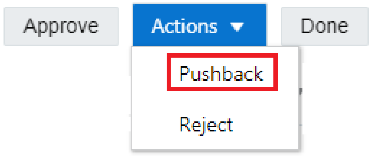
Pushing Back a Request

If you've been invited to approve or commit a request, you can push the request back to the submitter to make changes and resubmit.

Before pushing back a request, you can add comments to the request and request items and add attachments.

> **Note:**
>
> If you push back a request, any previous approvals and current invitations for the request are cleared.

1. From an open request that you are reviewing as an approver or committer, click **Actions**, and then click **Pushback**.



2. Enter a comment (required), press Tab, and then click **OK**.

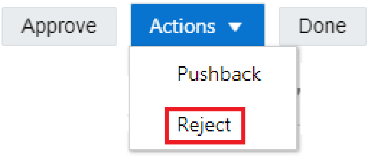
Rejecting a Request

If you've been invited to approve or commit a request, you can reject the request to close it completely with no further action available.

Before rejecting a request, you can add comments to the request and request items and add attachments.

> **Note:**
>
> If you reject a request, any previous approvals are kept, the reject is recorded in the request approval history, and current invitations for the request are cleared.

1. From an open request that you are reviewing as an approver or committer, click **Actions**, and then click **Reject**.



> **Note:**
>
> Service administrators can specify that certain types of requests cannot be rejected (see Request Settings). If the Reject option is not available, you can only Pushback or Recall the request.

2. Enter a comment (required), press Tab, and then click **Reject** to reject the request or **Pushback** to push it back to the submitter for changes.

Withdrawing an Approval for a Request

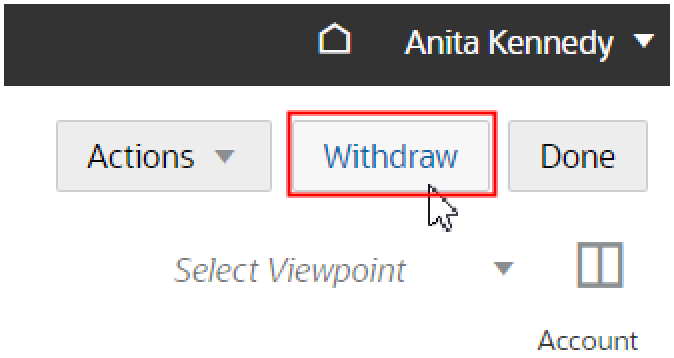
If a request that you've approved is still In Flight, you can withdraw your approval so that further edits can be made.

You can withdraw your approval from a request under these conditions:

- You contributed to the request with an Approve action.
- The request has not yet been completed and is still In Flight.
- You are not the request submitter (the request submitter must recall the request instead). See Recalling a Request.

To withdraw an approval from a request:

1. From Request Activity, open an In Flight request that you have previously approved.
2. Click **Withdraw**.



3. Enter a comment, and then click **OK**.

After you withdraw your approval, your approve action is removed from all approval policies associated with the request, and all commits are cleared. For serial policies, all subsequent approvals after your approval are also removed. See Understanding Withdrawing Request Approvals.

Commit Process

Committers are invited by email notification to commit approved requests.

The users who are invited to commit requests are based on the configuration of the commit policy. See Configuring Policies.

The commit process includes:

- Committing an approved request
- Rejecting an approved request
- Pushing an approved request back to submitter so that it can be updated and resubmitted

Committing a Request

You'll receive an email notification if you've been invited to commit a request.

When you open a request to commit, you can:

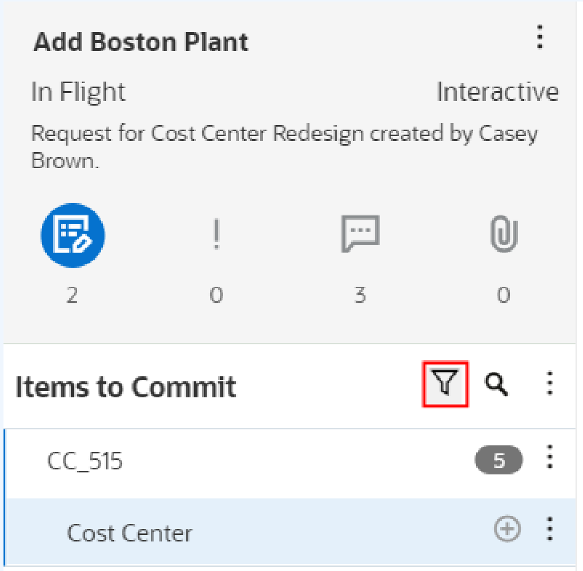
- Review the request items on the Items tab and visualize the changes in the viewpoint.
- View and add comments for specific request items on the Items tab.
- View and add comments for the request on the Comments tab.
- View and add attachments for the request on the Attachments tab.

No changes can be made to the request items during the commit phase.

1. From the email notification, click the link to open request activity and then open the request.

Note:

If you open a request where you are only committing a portion of the changes, the **Items to Commit** filter is enabled by default and displays only the request items that you're invited to commit. You can show all request items by clicking the filter icon.



Add Boston Plant

In Flight — Interactive

Request for Cost Center Redesign created by Casey Brown.

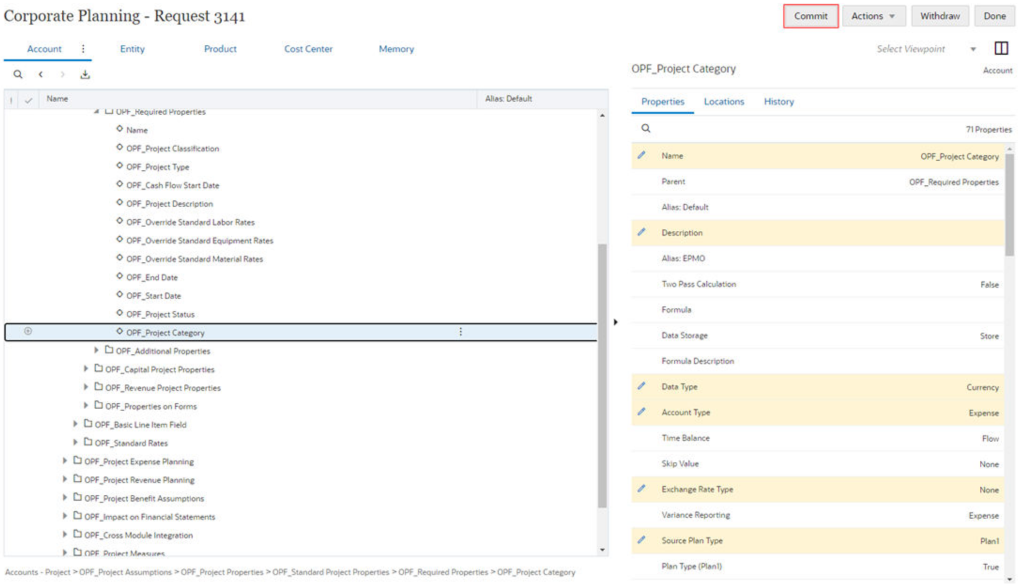
Items to Commit

CC_515 — 5

Cost Center

2. Review the changes in the request. View and add comments or attachments, and then perform an action:

   - Click **Commit**.



Corporate Planning - Request 3141

Name — Alias: Default

- OPF_Required Properties
  - Name
  - OPF_Project Classification
  - OPF_Project Type
  - OPF_Cash Flow Start Date
  - OPF_Project Description
  - OPF_Override Standard Labor Rates
  - OPF_Override Standard Equipment Rates
  - OPF_Override Standard Material Rates
  - OPF_End Date
  - OPF_Start Date
  - OPF_Project Status
  - OPF_Project Category
- OPF_Additional Properties
- OPF_Capital Project Properties
- OPF_Revenue Project Properties
- OPF_Properties on Forms
- OPF_Basic Line Item Field
- OPF_Standard Rates
- OPF_Project Expense Planning
- OPF_Project Revenue Planning
- OPF_Project Benefit Assumptions
- OPF_Impact on Financial Statements
- OPF_Cross Module Integration
- OPF_Project Measures

Accounts - Project > OPF_Project Assumptions > OPF_Project Properties > OPF_Standard Project Properties > OPF_Required Properties > OPF_Project Category

OPF_Project Category — Account

71 Properties

Name	OPF_Project Category
Parent	OPF_Required Properties
Alias: Default	
Description	
Alias: EPMO	
Two Pass Calculation	False
Formula	
Data Storage	Store
Formula Description	
Data Type	Currency
Account Type	Expense
Time Balance	Flow
Skip Value	None
Exchange Rate Type	None
Variance Reporting	Expense
Source Plan Type	Plan1
Plan Type (Plan1)	True

You can add a commit comment, and then click **OK**.

> **Note:**
>
> Validation errors must be corrected before you can commit a request. If there are validation warnings only, you must confirm that you want to continue committing the request. Click **Continue** to commit the request despite the warnings, or click **Cancel** to review the request.

The request is committed and closed.

- Click **Actions** and select one of the following:
  - **Push Back** (See Pushing Back a Request).
  - **Reject** (See Rejecting a Request).
  - **Recall** (See Recalling a Request).
- Click **Withdraw** (See Withdrawing an Approval for a Request). This action is available only if you approved the request in the approve stage.

14 Request Notifications

Email notifications are sent to users for workflow actions and subscription requests that they are assigned to.

Notifications are translated to the language that you selected in the **Language** user preference. If you do not set a language preference, the email content will be in the default language for the service. Notifications also use the time zone that you set in your preferences. If you do not select a time zone, the time zone of your browser locale is used. See Setting Your Preferences.

For more information, see:

- Subscription Notifications
- Workflow Notifications

Subscription Notifications

Email notifications are sent for subscription requests to make users aware of changes received through a subscription and to inform assignees of requests that have been assigned to them for review or issue resolution.

Notifications are sent to subscription users when a change to a source viewpoint is submitted. If a subscription request is auto-submitted, the user is notified that the changes were automatically applied to their viewpoint. Otherwise, the user is notified that the request is assigned to them for review and manual submission.

> **Note:**
>
> If the subscription assignee does not have an email address, or if that assignee has been removed from the system, the system sends the email notification to all users with Application Owner permissions in the applications contained in the source viewpoint.

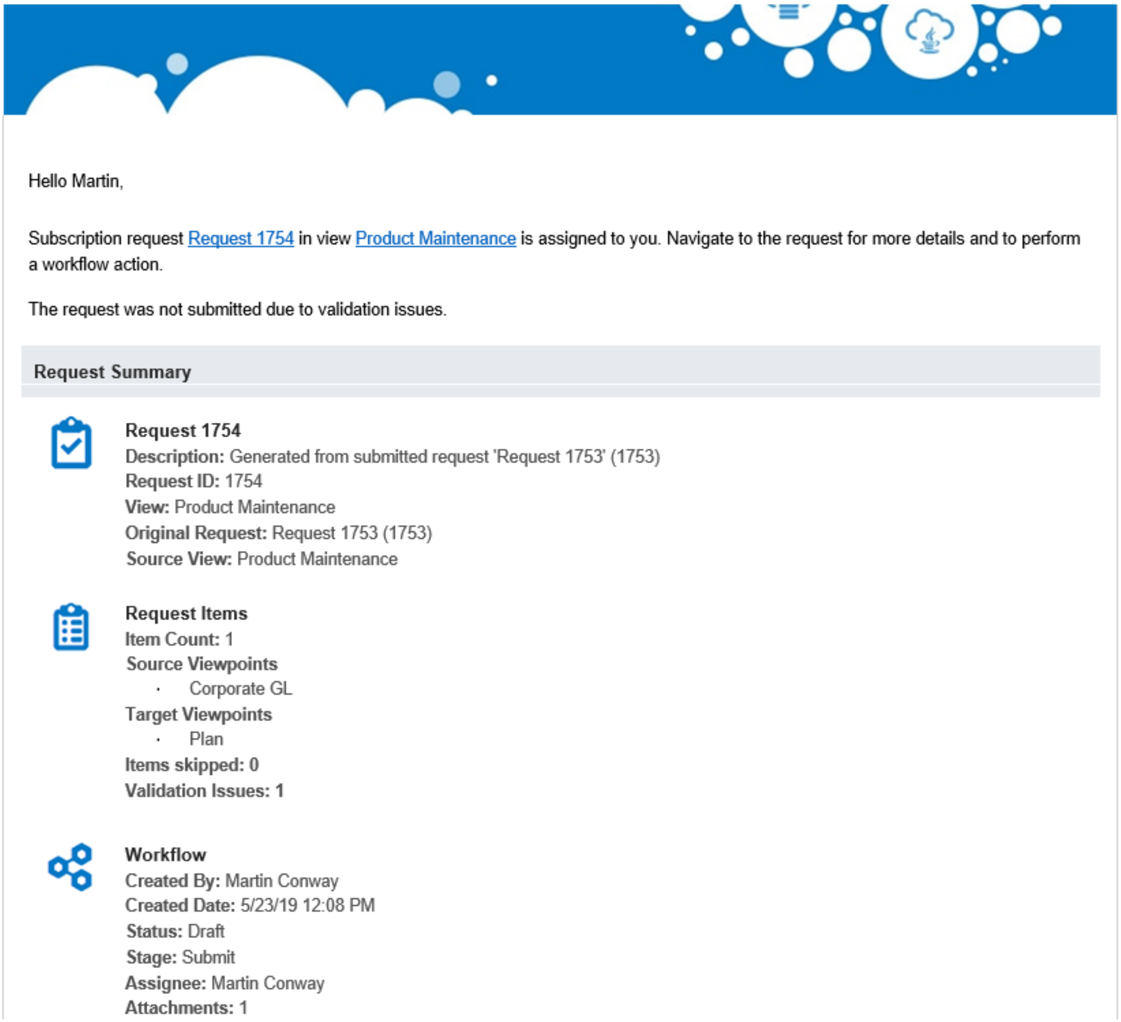
When a source viewpoint has been updated, the assignees of all of the subscribing target viewpoints receive email notifications with the request details.

> **Note:**
>
> If the **Collaborate** option is enabled for the subscription (see Creating, Editing, and Validating Subscriptions), then the alternate assignees are sent email notifications as well.

Subscription notification emails for requests that were successfully processed contain the following information:

- The request ID and title, with a link to open the request
- The source view and viewpoints
- The target view and viewpoints
- The number of request items in the subscription request and the number of items from the original request that were skipped
- An auto-submit status message that indicates whether the request was auto-submitted successfully.
- If enabled (see **Attach Subscription Request File** in Configuring System Settings), subscription notifications also contain the request file that was used to generate the request items in the subscription as an attachment. You can review the attachment to see a preview of the changes and any potential issues in the request. See Reviewing Request File Attachments.

As an example, the notification below shows that the subscription request was created, and that no records were skipped. This means that all items were successfully loaded into the subscription request, but you may still have to take action on some of the items before they can be submitted. For example, this subscription request has a validation issue that must be resolved before the request can be submitted. You can click the request link in the notification to open the request for more information.



Hello Martin,

Subscription request Request 1754 in view Product Maintenance is assigned to you. Navigate to the request for more details and to perform a workflow action.

The request was not submitted due to validation issues.

Request Summary

Request 1754
Description: Generated from submitted request 'Request 1753' (1753)
Request ID: 1754
View: Product Maintenance
Original Request: Request 1753 (1753)
Source View: Product Maintenance

Request Items
Item Count: 1
Source Viewpoints
- Corporate GL

Target Viewpoints
- Plan

Items skipped: 0
Validation Issues: 1

Workflow
Created By: Martin Conway
Created Date: 5/23/19 12:08 PM
Status: Draft
Stage: Submit
Assignee: Martin Conway
Attachments: 1

Notifications for Skipped or Failed Subscriptions

For subscription requests that were skipped or have failed, an email notification is sent to users with the *Owner* or *Metadata Manager* permission on the application or dimension of the target viewpoint making them aware of the result. For skipped subscriptions, a notification is also sent to the request assignee. These issues can result from a number of scenarios, such as:

- A subscription assignee does not have *Participant (Read)* permission to a source viewpoint and *Participant (Write)* permission in a target viewpoint
- A user was deleted, or a group has no users
- There are incompatible node types between the source and target viewpoints
- A top node filter of a subscription is invalid (for example, a top node was removed or deleted)
- A node filter expression is invalid

For failed subscriptions, when the issue has been resolved the View Owner or Service Administrator can replay the subscription for any requests where it failed previously. See Viewing Request Lineage.

Notifications for Blocked Subscription Requests

Notifications are sent if a subscription request is received for an application that is currently in an application blockout period. The notification contains the date and time that the blockout period will end.

Workflow Notifications

There are several types of email notifications that are sent in the workflow process.

Notification Type	Description
Approval	Sent to users who need to approve submitted requests
Assignment	Sent to users who are assigned a request
Collaborate	Sent to users who are asked to collaborate on requests
Commit	Sent to users who need to commit approved requests
Escalation	Sent to Data Managers for requests that need their attention due to a: • Timeout: An invitee has not responded to the reminder emails • Deadlock: There are not enough approvers available to meet the approval policy requirements
Invalid Policy	Sent to Owners and Metadata Managers for the application or dimension to which the policy applies if a policy to be enforced does not have any invitees (for example, there are no policy users available or a policy group does not contain any users) for an In Flight request.
Reminder	Sent to approvers or committers who have not yet taken action on a request

Notification Type	Description
User mentions	Sent to users who are mentioned in a request or request item comment
Updates	Sent to submitters of a request when the request is: • Completed • Pushed back • Rejected

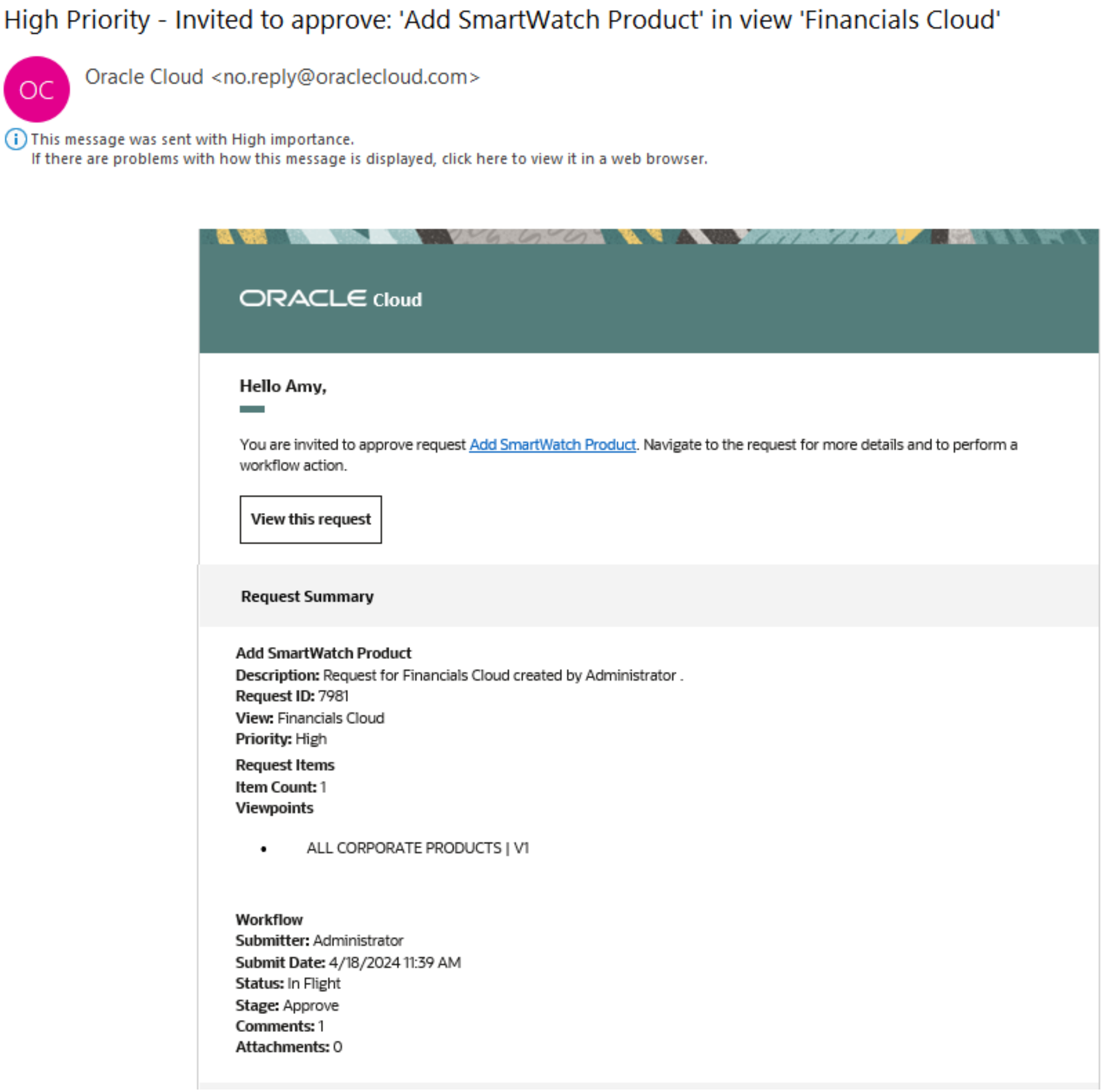
Note:

Reminder and escalation notifications are sent only if they have been configured in the approval or commit policies governing the request.

Workflow Email Example

The workflow notification below invites Amy to approve a submitted request named **Add SmartWatch Product**. The email provides a summary of the request: who created it, the view and viewpoints affected, the number of request items to approve, and the number of comments and attachments. It also indicates the request priority in the email body. If the priority is High, the email subject line is prefixed with "High Priority -" and the priority on the email object is set to High.

Amy can click the link to navigate directly to the request.



High Priority - Invited to approve: 'Add SmartWatch Product' in view 'Financials Cloud'

OC Oracle Cloud <no.reply@oraclecloud.com>

ⓘ This message was sent with High importance.
If there are problems with how this message is displayed, click here to view it in a web browser.

ORACLE Cloud

Hello Amy,

You are invited to approve request Add SmartWatch Product. Navigate to the request for more details and to perform a workflow action.

Request Summary

Add SmartWatch Product
Description: Request for Financials Cloud created by Administrator .
Request ID: 7981
View: Financials Cloud
Priority: High
Request Items
Item Count: 1
Viewpoints

- ALL CORPORATE PRODUCTS | V1

Workflow
Submitter: Administrator
Submit Date: 4/18/2024 11:39 AM
Status: In Flight
Stage: Approve
Comments: 1
Attachments: 0

15
Performing Audits

Audits enable you to view both transactional changes to data and metadata changes over time. They are accessed from the Audit screen. Use the filters on each audit screen to establish parameters for the changes that you want to view. For example, you can view all of the inserts made for a specific hierarchy set, or you can view the permissions granted to a specific group.

From the audit pages, you can drill through to view more details about the object that you are viewing. For example, you can drill through to the associated inspector for the Permissions audit, and you can drill through to the request for Transaction History.

For more information about audits, see:

- Auditing Transaction History
- Auditing System Events
- Auditing Permissions
- Auditing Policies
- Auditing Properties

Auditing Transaction History

Transaction history enables you to view changes made to data over time by viewing the transactions involving that data. You can filter the transactions that you want to view. For example, you can view all of the transactions made in the last month for a particular node, or you can view just the Move actions in a particular hierarchy set that were made over the last 7 days. You can then download a filtered view of the transactions that you want to audit to an Excel file.

Videos

Your Goal	Watch This Video
Learn about auditing transaction history.	▶ Performing Audits in Enterprise Data Management.

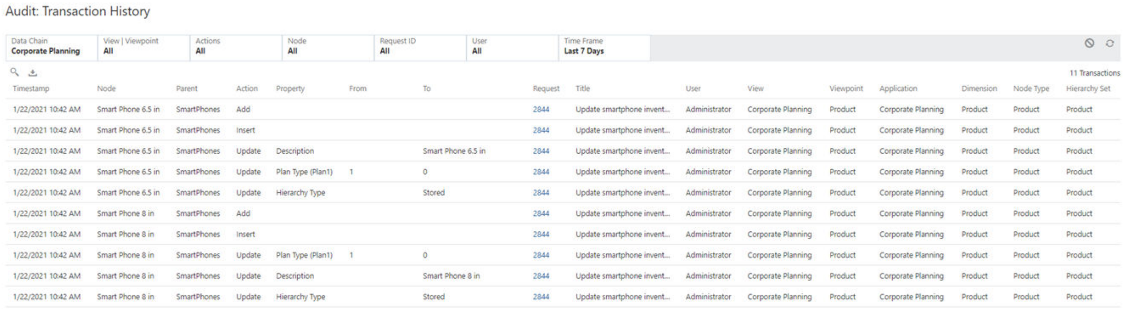
Security Considerations

The transactions that you can view depend on your application role and security permissions:

- Users with the *Service Administrator* predefined role can view changes made to data across all applications. They can also make changes to data across applications.
- Users with the *Audit* application role can view changes made to data across all applications; however, they cannot make any changes to data themselves.
- Users with at least *Data Manager* permission on a data object can view changes made to that data object only. For example, if you have *Data Manager* permission to the Account dimension within an application, you can view only the changes made in the Account dimension. You are not able to view data in the Entity or Product dimensions.

Viewing Transaction History

To access transaction history, click the **Audit** card on the home page, and then select **Transactions** in the left panel. The transactions over the last seven days are displayed by default.



Audit: Transaction History

Data Chain: Corporate Planning
View | Viewpoint: All
Actions: All
Node: All
Request ID: All
User: All
Time Frame: Last 7 Days

11 Transactions

Timestamp	Node	Parent	Action	Property	From	To	Request	Title	User	View	Viewpoint	Application	Dimension	Node Type	Hierarchy Set
1/22/2021 10:42 AM	Smart Phone 6.5 in	SmartPhones	Add				2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 6.5 in	SmartPhones	Insert				2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 6.5 in	SmartPhones	Update	Description		Smart Phone 6.5 in	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 6.5 in	SmartPhones	Update	Plan Type (Plan1)	1	0	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 6.5 in	SmartPhones	Update	Hierarchy Type		Stored	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 8 in	SmartPhones	Add				2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 8 in	SmartPhones	Insert				2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 8 in	SmartPhones	Update	Plan Type (Plan1)	1	0	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 8 in	SmartPhones	Update	Description		Smart Phone 8 in	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product
1/22/2021 10:42 AM	Smart Phone 8 in	SmartPhones	Update	Hierarchy Type		Stored	2844	Update smartphone invent...	Administrator	Corporate Planning	Product	Corporate Planning	Product	Product	Product

From Audit: Transaction History, you can filter the transactions that you want to view. The following filters are available:

- **Data Chain:** Select an application, dimension, node type, or hierarchy set to view the transactions on that object. Users with the *Audit* role can select any data chain object. Otherwise, you must have at least *Data Manager* permission on an object in order to select it.
- **View or Viewpoint:** Select a view or viewpoint to view the transactions from that view or viewpoint. If a node is in more than one viewpoint, only the transactions that took place in the selected viewpoint are displayed.

  > **Tip:**
  >
  > Because you may not know which viewpoint a change to a node was made in, when searching for transactions on a node it can be helpful to filter on the application or dimension rather than the viewpoint.

- **Actions:** Select the specific action that you want to display. The following actions are available to select. You can select more than one action to filter on.
  - Add
  - Insert
  - Move
  - Remove
  - Delete
  - Update
  - Rename
  - Reorder

> **Note:**
>
> For data import actions, a single transaction is displayed with "Import" as the action. The full details of what data was imported and where are not captured in Audit.

- **Node:** Enter an individual node to see the transactions on that node. Only exact matches are displayed, and wildcard characters are not supported.
- **Request ID:** Enter a specific request ID to see the transactions in that request. Only exact matches are displayed, and wildcard characters are not supported.
  Optional: Use the **Include Related Requests** option to include the transactions for all requests with the same lineage as the request ID that you entered (see Viewing Request Lineage) in the filtered results as well.

  > **Tip:**
  >
  > For subscription requests, the **Source Request** column displays the originating request that generated the subscription requests. You may have to scroll to see this column.

- **User:** Select a user name to see the transactions made by that user.
- **Time Frame:** From Period, select the time frame to see transactions made in that time period, or click Date Range and enter a custom date range. The following date ranges are available to select:
  - Last 7 Days
  - Last 30 Days
  - Last 60 Days
  - Last 90 Days
  - Last Year

When using the transaction history filters, click **Reset All Filters** ⊘ to reset all filters to their default values, and click **Refresh Data** ⟳ to refresh the data returned by the current filter settings.

Click **Search** 🔍 to search the transactions that are returned by your filter selection. The search is performed dynamically on all fields as you type. For example, searching for "**ro**" will return transactions with "Casey B**ro**wn" in the User column, "P**ro**duct" in the Viewpoint column, and "Corporate P**ro**ducts" in the Dimension column.

> **Note:**
>
> When you delete an application, all of the transaction history on the data objects in that application is deleted as well.

Saving Transaction History Filters

You can save your filter selections so that you can quickly view specific transactions on a repeated basis. Saved filters are only available to the user who saved them.

To save your filters:

1. In the filter header, click the **Saved Filters** menu ⋮, and then click **Save**.
2. Enter a name and click **Save**.

After you save your filters, the name of the saved filters is displayed in the left panel and in the Audit header. Click the arrow next to **Transactions** in the left panel to expand the list of saved filters.

From a saved filter, use the **Saved Filters** menu ⋮ to perform one of the following tasks:

- **Save**: Saves the filter
- **Save As**: Saves the filter with a different name.
- **Add Favorite**: Adds the saved filter to your Favorites list. Saved filters in your Favorites list are indicated by a ★ icon. You can remove a saved filter from your Favorites list by selecting **Undo Favorite**. See Working with Favorites.
- **Delete**: Deletes the saved filter.

To clear the selection of a saved filter, click **Reset All Filters** ⊘.

Inspecting Requests from Transaction History

In the **Request ID** column, you can click on a request ID to open that request in the request inspector. This enables you to see more information about each request, such as the approval workflow history.

Downloading Transaction History

After you have filtered the transaction history to display the transactions that you want to audit, you can download the results to an Excel file by clicking **Download** ⤓ and then selecting an option:

- **Transaction History**: downloads the transactions only.
- **Transaction and Workflow History**: downloads the transactions as well as the workflow history (such as submits, approvals, and commits) for all of the transactions on a separate tab.

> **Note:**
>
> The on-screen display returns up to 10,000 records, but the full result set gets downloaded to the Excel file. For example, if your filter returns 12,000 results, the first 10,000 are visible on-screen, but the full 12,000 will be downloaded to the Excel file when you click **Download** ⤓.

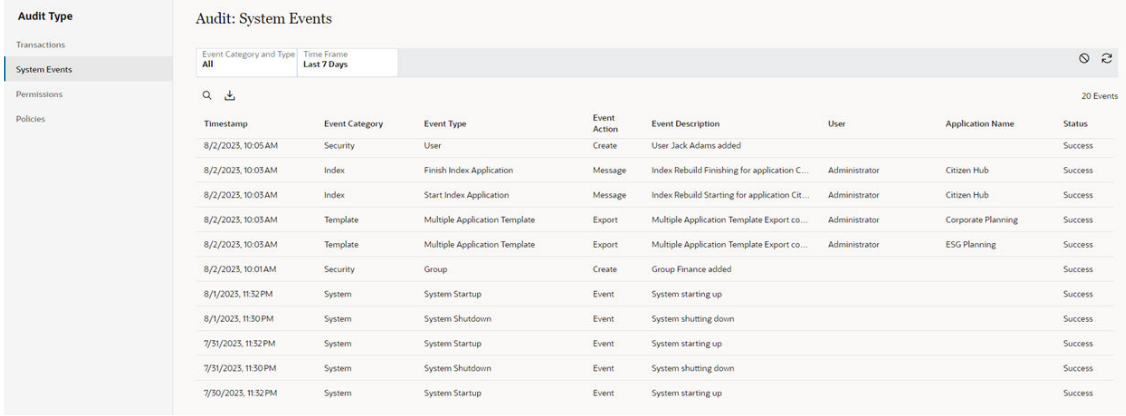
Click the linked Request ID in the Excel file to navigate to that request in Oracle Fusion Cloud Enterprise Data Management.

Auditing System Events

Auditing system events enables Service Administrators to see general events that occurred in the system (such as migration tasks and user changes), including when the events occurred and the status of each event.

You can filter the event audit by event type, and you can download the results to an Excel file.

To view system events, click the **Audit** card on the home page, and then select **System Events** in the Audit Type panel. The events in the last seven days are displayed by default.



Event Category and Type: All
Time Frame: Last 7 Days

20 Events

Timestamp	Event Category	Event Type	Event Action	Event Description	User	Application Name	Status
8/2/2023, 10:05 AM	Security	User	Create	User Jack Adams added			Success
8/2/2023, 10:03 AM	Index	Finish Index Application	Message	Index Rebuild Finishing for application C...	Administrator	Citizen Hub	Success
8/2/2023, 10:03 AM	Index	Start Index Application	Message	Index Rebuild Starting for application Cit...	Administrator	Citizen Hub	Success
8/2/2023, 10:03 AM	Template	Multiple Application Template	Export	Multiple Application Template Export co...	Administrator	Corporate Planning	Success
8/2/2023, 10:03 AM	Template	Multiple Application Template	Export	Multiple Application Template Export co...	Administrator	ESG Planning	Success
8/2/2023, 10:01 AM	Security	Group	Create	Group Finance added			Success
8/1/2023, 11:32 PM	System	System Startup	Event	System starting up			Success
8/1/2023, 11:30 PM	System	System Shutdown	Event	System shutting down			Success
7/31/2023, 11:32 PM	System	System Startup	Event	System starting up			Success
7/31/2023, 11:30 PM	System	System Shutdown	Event	System shutting down			Success
7/30/2023, 11:32 PM	System	System Startup	Event	System starting up			Success

From Audit: System Events, you can filter the events that you want to view. The following filters are available:

- **Event Category and Type**: Select the event category, and then select the type of events in that category that you want to view. The following categories types are available:

Category	Types	Description
All Excluding Index	All	Displays all events except index rebuilds
All	All	Displays all events including index rebuilds
Index	– Index Rebuild – Start Index Upgrade – Finish Index Upgrade – Start Index Application – Finish Index Application – Index Migration Import	Displays index events

Category	Types	Description
Metadata	– Application – Binding – Constraint – Compare Profile – Dimension – Extract – Extract Package – Global Connection – Hierarchy Set – Lookup Set – Node Set – Node Type – Permission – Policy – Property – Subscription – Time Label – Validation – View – Viewpoint – Viewpoint Query Profile	Displays create and delete events for metadata objects. For applications only: Displays when applications are copied and when initial and modified registration changes are applied.
Migration	– Migration Import – Migration Backup	Displays migration tasks such as imports and backups **Note**: For migration tasks, the User is always the System Administrator.
Security	– All – User – Group	Displays users and groups that were added or deleted
System	– System Startup – System Shutdown	Displays system startup and shutdown events
Template	– Application Template – Dimension Template – Multiple Application Template	Displays template import and export information for applications and dimensions

- **Time Frame:** From Period, select the time frame to see events in that time period, or click Date Range and enter a custom date range. The following date ranges are available to select:
  - Last 7 Days
  - Last 30 Days
  - Last 60 Days
  - Last 90 Days
  - Last Year
  - All (available only if **Event Category and Type** is not set to All or All Excluding Index).

When using the Audit: System Events filters, click ⊘ to reset all filters to their default values, and click 🔄 to refresh the data returned by the current filter settings.

Click 🔍 to search the events that are returned by your filter selection.

Downloading System Event Audit Results

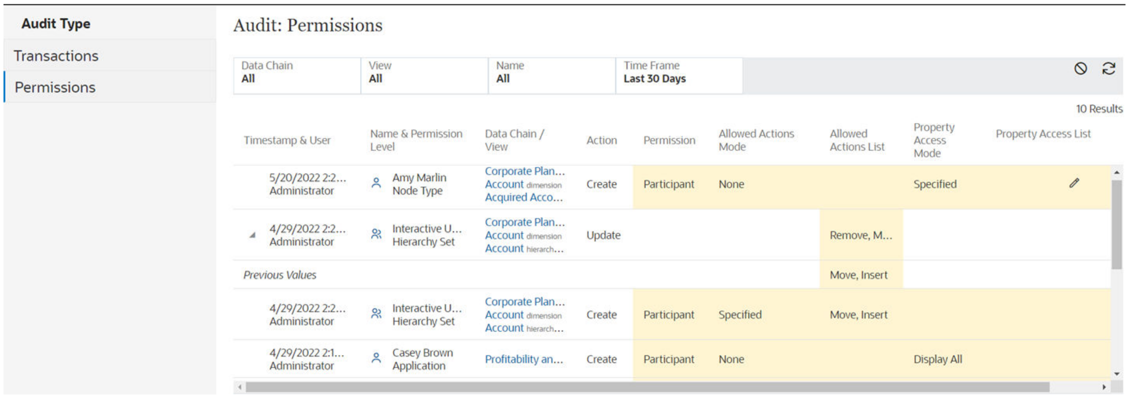
After you have adjusted the filters to display the events that you want to audit, you can download the results to an Excel file by clicking **Download** ⤓.

When you download the events, all events are included in the spreadsheet. This includes events that are not visible on the screen.

Auditing Permissions

Auditing permissions enables you to see the changes made to permissions over time. You can filter the permissions that you want to view. For example, you can view the permission changes on a specific data chain object, or for a particular user.

To view permission changes, click the **Audit** card on the home page, and then select **Permissions** in the Audit Type panel. The changes over the last seven days are displayed by default.



Audit: Permissions

Data Chain: All
View: All
Name: All
Time Frame: Last 30 Days

10 Results

Timestamp & User	Name & Permission Level	Data Chain / View	Action	Permission	Allowed Actions Mode	Allowed Actions List	Property Access Mode	Property Access List
5/20/2022 2:2... Administrator	Amy Marlin Node Type	Corporate Plan... Account dimension Acquired Acco...	Create	Participant	None		Specified	
4/29/2022 2:2... Administrator	Interactive U... Hierarchy Set	Corporate Plan... Account dimension Account hierarch...	Update			Remove, M...		
Previous Values						Move, Insert		
4/29/2022 2:2... Administrator	Interactive U... Hierarchy Set	Corporate Plan... Account dimension Account hierarch...	Create	Participant	Specified	Move, Insert		
4/29/2022 2:1... Administrator	Casey Brown Application	Profitability an...	Create	Participant	None		Display All	

From Audit: Permissions, you can filter the permission changes that you want to view. The following filters are available:

- **Data Chain:** Select an application, dimension, node type, or hierarchy set to view the permission changes for that object.
- **View:** Select a view to see the permission changes for that view.

> **Caution:**
>
> You can select to filter on a specific Data Chain object or a specific View, but not both. If you make a selection other than All on one of those filters, the other filter should be set to All.

- **Name:** Select a user or group name to see the permission changes that were made on that user or group.
- **Time Frame:** From Period, select the time frame to see permission changes made in that time period, or click Date Range and enter a custom date range. The following date ranges are available to select:
  - Last 7 Days

- Last 30 Days
- Last 60 Days
- Last 90 Days
- Last Year
- All (available only if at least one of the Data Chain, View or User filters are set to something other than All)

When using the Audit: Permissions filters, click ⊘ to reset all filters to their default values, and click ⟳ to refresh the data returned by the current filter settings.

For permissions that were changed, click **Expand** (▸) to display the previous values.

Changes to the Property Access List are not listed individually, but are indicated by a ✎ icon. You can view the details of these changes by downloading the permission audit results.

Downloading Permission Audit Results

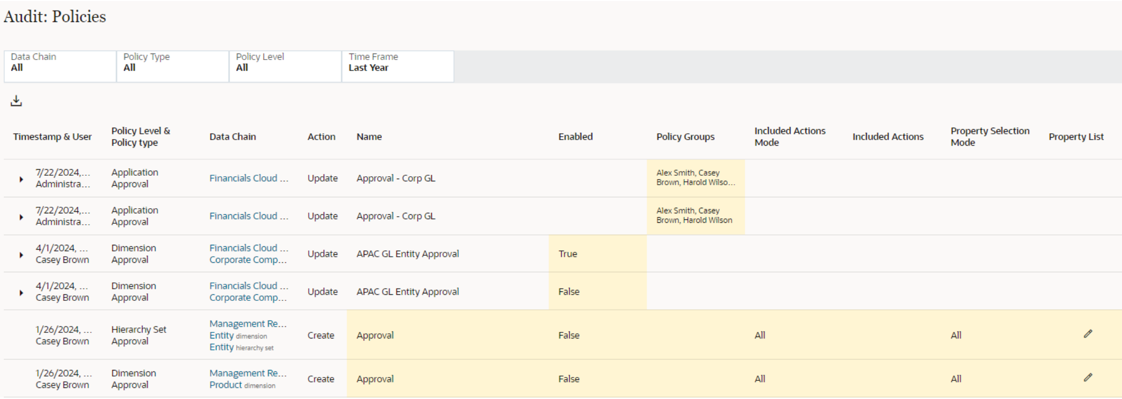
After you have adjusted the filters to display the permissions that you want to audit, you can download the results to an Excel file by clicking **Download** ⤓.

When you download the permission changes, all changes are included in the spreadsheet. This includes changes that are not visible on the screen, such as changes to the Property Access List.

Auditing Policies

Auditing policies enables you to see the changes made to policies over time. You can filter the policies that you want to view. For example, you can view the policy changes on a specific data chain object, or for a specific policy type.

To view policy changes, click the **Audit** card on the home page, and then select **Policies** in the Audit Type panel. The changes over the last seven days are displayed by default.



Audit: Policies

Data Chain	Policy Type	Policy Level	Time Frame
All	All	All	Last Year

⤓

Timestamp & User	Policy Level & Policy type	Data Chain	Action	Name	Enabled	Policy Groups	Included Actions Mode	Included Actions	Property Selection Mode	Property List
▸ 7/22/2024,... Administra...	Application Approval	Financials Cloud ...	Update	Approval - Corp GL		Alex Smith, Casey Brown, Harold Wilso...				
▸ 7/22/2024,... Administra...	Application Approval	Financials Cloud ...	Update	Approval - Corp GL		Alex Smith, Casey Brown, Harold Wilson				
▸ 4/1/2024, ... Casey Brown	Dimension Approval	Financials Cloud ... Corporate Comp...	Update	APAC GL Entity Approval	True					
▸ 4/1/2024, ... Casey Brown	Dimension Approval	Financials Cloud ... Corporate Comp...	Update	APAC GL Entity Approval	False					
1/26/2024, ... Casey Brown	Hierarchy Set Approval	Management Re... Entity dimension Entity hierarchy set	Create	Approval	False		All		All	✎
1/26/2024, ... Casey Brown	Dimension Approval	Management Re... Product dimension	Create	Approval	False		All		All	✎

From Audit: Polices, you can filter the policy changes that you want to view. The following filters are available:

- **Data Chain:** Select an application, dimension, node type, or hierarchy set to view the policy changes for that object.

> **Note:**
>
> Filtering policy changes on a data chain object will also display the policy changes for data chain objects contained by the object that you selected. For example, filtering on an application will display any policy changes for the dimensions, hierarchy sets, and node types within that application. Use the **Policy Level** filter to further refine your results.

- **Policy Type:** Select a policy type (Approval, Commit, or Notify) to view policy changes for that type only.
- **Policy Level:** Select a level to view all policy changes that were made for that level. For example, you can view all policy changes made at the hierarchy set level.
- **Time Frame:** From Period, select the time frame to see permission changes made in that time period, or click Date Range and enter a custom date range. The following date ranges are available to select:
  - Last 7 Days
  - Last 30 Days
  - Last 60 Days
  - Last 90 Days
  - Last Year
  - All (available only if at least one of the Data Chain, Policy Type or Policy Level filters are set to something other than All)

When using the Audit: Polices filters, click ⊘ to reset all filters to their default values, and click ⟳ to refresh the data returned by the current filter settings.

For policies that were changed, click **Expand** (▶) to display the previous values. Changes to Property List and Node Condition Filter are not listed individually, but are indicated by a ✎ icon. You can view the details of these changes by downloading the policy changes.

For Policy Groups, an ellipsis (…) is displayed if there are more group member changes than can be displayed on the screen. Click the group to display the complete list of policy group changes.

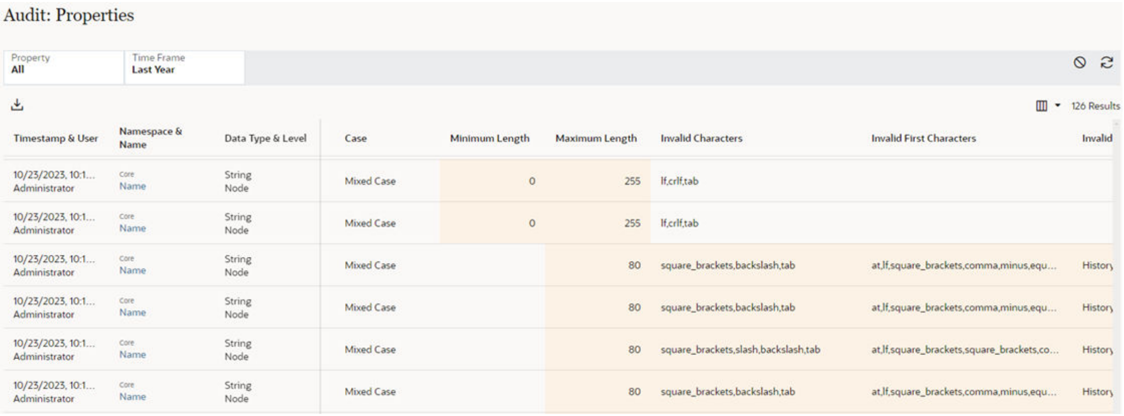
After you have filtered the policies to display the changes that you want to audit, you can download the results to an Excel file by clicking **Download** ⤓.

When you download the policy changes, all changes are included in the spreadsheet. This includes changes that are not visible on the screen, such as the Policy Groups.

Auditing Properties

Auditing properties enables you to see the changes made to properties over time. You can filter the properties that you want to view. For example, you can view the changes made to a specific property, or changes made over a specific time period.

To view property changes, click the **Audit** card on the home page, and then select **Properties** in the Audit Type panel. The changes over the last seven days are displayed by default.



Audit: Properties

Property: All
Time Frame: Last Year

126 Results

Timestamp & User	Namespace & Name	Data Type & Level	Case	Minimum Length	Maximum Length	Invalid Characters	Invalid First Characters	Invalid
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case	0	255	lf,crlf,tab		
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case	0	255	lf,crlf,tab		
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case		80	square_brackets,backslash,tab	at,lf,square_brackets,comma,minus,equ...	History
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case		80	square_brackets,backslash,tab	at,lf,square_brackets,comma,minus,equ...	History
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case		80	square_brackets,slash,backslash,tab	at,lf,square_brackets,square_brackets,co...	History
10/23/2023, 10:1... Administrator	Core Name	String Node	Mixed Case		80	square_brackets,backslash,tab	at,lf,square_brackets,comma,minus,equ...	History

From Audit: Properties, you can filter the property changes that you want to view. The following filters are available:

- **Property**: Select the property that you want to view changes for, or select All to view changes to all properties (not available if Time Frame is set to All).
- **Time Frame:** From Period, select the time frame to see property changes made in that time period, or click Date Range and enter a custom date range. The following date ranges are available to select:
  - Last 7 Days
  - Last 30 Days
  - Last 60 Days
  - Last 90 Days
  - Last Year
  - All (available only if at Property is set to something other than All)

When using the Audit: Properties filters, click ⃠ to reset all filters to their default values, and click ⟳ to refresh the data returned by the current filter settings.

Selecting Columns

On the Audit screen, there are two types of columns. Common columns are always displayed. These include:

- **Timestamp and User**
- **Action**
- **Namespace and Name**
- **Data Type and Level**
- **Definition Section**
- **Data Chain**
- **Description**
- **Default Type**
- **Default Value**
- **Editable**

- **Lock on Commit**
- **Inheritance**

The other columns on the screen are data-specific, and the columns that are displayed depend on the column set that you select. For example, for the string column set the Minimum and Maximum Length columns are displayed, and for a node property the Assigned Node Set column is displayed.

Use the **Select Column Set** ▥ ▾ menu to change the data-specific columns that are displayed. You can select from these options:

- **None**: Do not display any data-specific columns.
- **All**: Display data-specific columns for all data types

> **Note:**
>
> When all properties are displayed, you can select multiple data-specific column groups to display.

- **String**: Display the string-specific columns (for example, Invalid Characters and Invalid Values) for the records that are being displayed. Includes string, list, and memo data types.
- **Boolean**: Display Boolean-specific columns (for example, Boolean Display Type) for the records that are being displayed.
- **Numeric**: Display numeric-specific columns (for example, Minimum and Maximum Values) for the records that are being displayed. Includes float, integer, and numeric string data types.
- **Sequence**: Display sequence-specific columns (for example, Starting and Step Values) for the records that are being displayed.
- **Node**: Display node-specific columns (for example, Assigned Node Set) for the records that are being displayed.

Changes that are too complex to list on screen are indicated by a ✎ icon. You can view the details of these changes by downloading the property changes. See Downloading Audit Results

Downloading Audit Results

After you have filtered the properties to display the changes that you want to audit, you can download the results to an Excel file by clicking **Download** ⤓.

When you download the property changes, all changes are included in the spreadsheet. This includes changes that are not visible on the screen. For example, if the complete list of Invalid Values is too long to display on the screen, the downloaded spreadsheet will contain the full list.

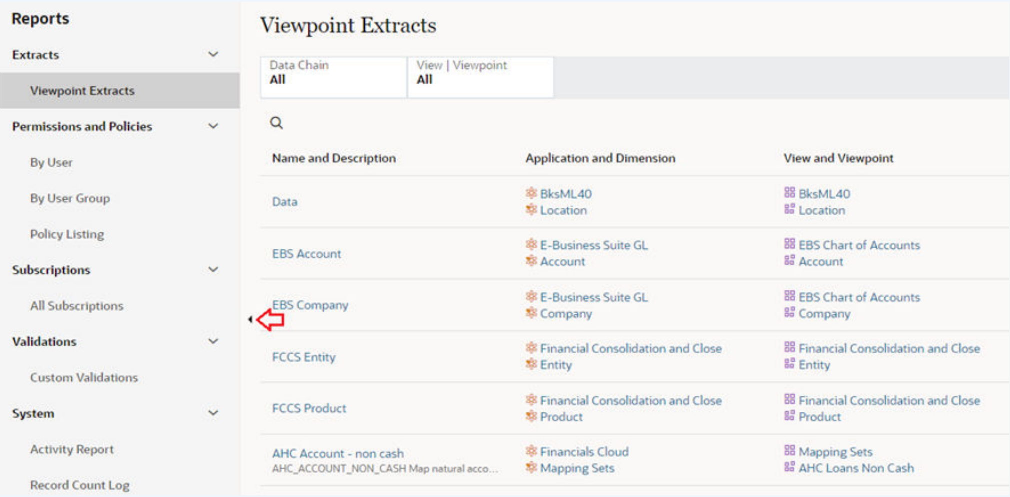
16 Working with Reports

Reports are grouped by type.

These report types are available:

- Extracts Reports
- Permissions and Policies Reports
- Subscriptions Reports
- Custom Validations Report
- System Activity Reports
- Record Count Log Report

> **Note:**
>
> For all report types, you can collapse the sidebar to provide more space on the report screen. Hover over the line between the reports sidebar and the report screen to locate the collapse and expand icon.



Name and Description	Application and Dimension	View and Viewpoint
Data	BksML40 Location	BksML40 Location
EBS Account	E-Business Suite GL Account	EBS Chart of Accounts Account
EBS Company	E-Business Suite GL Company	EBS Chart of Accounts Company
FCCS Entity	Financial Consolidation and Close Entity	Financial Consolidation and Close Entity
FCCS Product	Financial Consolidation and Close Product	Financial Consolidation and Close Product
AHC Account - non cash AHC_ACCOUNT_NON_CASH Map natural acco...	Financials Cloud Mapping Sets	Mapping Sets AHC Loans Non Cash

Extracts Reports

Extracts reports provide a view of all extracts or extract packages that you have access to. From the reports, you can inspect an extract or extract package, an application or dimension, or a view or viewpoint. You can also run an extract or an extract package to a file if you have the appropriate access.

The following extract reports are available:

- Viewpoint Extracts Report: Displays all extracts for which you have *Participant (Read)* permission (or better) on at least one data chain object in at least one viewpoint in the view.
- Extract Packages Report: Displays extract packages for which you have access to run all of the extracts in the package.

Navigating Extract Reports

You can search the report results by clicking **Search** 🔍 and entering text. The columns of the report are searched for the text that you enter.

Click **Refresh Data** ⟳ to rerun a report after you change a filter, and click **Reset All Filters** icon ⊘ to reset the filters on any report to the default settings.

To run an extract or an extract package to a file, click **Run Extract to File** or **Run Extract Package to File** ⭳. For extracts, enter a file name (or accept the default), and then click **Run**. You must have *Participant (Read)* permission (or better) on all of the properties in all of the node types in a viewpoint in order to run the extract.

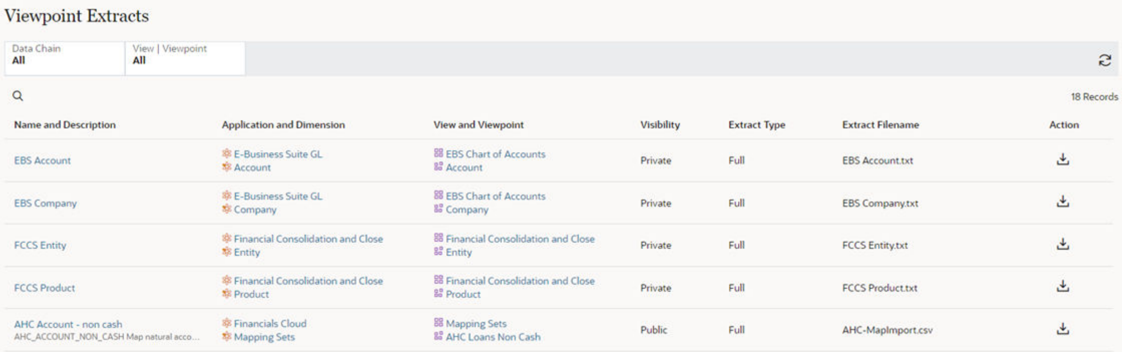
> **Note:**
>
> For incremental extracts only, you can also override the From and To dates for the current extract. This does not change the default values on the extract itself. Click **Remove Override** ⏻ to restore the original values from the extract.
> The time zone for the incremental extract is set at the extract level. Inspect the extract and navigate to the **Options** tab to change it. See Specifying Extract Options.

Viewpoint Extracts Report

The report displays all of your private extracts, as well as the public extracts for all views for which you have the *Participant (Read)* permission (or better) on at least one data chain object in at least one viewpoint in the view.

To customize your report, make selections from these filters:

- **Data Chain**: Select **All**, or choose an application and, optionally, a dimension.
- **View|Viewpoint**: By default, all viewpoints in the first view (alphabetically) that you have access to is displayed in the report. Use this filter to select a different view or viewpoint.



Viewpoint Extracts

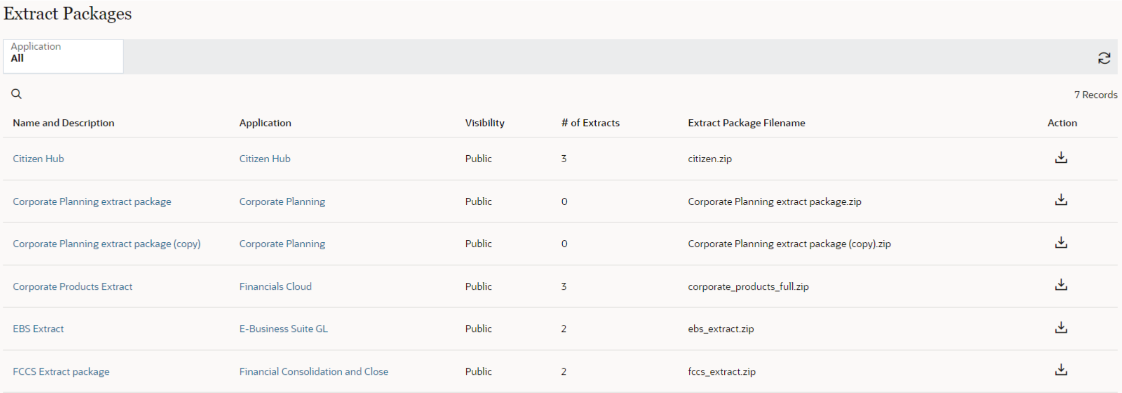
Data Chain: All
View | Viewpoint: All

18 Records

Name and Description	Application and Dimension	View and Viewpoint	Visibility	Extract Type	Extract Filename	Action
EBS Account	E-Business Suite GL Account	EBS Chart of Accounts Account	Private	Full	EBS Account.txt	⭳
EBS Company	E-Business Suite GL Company	EBS Chart of Accounts Company	Private	Full	EBS Company.txt	⭳
FCCS Entity	Financial Consolidation and Close Entity	Financial Consolidation and Close Entity	Private	Full	FCCS Entity.txt	⭳
FCCS Product	Financial Consolidation and Close Product	Financial Consolidation and Close Product	Private	Full	FCCS Product.txt	⭳
AHC Account - non cash AHC_ACCOUNT_NON_CASH Map natural acco...	Financials Cloud Mapping Sets	Mapping Sets AHC Loans Non Cash	Public	Full	AHC-MapImport.csv	⭳

Extract Packages Report

To customize your report, use the **Application** filter to select a specific application to view extract packages for.



Extract Packages

Application: All

7 Records

Name and Description	Application	Visibility	# of Extracts	Extract Package Filename	Action
Citizen Hub	Citizen Hub	Public	3	citizen.zip	⤓
Corporate Planning extract package	Corporate Planning	Public	0	Corporate Planning extract package.zip	⤓
Corporate Planning extract package (copy)	Corporate Planning	Public	0	Corporate Planning extract package (copy).zip	⤓
Corporate Products Extract	Financials Cloud	Public	3	corporate_products_full.zip	⤓
EBS Extract	E-Business Suite GL	Public	2	ebs_extract.zip	⤓
FCCS Extract package	Financial Consolidation and Close	Public	2	fccs_extract.zip	⤓

Permissions and Policies Reports

Permissions and policies reports provide a comprehensive view of permissions and policy types assigned to users and groups across all applications. These reports make it easy to view user access across applications so you can set up access or modify it. From a report, you can inspect the associated applications, dimensions, and data chain objects to view more details or make changes to the permissions and policies.

> **Note:**
>
> Permissions and policies on archived data chain objects are not included in the reports.

The following reports are available:

- By User: Displays permissions and policies for a specified user.
- By Group: Displays permissions and policies for a specified user group.
- Policy Listing: Displays a list of all policies across selected data chain objects.

Navigating Permissions and Policies Reports

These icons are used in permissions and policies reports:

- 👥 - Indicates a permission
- 📄 - Indicates a policy
- 👤 - Indicates a user

- 👥 - Indicates a user group

You can search the report results by clicking 🔍 and entering text. The columns of the report are searched for the text that you enter. You can also download the report by clicking ⤓.

Use the filters to customize a report to display specific permission and policy information. Click **Refresh** ⟳ to rerun the report after you change a filter, and click **Reset All Filters** icon ⊘ to reset the filters on any report to the default settings.

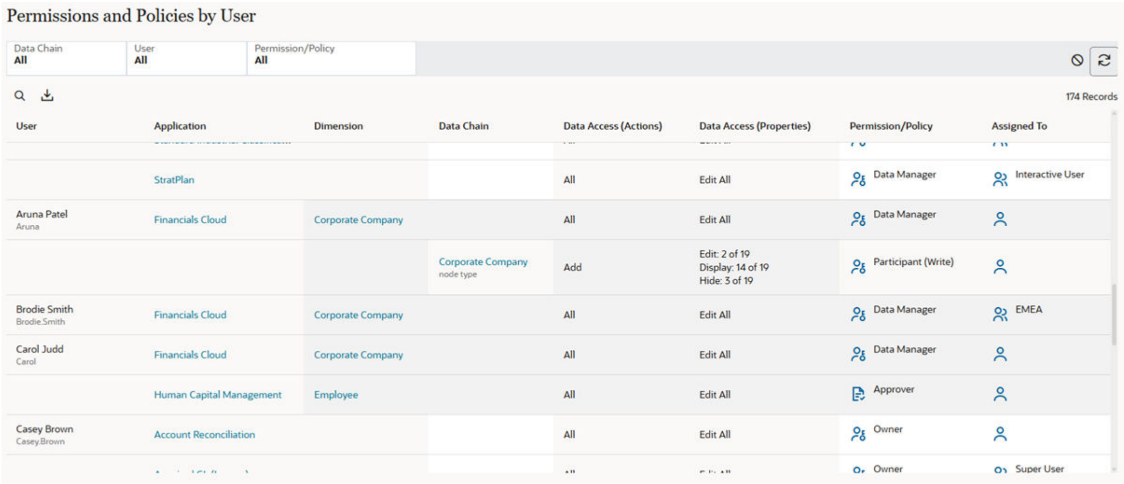
By User

To create a report by user, make selections from these filters:

- **Data Chain** - Select **All** or make selections from: application, dimension, node type, and hierarchy set.
- **User** - Select **All**, **Current User**, or a specific user. The default setting is **Current User** when the report is first opened.
- **Policy/Permission** - Select **All**, **Approver**, **Committer**, **Data Manager**, **Metadata Manager**, **Notified**, **Owner**, **Participant (Read)**, or **Participant (Write)**.

The report displays the following information:

- **User**: The user that the report is for.
- **Application/Dimension/Data Chain**: The data chain objects that a user has access to
- **Data Access (Actions)**: The allowed actions that the user has for each data chain object
- **Data Access (Properties)**: The number of properties that the user has read or edit access to (node type or hierarchy set level only)
- **Permission/Policy**: The permission 👥 or policy 📝 to that data chain object for that user
- **Assigned To**: Indicates whether the permission was assigned directly to the user 👤 or to a group 👥 that the user belongs to.



Permissions and Policies by User

Data Chain: All | User: All | Permission/Policy: All

174 Records

User	Application	Dimension	Data Chain	Data Access (Actions)	Data Access (Properties)	Permission/Policy	Assigned To
	StratPlan			All	Edit All	Data Manager	👥 Interactive User
Aruna Patel Aruna	Financials Cloud	Corporate Company		All	Edit All	Data Manager	👤
			Corporate Company node type	Add	Edit: 2 of 19 Display: 14 of 19 Hide: 3 of 19	Participant (Write)	👤
Brodie Smith Brodie.Smith	Financials Cloud	Corporate Company		All	Edit All	Data Manager	👥 EMEA
Carol Judd Carol	Financials Cloud	Corporate Company		All	Edit All	Data Manager	👤
	Human Capital Management	Employee		All	Edit All	Approver	👤
Casey Brown Casey.Brown	Account Reconciliation			All	Edit All	Owner	👤
						Owner	👥 Super User

By Group

To create a report by group, make selections from these filters:

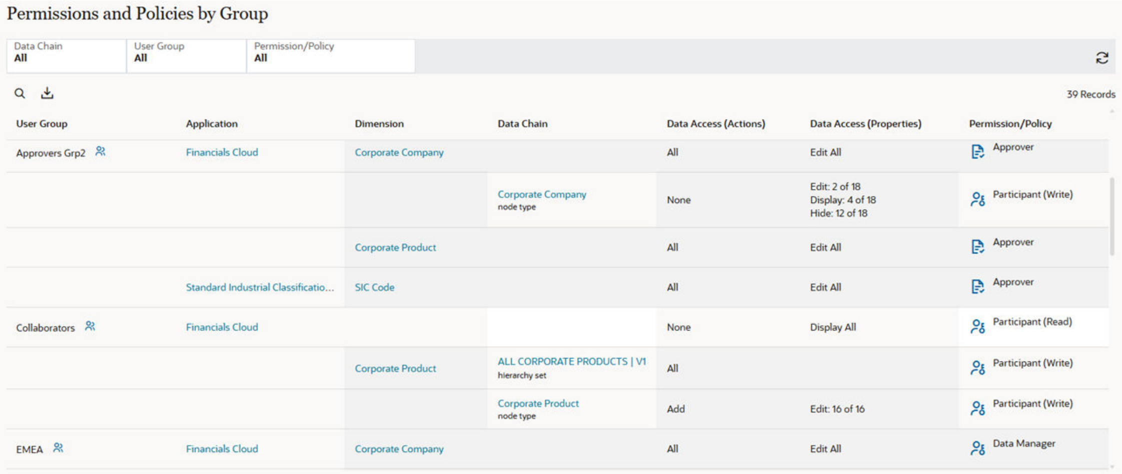
- **Data Chain**: Select **All** or make selections from: application, dimension, node type, and hierarchy set.
- **User Group**: Select **All** or a specific group.

> **Note:**
>
> Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control* .

- **Policy/Permission**: Select **All**, **Approver**, **Committer**, **Data Manager**, **Metadata Manager**, **Notified**, **Owner**, **Participant(Read)**, or **Participant(Write)**.

The report displays the following information:

- **User Group**: The group that the report is for.
- **Application/Dimension/Data Chain**: The data chain objects that a group has access to
- **Data Access (Actions)**: The allowed actions that the group has for each data chain object
- **Data Access (Properties)**: The number of properties that the group has read or edit access to (node type or hierarchy set level only)
- **Permission/Policy**: The permission or policy to that data chain object for that group.



Permissions and Policies by Group

Data Chain: All
User Group: All
Permission/Policy: All

39 Records

User Group	Application	Dimension	Data Chain	Data Access (Actions)	Data Access (Properties)	Permission/Policy
Approvers Grp2	Financials Cloud	Corporate Company		All	Edit All	Approver
			Corporate Company node type	None	Edit: 2 of 18 Display: 4 of 18 Hide: 12 of 18	Participant (Write)
		Corporate Product		All	Edit All	Approver
	Standard Industrial Classificatio...	SIC Code		All	Edit All	Approver
Collaborators	Financials Cloud			None	Display All	Participant (Read)
		Corporate Product	ALL CORPORATE PRODUCTS | V1 hierarchy set	All		Participant (Write)
			Corporate Product node type	Add	Edit: 16 of 16	Participant (Write)
EMEA	Financials Cloud	Corporate Company		All	Edit All	Data Manager

> **Tip:**
>
> In the report, click the information icon next to a group to display the users in that group.

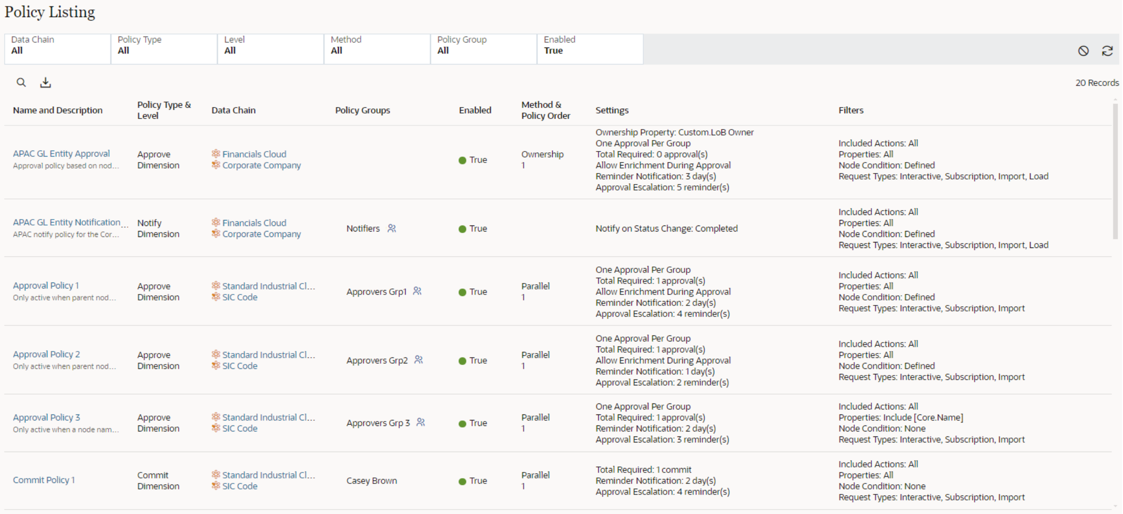
Policy Listing

To view policies, make selections from these filters:

- **Data Chain**: Select **All** or make selections from **Application**, **Dimension**, **Hierarchy Set**, or **Node Type**. Use this filter to select the level at which the policies are defined.
- **Policy Type**: Select **Approval**, **Commit**, or **Notify**.
- **Level**: Select **All**, **Application**, **Dimension**, **Hierarchy Set**, or **Node Type**. Use this filter to select the data chain object that the policy is on.
  For example, select the Corporate Planning application in the **Data Chain** filter and Hierarchy Set in the **Level** filter to view all of the policies on all of the hierarchy sets in the Corporate Planning application.
- **Method**: Select **All**, , **Parallel**, **Serial**, **Ownership**, or **Management Hierarchy**.
- **Policy Group**: Select **All** or a specific user or group.

  > **Tip:**
  >
  > In the report, click the information icon next to a group to display the users in that group.

- **Enabled**: Select **All**, **True**, or **False**.



Policy Listing

Data Chain: All
Policy Type: All
Level: All
Method: All
Policy Group: All
Enabled: True

20 Records

Name and Description	Policy Type & Level	Data Chain	Policy Groups	Enabled	Method & Policy Order	Settings	Filters
APAC GL Entity Approval Approval policy based on nod...	Approve Dimension	Financials Cloud Corporate Company		● True	Ownership 1	Ownership Property: Custom.LoB Owner One Approval Per Group Total Required: 0 approval(s) Allow Enrichment During Approval Reminder Notification: 3 day(s) Approval Escalation: 5 reminder(s)	Included Actions: All Properties: All Node Condition: Defined Request Types: Interactive, Subscription, Import, Load
APAC GL Entity Notification... APAC notify policy for the Cor...	Notify Dimension	Financials Cloud Corporate Company	Notifiers	● True		Notify on Status Change: Completed	Included Actions: All Properties: All Node Condition: Defined Request Types: Interactive, Subscription, Import, Load
Approval Policy 1 Only active when parent nod...	Approve Dimension	Standard Industrial Cl... SIC Code	Approvers Grp1	● True	Parallel 1	One Approval Per Group Total Required: 1 approval(s) Allow Enrichment During Approval Reminder Notification: 2 day(s) Approval Escalation: 4 reminder(s)	Included Actions: All Properties: All Node Condition: Defined Request Types: Interactive, Subscription, Import
Approval Policy 2 Only active when parent nod...	Approve Dimension	Standard Industrial Cl... SIC Code	Approvers Grp2	● True	Parallel 1	One Approval Per Group Total Required: 1 approval(s) Allow Enrichment During Approval Reminder Notification: 1 day(s) Approval Escalation: 2 reminder(s)	Included Actions: All Properties: All Node Condition: Defined Request Types: Interactive, Subscription, Import
Approval Policy 3 Only active when a node nam...	Approve Dimension	Standard Industrial Cl... SIC Code	Approvers Grp 3	● True	Parallel 1	One Approval Per Group Total Required: 1 approval(s) Reminder Notification: 2 day(s) Approval Escalation: 3 reminder(s)	Included Actions: All Properties: Include [Core.Name] Node Condition: None Request Types: Interactive, Subscription, Import
Commit Policy 1	Commit Dimension	Standard Industrial Cl... SIC Code	Casey Brown	● True	Parallel 1	Total Required: 1 commit Reminder Notification: 2 day(s) Approval Escalation: 4 reminder(s)	Included Actions: All Properties: All Node Condition: None Request Types: Interactive, Subscription, Import

For more information about permissions and policies see:

- Working with Permissions
- Configuring Policies

Subscriptions Reports

Subscriptions reports provide a complete listing of subscriptions that have been defined across all views and applications. Subscriptions in the report can be filtered by data chain, view, and assignee. You can inspect a subscription directly from the report, allowing you to view more details or make changes to the subscription.

To create a subscriptions report, make selections from these filters:

- **Application** - Select **All** or select an application.
- **View | Viewpoint** - Select **All** or a specific view. If you select a view, you can select **All** viewpoints or a specific viewpoint.
- **Assignee** - Select **Current User**, **All**, **Data Manager**, or a specific user. Assignee can be a user or a group and applies to the default assignee or alternate assignees.

You can search the report results by clicking 🔍 and entering text. The columns of the report are searched for the text that you enter.

You can download the report to an Excel file by clicking ⤓.

> **Note:**
>
> Subscriptions for archived source and target viewpoints are not included in subscription reports.

The subscriptions report displays:

- Name and description of the subscription
- Source viewpoint (including view and application)
- Target viewpoint (including view and application)
- Default and alternate subscription assignees

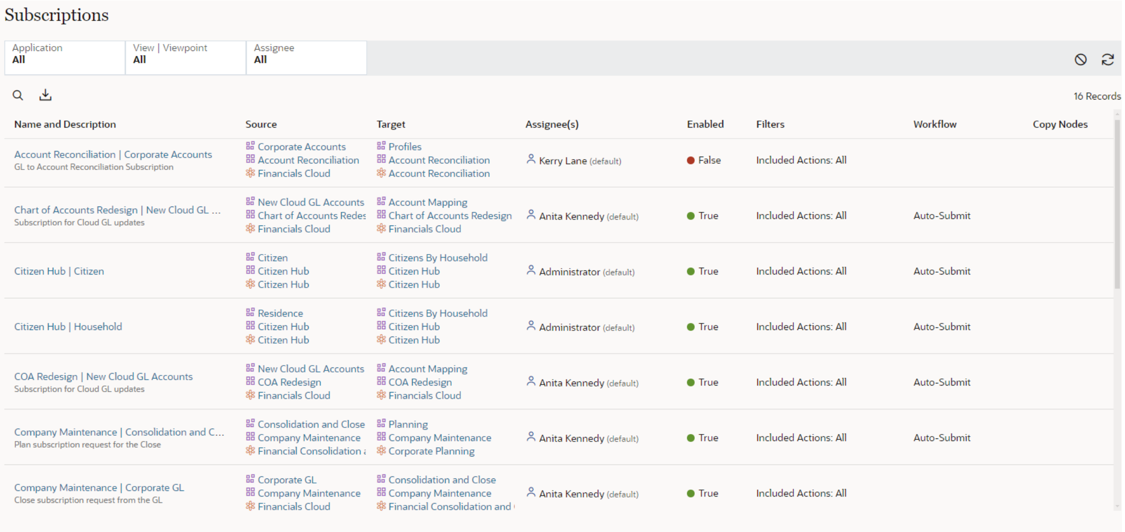
  > **Note:**
  >
  > Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control*.

- Whether or not the subscription is enabled
- The filters and conditions that are configured for the subscription, including:
  - Included Actions
  - Top Nodes
  - Node Condition
  - Inclusion Property

  See Adding Filters and Conditions to Subscriptions

- Workflow options for the subscription such as auto-submit and bypass approvals
- Copy and model after options for the subscription. See Performing Copy and Model After Operations in Subscriptions.

For example, in this report, we can see that the Account Reconciliation | Corporate Accounts subscription is disabled, while all other subscriptions are enabled.



Subscriptions

Application: All
View | Viewpoint: All
Assignee: All

16 Records

Name and Description	Source	Target	Assignee(s)	Enabled	Filters	Workflow	Copy Nodes
Account Reconciliation | Corporate Accounts GL to Account Reconciliation Subscription	Corporate Accounts Account Reconciliation Financials Cloud	Profiles Account Reconciliation Account Reconciliation	Kerry Lane (default)	● False	Included Actions: All		
Chart of Accounts Redesign | New Cloud GL ... Subscription for Cloud GL updates	New Cloud GL Accounts Chart of Accounts Redes Financials Cloud	Account Mapping Chart of Accounts Redesign Financials Cloud	Anita Kennedy (default)	● True	Included Actions: All	Auto-Submit	
Citizen Hub | Citizen	Citizen Citizen Hub Citizen Hub	Citizens By Household Citizen Hub Citizen Hub	Administrator (default)	● True	Included Actions: All	Auto-Submit	
Citizen Hub | Household	Residence Citizen Hub Citizen Hub	Citizens By Household Citizen Hub Citizen Hub	Administrator (default)	● True	Included Actions: All	Auto-Submit	
COA Redesign | New Cloud GL Accounts Subscription for Cloud GL updates	New Cloud GL Accounts COA Redesign Financials Cloud	Account Mapping COA Redesign Financials Cloud	Anita Kennedy (default)	● True	Included Actions: All	Auto-Submit	
Company Maintenance | Consolidation and C... Plan subscription request for the Close	Consolidation and Close Company Maintenance Financial Consolidation a	Planning Company Maintenance Corporate Planning	Anita Kennedy (default)	● True	Included Actions: All	Auto-Submit	
Company Maintenance | Corporate GL Close subscription request from the GL	Corporate GL Company Maintenance Financials Cloud	Consolidation and Close Company Maintenance Financial Consolidation and	Anita Kennedy (default)	● True	Included Actions: All		

For more information on subscriptions, see Subscribing to Viewpoints.

Custom Validations Report

The custom validations report provides a view of all validations that were manually created for data chain objects across all applications. From the report, you can inspect a validation, application, dimension, node type, or hierarchy set to view more details or to make changes to that object.

You can filter the validations in the report by application, dimension, node type, or hierarchy set.

You can search the report results by clicking 🔍 and entering text. The columns of the report are searched for the text that you enter. You can also download the report by clicking ⤓.

For more information about custom validations, see Custom Validations.

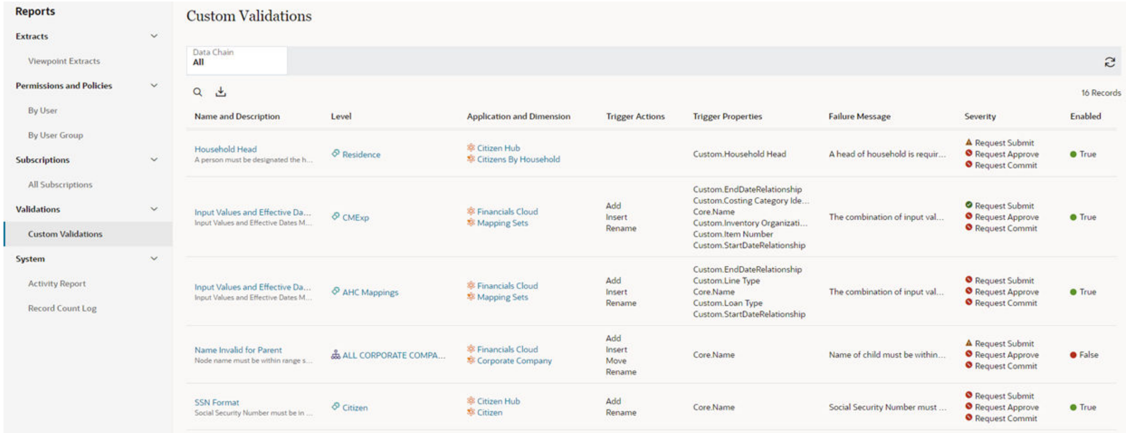
> **Note:**
>
> The custom validations report does not include system or predefined validations.

The custom validations report displays:

- Name and description of the validation
- Validation level:
  - **Application**: ❋

  - **Dimension**:
  - **Hierarchy Set**:
  - **Node Type**:
- Application and dimension that contain the validation
- The actions and properties that trigger the validation check
- The validation failure message
- The validation severity:
  - **Warning**: ⚠
  - **Error**: ⛔
  - **Ignore**: ✓
- Whether or not the validation is enabled

For example, this report lists several validations with different severity levels at both the hierarchy set and node type levels across several applications. Notice that the "Name Invalid for Parent" validation is disabled.



Custom Validations

Data Chain
All

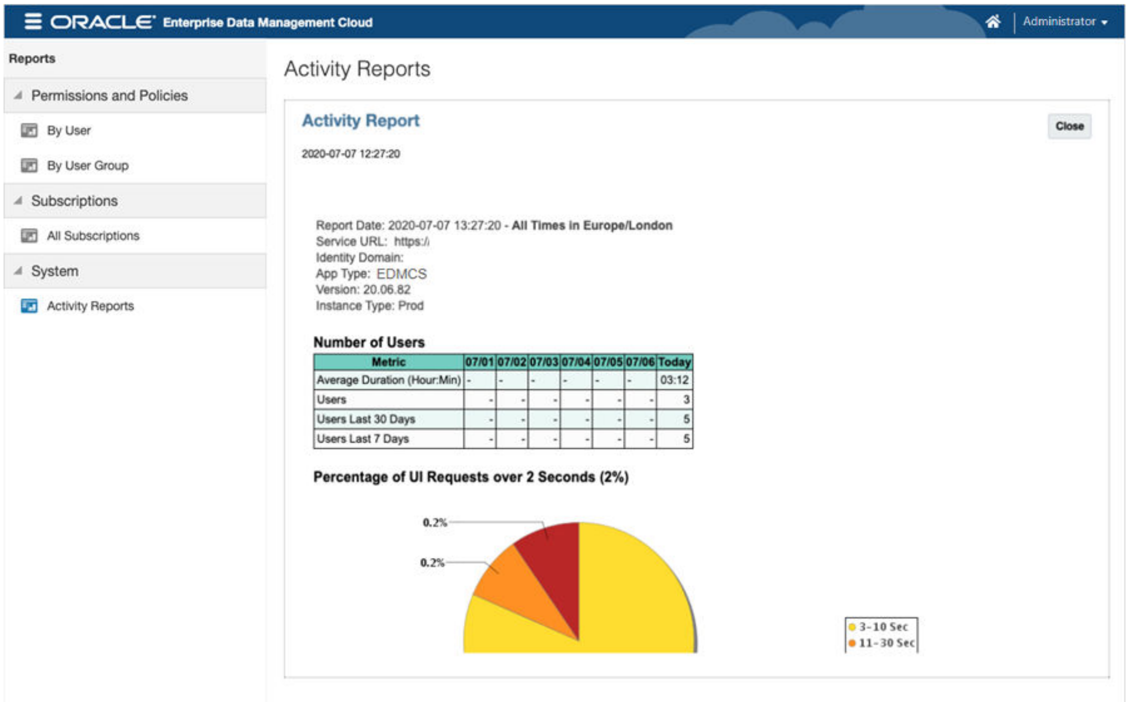
16 Records

Name and Description	Level	Application and Dimension	Trigger Actions	Trigger Properties	Failure Message	Severity	Enabled
Household Head A person must be designated the h...	Residence	Citizen Hub Citizens By Household		Custom.Household Head	A head of household is requir...	⚠ Request Submit ⛔ Request Approve ⛔ Request Commit	● True
Input Values and Effective Da... Input Values and Effective Dates M...	CMExp	Financials Cloud Mapping Sets	Add Insert Rename	Custom.EndDateRelationship Custom.Costing Category Ide... Core.Name Custom.Inventory Organizati... Custom.Item Number Custom.StartDateRelationship	The combination of input val...	✓ Request Submit ⛔ Request Approve ⛔ Request Commit	● True
Input Values and Effective Da... Input Values and Effective Dates M...	AHC Mappings	Financials Cloud Mapping Sets	Add Insert Rename	Custom.EndDateRelationship Custom.Line Type Core.Name Custom.Loan Type Custom.StartDateRelationship	The combination of input val...	⛔ Request Submit ⛔ Request Approve ⛔ Request Commit	● True
Name Invalid for Parent Node name must be within range s...	ALL CORPORATE COMPA...	Financials Cloud Corporate Company	Add Insert Move Rename	Core.Name	Name of child must be within...	⚠ Request Submit ⛔ Request Approve ⛔ Request Commit	● False
SSN Format Social Security Number must be in ...	Citizen	Citizen Hub Citizen	Add Rename	Core.Name	Social Security Number must ...	⛔ Request Submit ⛔ Request Approve ⛔ Request Commit	● True

System Activity Reports

System activity reports are automatically generated every day and enable Service Administrators to monitor system usage over time. They include user activity metrics, resource utilization, performance statistics, and troubleshooting information. System logs can also be downloaded for offline review and analysis.

You can access the system activity reports from the reports page by clicking **Activity Reports** under System. From the list of available activity reports, click **View** to view a report or **Download** to download the access logs. For more information, see About the Activity Report in *Getting Started Guide for Administrators*.



Activity Report

2020-07-07 12:27:20

Report Date: 2020-07-07 13:27:20 - **All Times in Europe/London**
Service URL: https:/
Identity Domain:
App Type: EDMCS
Version: 20.06.82
Instance Type: Prod

Number of Users

Metric	07/01	07/02	07/03	07/04	07/05	07/06	Today
Average Duration (Hour:Min)	-	-	-	-	-	-	03:12
Users	-	-	-	-	-	-	3
Users Last 30 Days	-	-	-	-	-	-	5
Users Last 7 Days	-	-	-	-	-	-	5

Percentage of UI Requests over 2 Seconds (2%)

Record Count Log Report

The record count log report enables you to identify how nodes from different applications and dimensions contribute to the total record count determined by the system.

The record count log report is available to Service Administrators only.

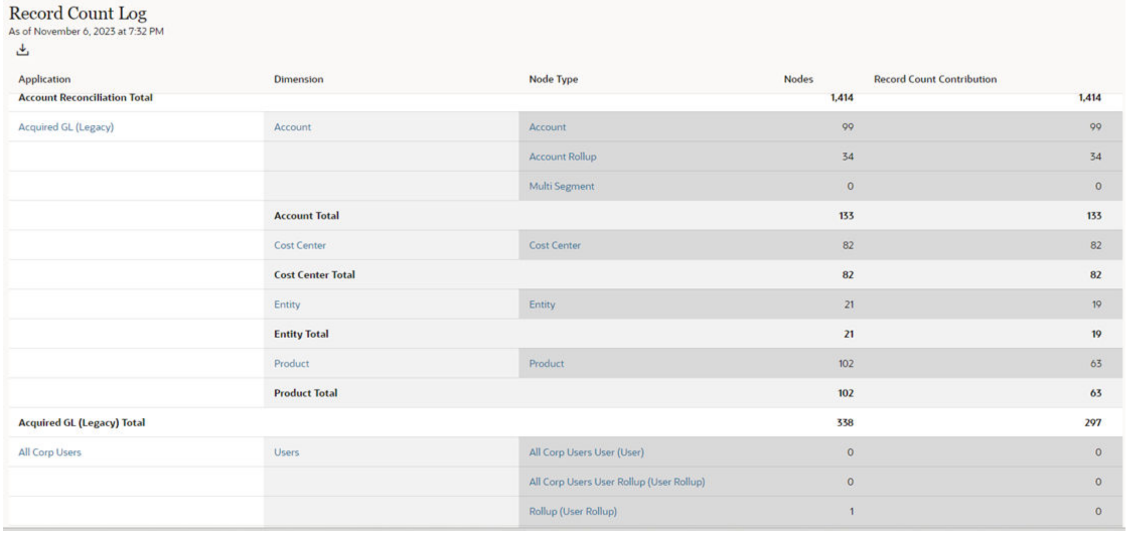
Understanding the Record Count

- When calculating node counts, two nodes are counted as equivalent if the source and target nodes have the same base name after the qualifier is removed (default or alternate qualifier, prefix or suffix). See Working with Node Type Qualifiers.
- Nodes that have a specialty node type class such as Lookup or User are identified with the corresponding labels (in parentheses) in the Record Count Log report, but they are not included in the total record count in your environment.
- Multiple instances of a node (shared nodes) are counted as one record.
- Orphan nodes (that is, nodes that are not in a hierarchy set but that do exist in a node type) do count towards the record count. Use the Orphan Nodes viewpoint query to identify these nodes. See Querying a Viewpoint.
- The counting of unique nodes for the record count is performed in order by application, dimension, and then node type. Therefore, applications that appear early in the record count log tend to have more of their nodes count towards the total record count than applications that appear further down in the log. It is likely that the applications, dimensions, and node types later in the log have had many of their nodes already counted in an earlier application, dimension, or node type.

Accessing the Record Count Log Report

Access the record count log report from the Reports page by clicking **Record Count Log** under System. The node count for each application, dimension, and node type is displayed in the Nodes column, along with the additional unique records from each of them which are included in the Record Count Contribution column.

> **Note:**
>
> The record count gets updated after the daily maintenance window or after the service is restarted. The date and time that the record count was last updated is displayed under the report title.



Record Count Log

As of November 6, 2023 at 7:32 PM

Application	Dimension	Node Type	Nodes	Record Count Contribution
Account Reconciliation Total			**1,414**	**1,414**
Acquired GL (Legacy)	Account	Account	99	99
		Account Rollup	34	34
		Multi Segment	0	0
	Account Total		**133**	**133**
	Cost Center	Cost Center	82	82
	Cost Center Total		**82**	**82**
	Entity	Entity	21	19
	Entity Total		**21**	**19**
	Product	Product	102	63
	Product Total		**102**	**63**
Acquired GL (Legacy) Total			**338**	**297**
All Corp Users	Users	All Corp Users User (User)	0	0
		All Corp Users User Rollup (User Rollup)	0	0
		Rollup (User Rollup)	1	0

Click **Download** ⤓ to download the information from the Record Count Log report to an Excel file. Node and record counts for each node type are included in the downloaded file.

17

Working with Dashboards

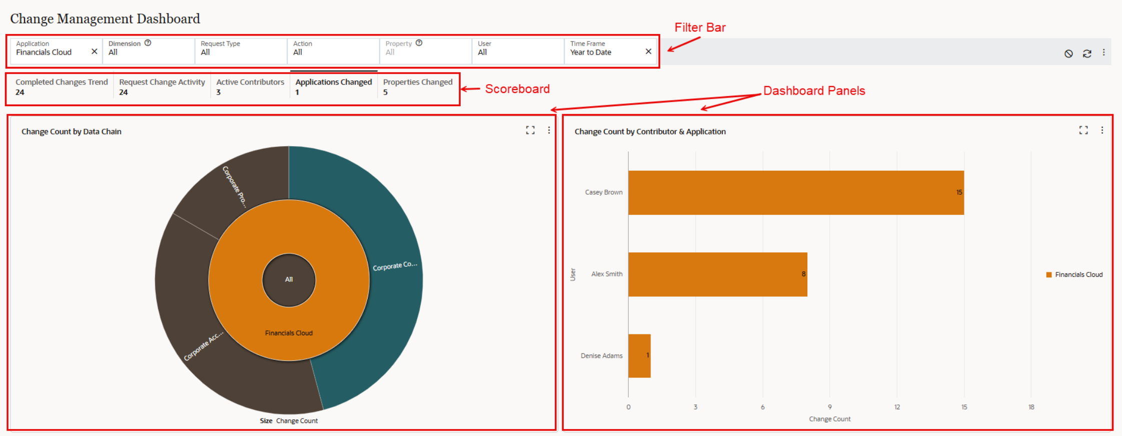
Dashboards display metrics that enable users with the appropriate permissions to analyze performance and activity using predefined key performance indicators (KPIs).

Available Dashboards

The following dashboards are available:

- Change Management Dashboard: Displays completed request changes for specific applications and dimensions, request actions, properties, and users.
- Workflow Analysis Dashboard: Displays historical workflow information for closed requests as a trend over time or by groupings such as request type and request stage.

Dashboard Overview



Change Management Dashboard

Application: Financials Cloud ✕ | Dimension: All | Request Type: All | Action: All | Property: All | User: All | Time Frame: Year to Date ✕

Completed Changes Trend 24 | Request Change Activity 24 | Active Contributors 3 | Applications Changed 1 | Properties Changed 5

Filter Bar

Scoreboard

Dashboard Panels

The dashboard page consists of the following sections:

- **Dashboard Selection Panel**: Enables you to select the dashboard that you want to view. If you have any saved filters, those are also displayed in this panel under the dashboard to which they apply.
- **Filter Bar**: Enables you to customize the dashboard to display data and metrics based on specified criteria, such as the application name or request type. See Working with the Filter Bar.
- **Scoreboard**: Displays the categories of metrics available for the current dashboard, such as Properties Changed or Active Contributors. See Working with the Scoreboard
- **Dashboard Panels**: Displays the visualized data for the metrics that you have selected. See Working with Dashboard Panels.

Working with the Filter Bar

Use the filter bar to refine the data being displayed in the dashboard panels to show the information that is relevant to your interests. For example, use the Application filter to display metrics for a specific application, or use the Time Frame filter to display information from a specific period of time.

> **Note:**
>
> The filter drop down menus display only the data chain objects that you have the appropriate permission to view. For example, the Application filter displays only the applications for which you have at least *Data Manager* permission or the *Audit* application role.

Use the following controls on the filter bar:

- **Refresh Data** ⟳ : Refreshes the data in the dashboard panels
- **Reset All Filters** ⊘: Resets the filters to the default settings.
- **Saved Filter Menu** ⋮: After you make changes to the default filter, enables you to save the current filter settings and add a saved filter as a Favorite. See *Saving Filters and Adding Favorites*, below.

From the filter bar, select one or more filter items, or click **Select All** ☑ to select all items in the filter. Click the checkbox next to a selected filter item to de-select it. For filters with a variable number of filter items, you can search for filter items in the filter search bar.

When you have finished selecting your filter items, click **OK** to apply them to the filter.

The filters that you select are applied to all dashboard panels on the dashboard page. For details about filters for the Change Management dashboard, see Change Management Dashboard.

Saving Filters and Adding Favorites

After you make changes to the default filter, you can save the current filter settings to make it easier to reapply those settings in the future, and you can also add saved filter settings as Favorites in order to easily navigate to your saved filter settings from the Favorites menu.

From the **Saved Filter Menu** ⋮ in the filter bar, select an option:

- **Save**: Save the current filter settings.
- **Save As**: Save the current filter settings with a different name.
- **Add Favorite**: Adds a saved filter as a Favorite. See Working with Favorites.
- **Delete**: Deletes a saved filter. Deleting a filter will also remove it from the Favorites list.

Saved filters are displayed in the dashboard selection panel. They are visible only to the user who saved them. You cannot create a public saved filter.

Working with the Scoreboard

The scoreboard displays scorecards for specific metrics for a dashboard. For example, the Change Management scoreboard displays metrics for changes to your applications over time, such as completed changes and properties changed. Select a scorecard on the scoreboard to display the metrics for that card.

Each scorecard displays the name of the metric and the metric value as filtered by the filter bar.

Working with Dashboard Panels

Dashboard panels display data visualizations for the metrics that you selected in the scoreboard, such as charts, tables, or diagrams.

> **Tip:**
>
> The colors in the data visualizations serve to visually distinguish the various elements of the data being visualized only. There is no specific meaning assigned to each color, and the colors of the elements may fluctuate as the data is drilled into and refreshed.

From a dashboard panel, click **Maximize** to expand the selected panel to display across the full dashboard page. From a maximized panel, click **Restore** to return the panel to a tiled display on the page.

Most dashboard panels have an **Action** menu ⋮ that enable you to take further action on that panel, such as refreshing the data in the visualization or drilling down to the underlying data in the visualization.

Drill Down

When working with the data visualizations in the dashboard panels, you can drill down to the underlying data. For example, in the Change Management dashboard you can drill down to view the associated transaction history records in the visualization.

- To view the data for the entire visualization, click the **Action** menu ⋮ in the upper right corner of the dashboard panel and select the option to view the underlying data. For example, in the Change Management dashboard you would select **View Transaction History**.
- To view the data for a specific chart element, right-click that element and select the drill down option. For example, for an element in a Change Management dashboard panel you would select **Drill Down to Transaction History**.

When you select a drill down option, the data that supports the option that you selected is displayed in the drill down drawer on the bottom half of the screen. The name of the dashboard panel is displayed along with the current filters for the data.

From the drill down drawer:

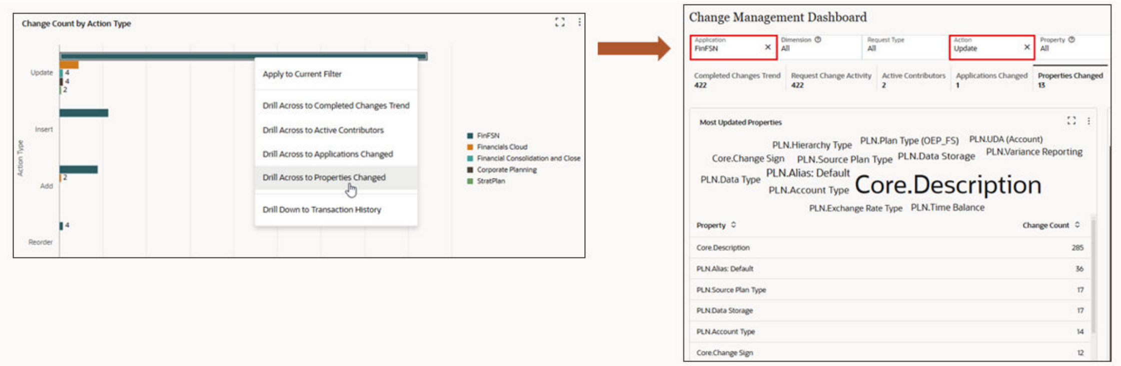
- Click **Download to File** to download the data to an Excel file. For example, for the Change Management dashboard you can download the transaction history.
- Click **Maximize** to expand the data to display across the full dashboard page. From a maximized panel, click **Restore** to restore the data to the bottom half of the page.
- Click **X** to close the drill down drawer.

Drill Across

Drilling across from a dashboard panel element lets you apply the filter in one data visualization to another visualization. To drill across, from a dashboard panel element, right-click and select a drill across option. This applies the currently visualized item to the filter bar and navigates to the appropriate scoreboard card.

Let's look at an example to understand the drill across feature. The Change Management dashboard has a visualization of the number of changes by action type. In this example, we will drill across from a specific request action type for a specific application to see which properties were changed.

To do this, we right-click the **Update** action type for the **FinFSN** application, and select **Drill Across to Properties Changed**. This action navigates to the Properties Changed card and automatically applies the FinFSN application and Update action to the page filter:



Change Management Dashboard

Application: FinFSN
Dimension: All
Request Type: All
Action: Update
Property: All

Completed Changes Trend 422
Request Change Activity 422
Active Contributors 2
Applications Changed 1
Properties Changed 13

Property	Change Count
Core.Description	285
PLN.Alias: Default	36
PLN.Source Plan Type	17
PLN.Data Storage	17
PLN.Account Type	14
Core.Change Sign	12

Apply to Current Filter

The **Apply to Current Filter** action applies the filter for the selected data element in a visualization to the page filter. This effectively enables you to zoom in on a particular data element. For example, from the Properties Changed tab, you can select a specific property and apply it to the current page filter. This enables you to see the users who changed that property.

To apply a filter for a data element to the page filter, right-click the data element and select **Apply to Current Filter**.

Change Management Dashboard

The Change Management dashboard enables you to analyze completed request changes for specific applications and dimensions, request actions, properties, and users.

Considerations

- You must have at least *Data Manager* permission on an application in order to view change management information for that application. Users with the *Auditor* application role can view change management information for all applications.
- The Change Management dashboard displays changes to data that were committed to the system by completed requests only. It does not display changes from Draft or In Flight requests.
- For dashboard panels that display metrics from contributors, the contributors data also includes assignees for subscription requests.
- The dates for the transaction data in all charts are based on Coordinated Universal Time (UTC). Aggregated changes represented in the charts may vary by a day when compared to transaction history due to time zone differences.

Change Management Scoreboard

The following scorecards are available from the Change Management Scoreboard. Each scorecard displays the name of the metric and the metric value as filtered by the filter bar.

- Completed Changes Trend
- Request Change Activity
- Active Contributors
- Applications Changed
- Properties Changed

Filter Bar

To view request changes in your system, make selections from the following filters.

- **Data Chain**: Select the applications, dimensions, hierarchy sets, and node types to view changes for. Select one or more data chain objects in the drop down list. You can select a maximum of 15 items in a single drop down list.
  - **Application**: Select one or more applications to view changes for.
  - **Dimension**: For a selected application, select one or more dimensions to view changes for.

    > **Note:**
    >
    > The Dimension filter is available only when a single application has been selected in the Application filter.

  - **Hierarchy Set**: For a selected dimension, select one or more hierarchy sets to view changes for.

    > **Note:**
    >
    > The Hierarchy set filter is available only when a single dimension has been selected in the Dimension filter.

  - **Node Type**: For a selected dimension, select one or more node types to view changes for.

    > **Note:**
    >
    > The Node Type filter is available only when a single dimension has been selected in the Dimension filter.

- **Request Type**: Select one or more request types that you want to view completed changes for. The following request types are available to select:
  - Consolidation
  - Import
  - Interactive

  - Subscription
  - Load
- **Action**: Select one or more request actions that you want to view completed changes for. The following request actions are available to select:
  - Add
  - Insert
  - Move
  - Remove
  - Delete
  - Update
  - Rename
  - Reorder
- **Property**: Select one or more properties that you want to view completed update changes for.

  > **Note:**
  >
  > The Property filter is available only when the Action filter is set to the single action **Update**.

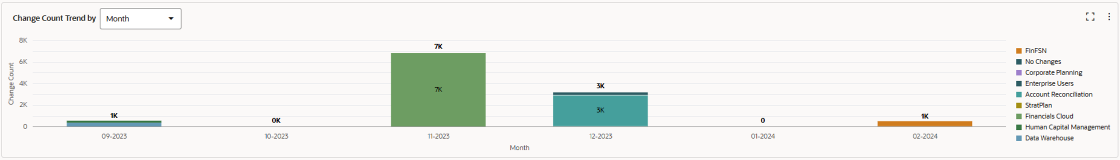
- **User**: Select one or more users to view completed changes for.
- **Time Frame**: Select the time frame to view completed changes for. You can select only a single value for the Time Frame filter.
  From **Period**, select the time period to view changes for, or click **Date Range** and enter a custom date range (month and year). The following periods are available to select:
  - Month to Date
  - Quarter to Date
  - Year to Date
  - Previous Month
  - Previous Quarter
  - Previous Year

Completed Changes Trend

The Completed Changes Trend card displays trends of completed changes over time.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels. Exceptions to these actions are described in the sections for each dashboard panel, below.

Change Count Trend

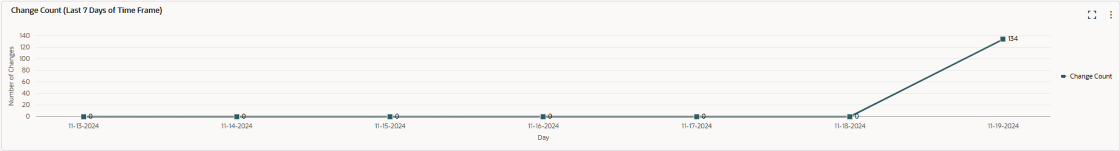


Change Count Trend by Month

The **Change Count Trend** dashboard panel displays a visualization of the number of changes over time by either month, quarter, or day, depending on your time frame filter (use the dropdown menu to select the time period for the trend).

The data elements in this chart cannot be drilled across or applied to the current filter. When viewing changes over time by Day, you can right-click a data element to drill to transaction history. When viewing changes over time by Day, Month, Quarter, or Year, you can use the **Actions** menu ⋮ to view and download the transactions in the visualization. See Working with Dashboard Panels.

Change Count (Last 7 Days of Time Frame)



The **Change Count (Last 7 Days of Time Frame)** dashboard panel displays changes over the last 7 days of the Time Frame that is selected in the filter bar.

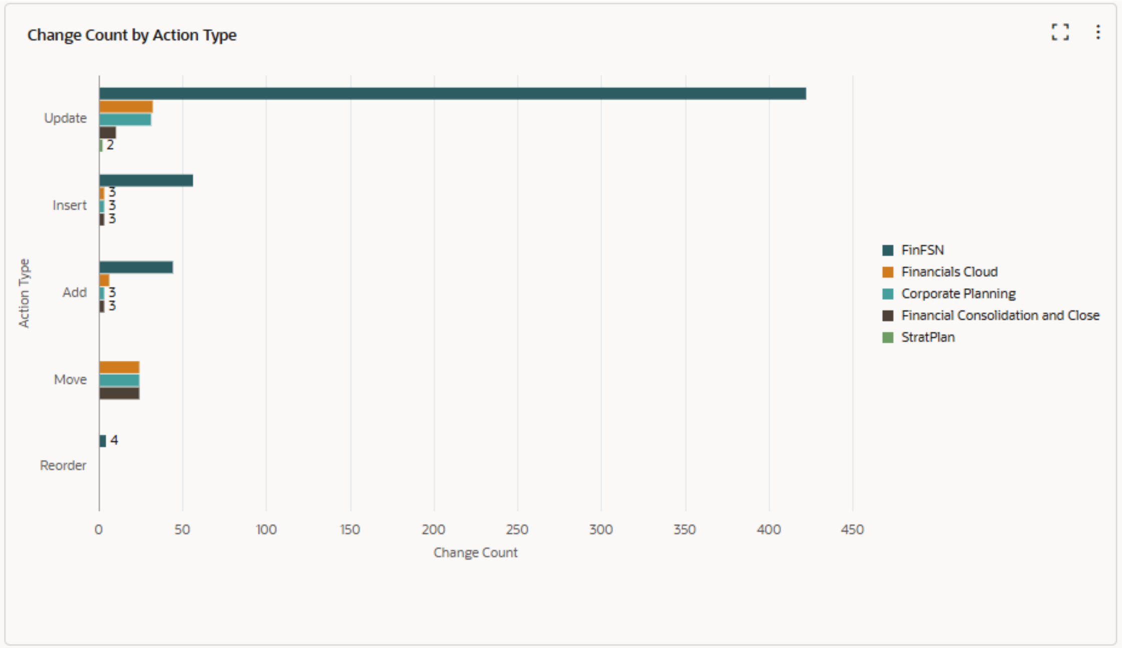
You can right-click a data point in the chart to view the specific transactions on that day. However, you can not drill across from this chart to another visualization.

Request Change Activity

The Request Change Activity card displays counts of completed changes by action type and request type.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

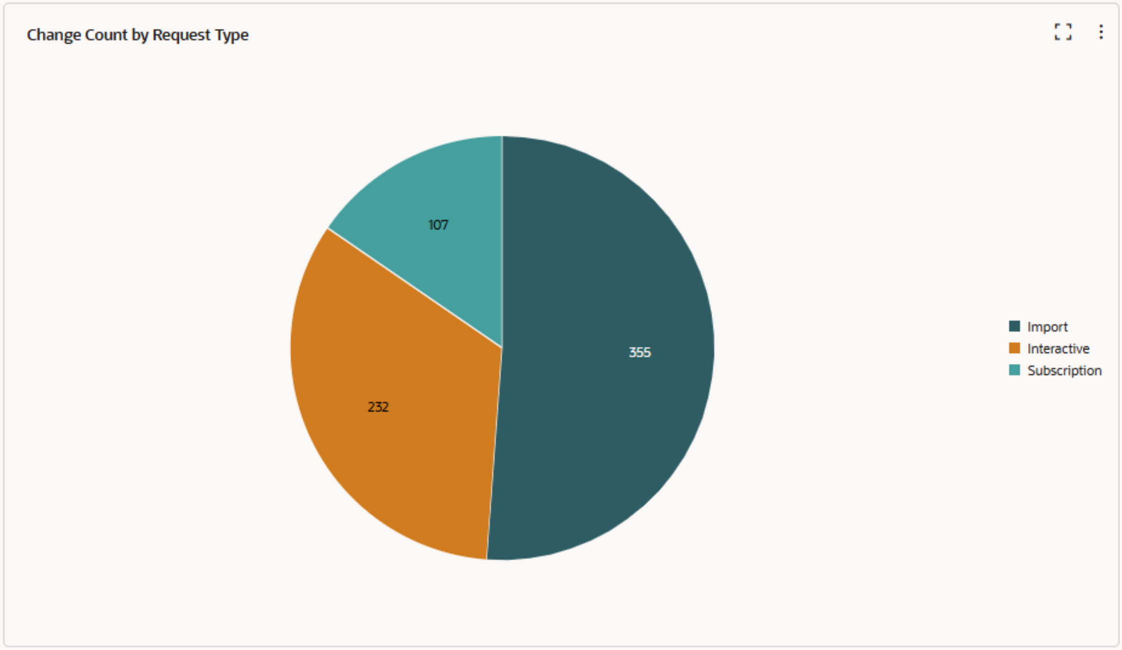
Change Count by Action Type



The **Change Count by Action Type** dashboard panel displays a visualization of the request change count by application per action type.

For each element in the chart, you can drill down to view the transaction history, drill across to view the changes in another visualization, or apply the specifics of that element to the page filter (see Working with Dashboard Panels).

Change Count by Request Type



The **Change Count by Request Type** dashboard panel displays a visualization of the request change count by application per request type.

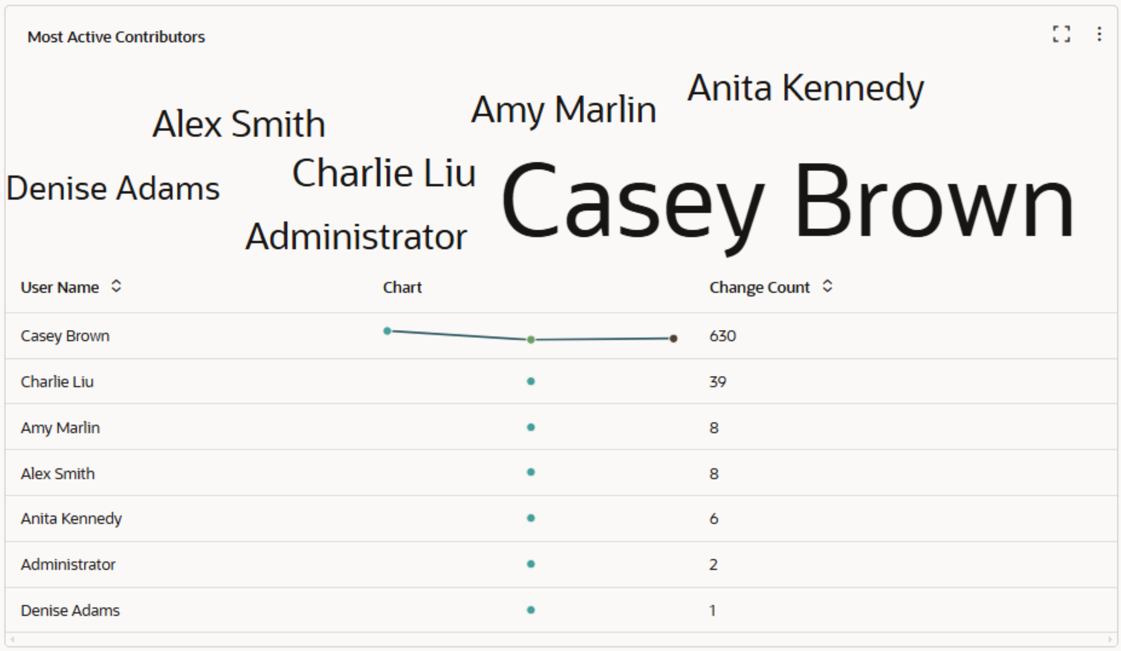
For each element in the chart, you can drill down to view the transaction history, drill across to view the changes in another visualization, or apply that element to the page filter (see Working with Dashboard Panels).

Active Contributors

The Active Contributors card displays visualizations of the users in your system that have made the most changes overall, by action type, and by application.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Most Active Contributors



User Name	Chart	Change Count
Casey Brown		630
Charlie Liu		39
Amy Marlin		8
Alex Smith		8
Anita Kennedy		6
Administrator		2
Denise Adams		1

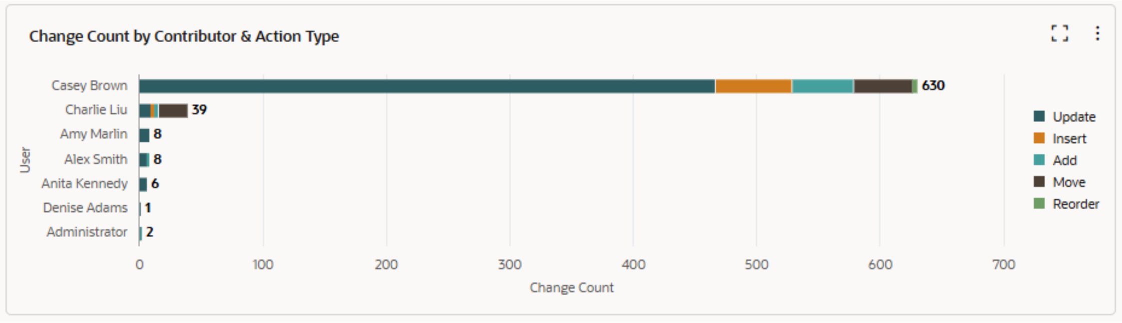
The **Most Active Contributors** dashboard panel displays the contributors (up to the top 100) with the highest total change count as determined by your page filter. The visualization is presented as a tag cloud, with contributors with more changes displayed in bigger font sizes. The contributors are also displayed in a table with a spark chart of the trend of changes over time (as determined by your filter) and a total change count per user.

From the tag cloud, you can drill down to view the transaction history for a particular user, drill across to view the changes for that user in another visualization, or apply the specifics of that user to the page filter (see Working with Dashboard Panels).

From the table, right-click any column to resize it. You can also right-click the User Name and Change Count columns to sort them in ascending or descending order.

Hover over a spark chart to view the data in the points on the chart.

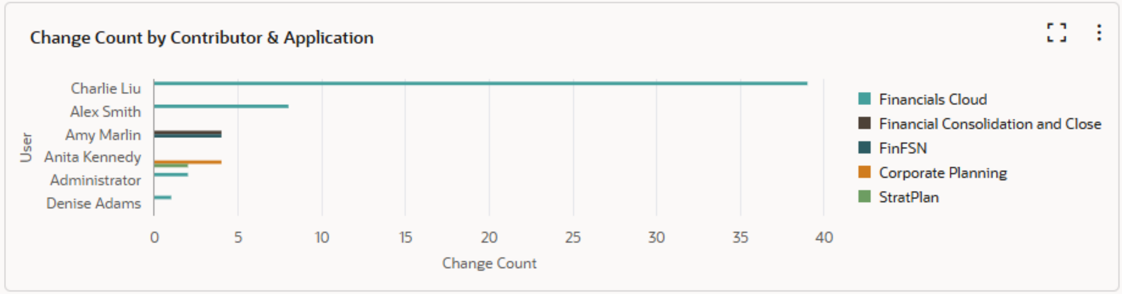
Change Count by Contributor and Action Type



The **Change Count by Contributor and Action Type** dashboard panel displays the total number of changes per user (as determined by your page filter) in a relative stacked bar chart, with each segment representing an action type.

For each element in the chart, you can drill down to view the transaction history for that user and action type, drill across to view the changes in another visualization, or apply the specifics of that element to the page filter (see Working with Dashboard Panels).

Change Count by Contributor and Application



The **Change Count by Contributor and Application** dashboard panel displays the total number of changes per user and application (as determined by your page filter) in a horizontal bar chart, with each segment representing a user and an application.

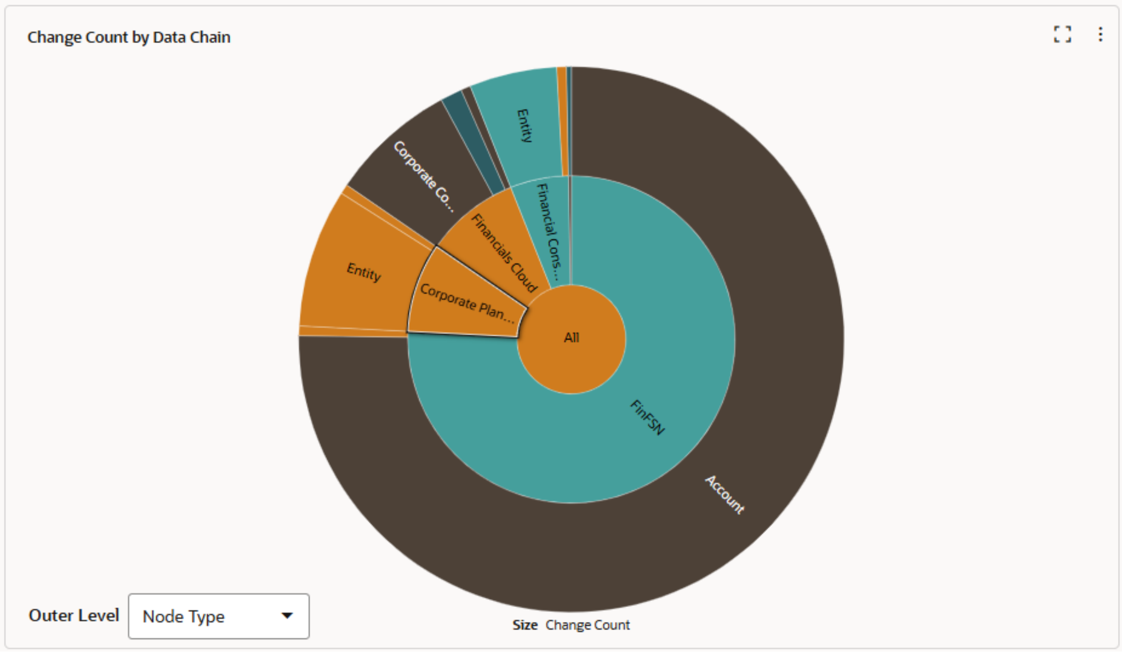
For each element in the chart, you can drill down to view the transaction history for that user and application, drill across to view the changes in another visualization, or apply the specifics of that element to the page filter (see Working with Dashboard Panels).

Applications Changed

The Applications Changed card displays the changes made to application by user and data chain object.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Change Count by Data Chain



The **Change Count by Data Chain** dashboard panel displays the changes by data chain object in a sunburst chart, with each segment of the chart representing a data chain object and the inner rings of the chart representing container objects for the next outer ring. For example, by default the inner ring represents all applications, the next outer ring represents each application, the next ring represents the dimensions in each application, and the outermost ring represents the node types in the dimensions.

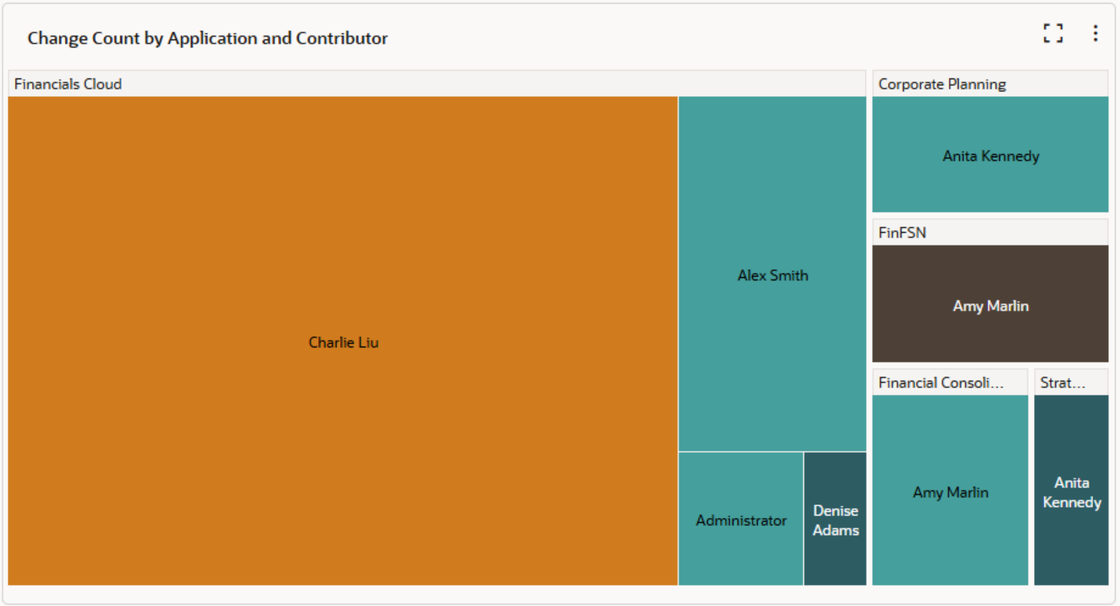
You can double-click on a segment to drill into that segment. For example, double-clicking on an application will zoom the chart in so that the application is in the center of the chart and the outer ring displays the dimensions in that application. Click the breadcrumb links in the upper left of the panel to navigate the chart back out again.

Tip:

Use the **Outer Level** drop down menu to toggle between displaying changes by node type or hierarchy set in the outermost ring.

For each element in the chart, you can drill down to view the transaction history, drill across to view the changes in another visualization, or apply the specifics of that element to the page filter (see Working with Dashboard Panels).

Change Count by Application and Contributor



The **Change Count by Application and Contributor** dashboard panel displays the total number of changes per application and user user in a treemap chart. Each segment of the chart represents the changes to a specific application by a specific user, with bigger segments representing greater numbers of changes.

From a segment for a specific application, in the property header click **Isolate** ⛶ to display the treemap for that application only. Click **Restore** to display all of the applications again.

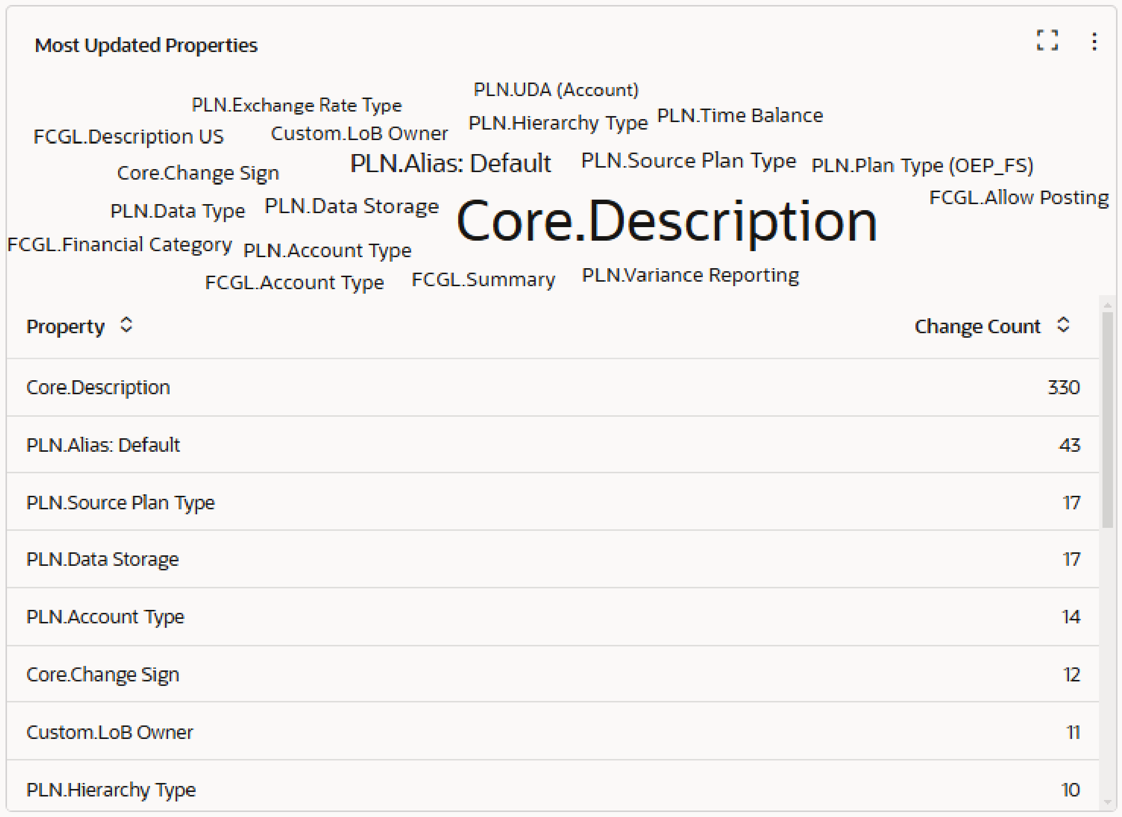
For each element in the chart, you can drill down to view the transaction history, drill across to view the changes in another visualization, or apply the specifics of that element to the page filter (see Working with Dashboard Panels).

Properties Changed

The Properties Changed card displays visualizations of the properties that were updated the most by your users.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Most Updated Properties



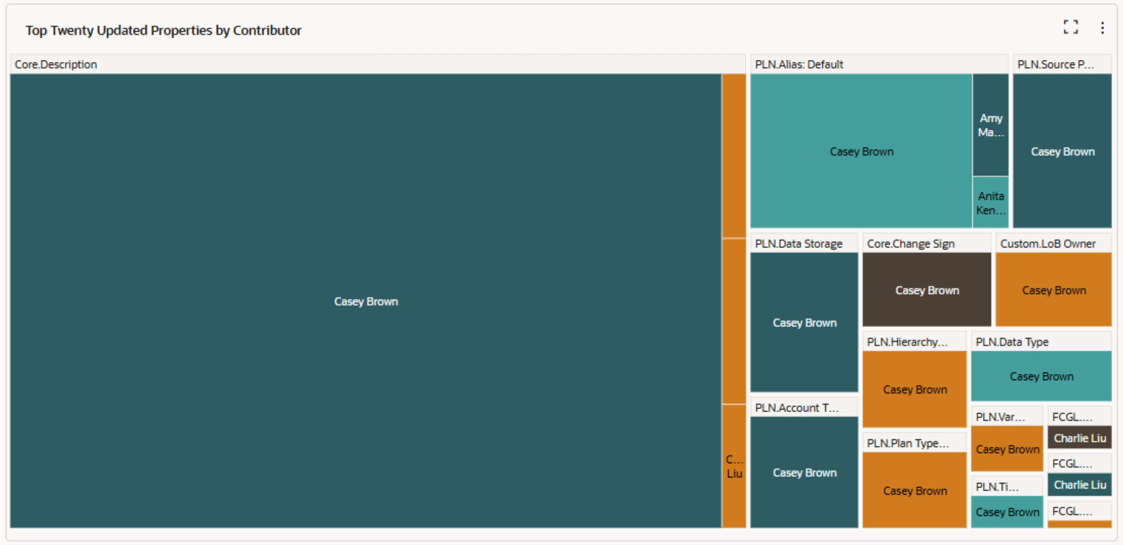
Property	Change Count
Core.Description	330
PLN.Alias: Default	43
PLN.Source Plan Type	17
PLN.Data Storage	17
PLN.Account Type	14
Core.Change Sign	12
Custom.LoB Owner	11
PLN.Hierarchy Type	10

The **Most Updated Properties** dashboard panel displays the properties (up to the top 100) with the highest total change count as determined by your page filter. The visualization is presented as a tag cloud, with properties with more changes displayed in bigger font sizes. The properties are also displayed in a table with the total change count for each property.

From the tag cloud, you can drill down to view the transaction history for a particular property, drill across to view the changes for that property in another visualization, or apply the specifics of that property to the page filter (see Working with Dashboard Panels).

From the table, right-click a column to resize it or to change the sort order.

Top Twenty Updated Properties by Contributor



The **Top Twenty Updated Properties by Contributor** dashboard panel displays the top twenty properties that have been updated per contributor in a treemap chart. Each segment of the chart represents the changes to a specific property by a specific user, with bigger segments representing greater numbers of changes.

From a segment for a specific property, in the property header click **Isolate** to display the treemap for that property only. Click **Restore** to display all of the properties again.

For each segment in the chart, you can drill down to view the transaction history, drill across to view the changes in another visualization, or apply the specifics of that segment to the page filter (see Working with Dashboard Panels).

Workflow Analysis Dashboard

The Workflow Analysis dashboard enables you to analyze historical workflow information for closed requests as a trend over time or by groupings such as request type and request stage.

This enables you to assess the impact of various aspects of your workflow on the time it takes for your requests to be completed and closed. For example, you can see if requests from a specific application or dimension are taking longer than expected, or if requests from a specific owner are taking longer to complete. For requests that are configured to submit automatically, you can assess how many of them are actually being auto-approved and where the potential issues are that might require manual intervention.

Considerations

- You must have at least *Data Manager* permission on a viewpoint in order to view workflow information for requests from that viewpoint. Users with the *Auditor* application role can view workflow information for all viewpoints.
- The Workflow Analysis dashboard displays workflow information for closed requests only. It does not display changes from Draft or In Flight requests.
- The dates for the request data in all charts are based on Coordinated Universal Time (UTC). Aggregated changes represented in the charts may vary by a day when compared to request summary due to time zone differences.

Workflow Analysis Scoreboard

The following scorecards are available from the Workflow Analysis Scoreboard. Each scorecard displays the name of the metric and the metric value as filtered by the filter bar.

- Closed Requests Overview
- Request Cycle Time
- Approval Cycle Time
- Auto Submit Analysis

Filter Bar

To view closed request information in your system, make selections from the following filters.

- **Data Chain**: Select the applications and dimensions to view requests for. Select one or more data chain objects in the drop down list. You can select a maximum of 15 items in a single drop down list.
  - **Application**: Select one or more applications to view requests for.
  - **Dimension**: For a selected application, select one or more dimensions to view requests for.

    > **Note:**
    >
    > The Dimension filter is available only when a single application has been selected in the Application filter.

- **View**: Select one or more views that you want to view requests for. You can select from all views for which you have at least *Data Manager* permission. You can select a maximum of 15 items in a single drop down list.
- **Request Type**: Select one or more request types that you want to view requests for. The following request types are available to select:
  - Consolidation
  - Import
  - Interactive
  - Subscription
  - Load
- **Request Status**: Select one or more request statuses that you want to view closed requests for. The following request statuses are available to select:
  - Completed
  - Consolidated
  - Rejected
- **Owner**: Select one or more request owners to view requests for.
- **Time Frame**: Select the time frame to view requests for. You can select only a single value for the Time Frame filter.
  From **Period**, select the time period to view changes for, or click **Date Range** and enter a custom date range (day, month and year). The following periods are available to select:

- Week to Date
- Month to Date
- Quarter to Date
- Year to Date
- Previous Week
- Previous Month
- Previous Quarter
- Previous Year

Closed Requests Overview

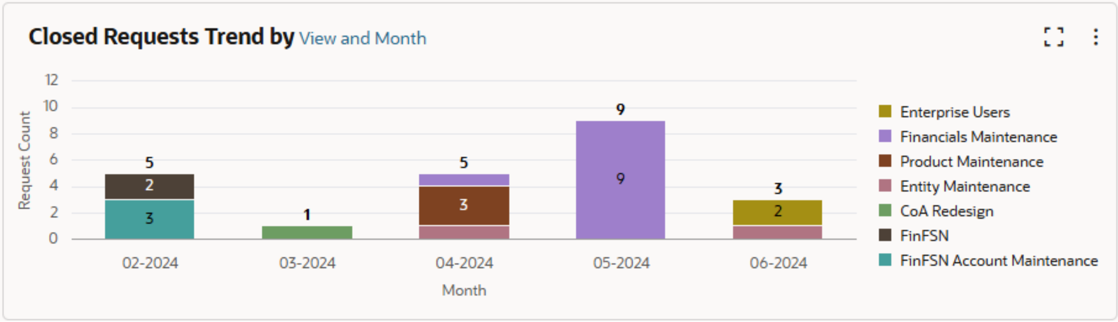
The Closed Requests Overview card displays trend information about request completion over time by data object.

The dashboard panels on this card help you identify the applications and dimensions that require longer cycle times to complete. You can drill across on an application or dimension to the Request Cycle Time or Approval Cycle Time cards to analyze other factors in play for those objects, such as the number of items or contributors involved.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels. Exceptions to these actions are described in the sections for each dashboard panel, below.

> **Note:**
>
> When you download the Closed Request Summary for any of the visualizations on this card, the request summary is grouped by dimension. If you have a request that contains request items from multiple dimensions, it will be displayed multiple times in the summary (once for each dimension).

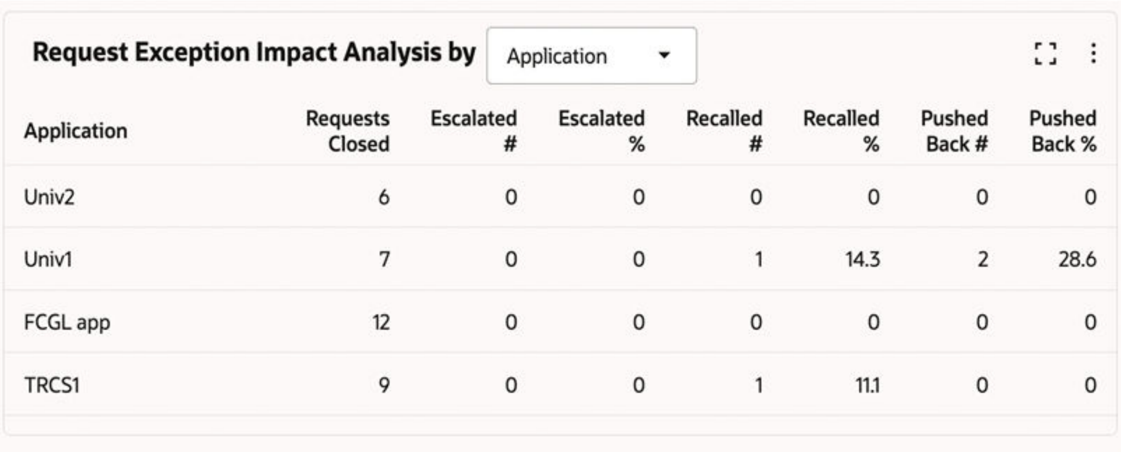
Closed Requests Trend by (Data Chain Object and Time Period)



The Closed Requests Trend by (Data Chain Object and Time Period) panel displays the number of closed requests by either view or application (or dimension if there is a single application selected in the Data Chain filter bar) over a specific time period. By default, it displays data by Application and Month. Click the hyperlink in the title to change the data chain object or to adjust the time period. You can select a time period within the Time Frame that is selected in the filter bar (for example, when the Time Frame is set to Year to Date, you can select Month or Quarter).

The data elements in this chart cannot be drilled across or applied to the current filter. You can use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Request Exception Impact Analysis by (Data Chain Object)



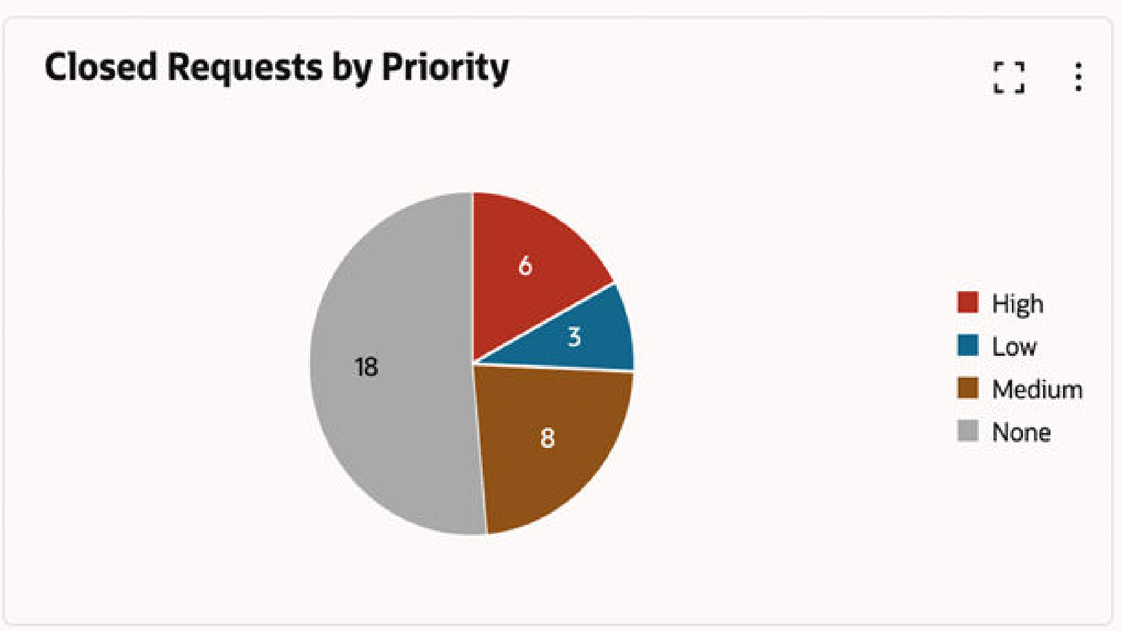
Request Exception Impact Analysis by Application

Application	Requests Closed	Escalated #	Escalated %	Recalled #	Recalled %	Pushed Back #	Pushed Back %
Univ2	6	0	0	0	0	0	0
Univ1	7	0	0	1	14.3	2	28.6
FCGL app	12	0	0	0	0	0	0
TRCS1	9	0	0	1	11.1	0	0

The Request Exception Impact Analysis by (Data Chain Object) panel displays information about how many requests had exceptions (such as escalations, recalls, and push backs) that may have impacted their time to close. You can display the panel by view or application (or dimension if there is a single application selected in the Data Chain filter bar). The total number of each type of exception is displayed, along with a percentage of that exception of the total number of closed requests.

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

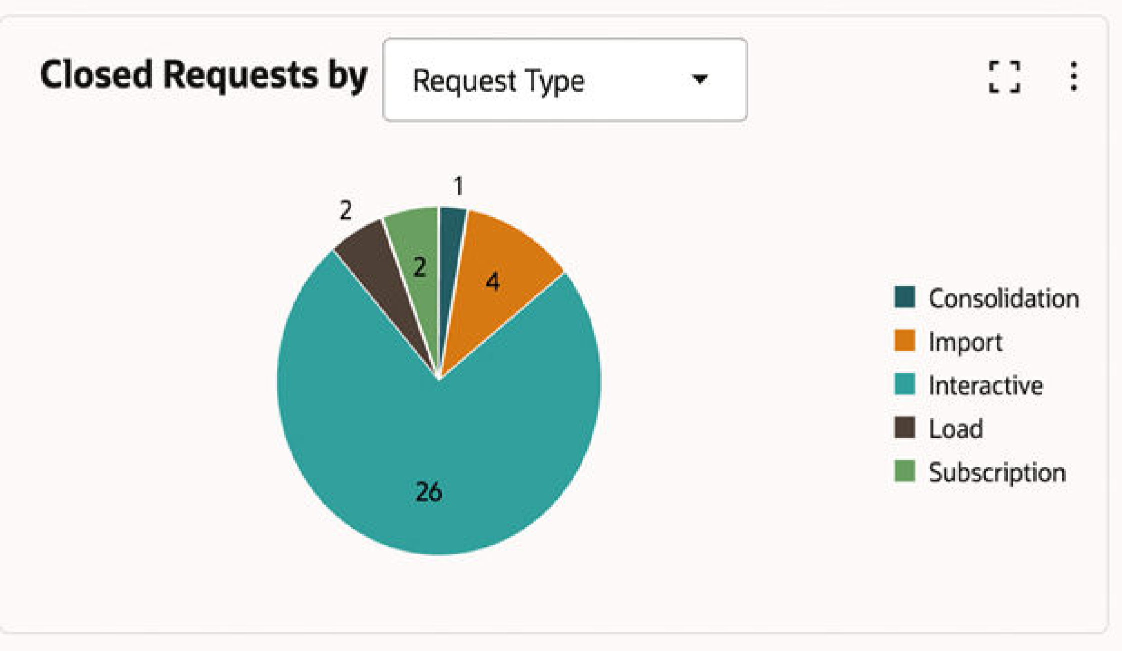
Closed Requests by Priority



The Closed Requests by Priority panel displays the total number of closed requests of each priority over the time period that is specified in the filter bar.

For each element in the chart, you can drill down to view the request activity. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Closed Requests by (Request Type or Status)



The Closed Requests by (Request Type or Status) panel displays the number of closed requests by either Request Type (such as Import, Interactive, or Subscription) or Request

Status (such as Completed, Consolidated, or Rejected) over the time period specified in the filter bar.

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Request Cycle Time

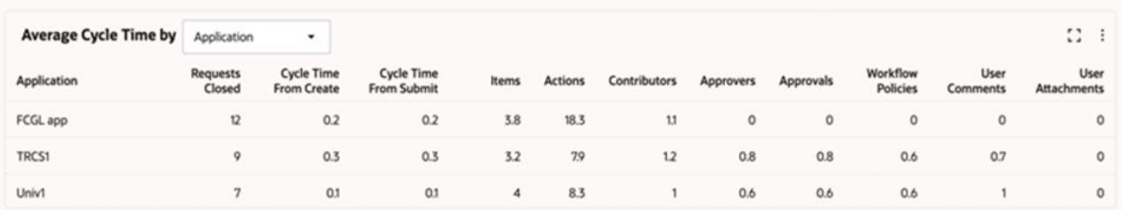
The Request Cycle Time card displays information on how long it takes requests to close based on various factors, such as the request type, the application, or the number of request items.

This enables you to analyze potential problem areas that can cause requests to take longer than expected to complete.

> **Note:**
>
> Cycle time in the dashboard panels is displayed in increments of days. So a cycle time of .5 represents 50% of a day, or 12 hours.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Average Cycle Time by (Application, Dimension, Request Type, or Request Owner)



Average Cycle Time by Application

Application	Requests Closed	Cycle Time From Create	Cycle Time From Submit	Items	Actions	Contributors	Approvers	Approvals	Workflow Policies	User Comments	User Attachments
FCGL app	12	0.2	0.2	3.8	18.3	1.1	0	0	0	0	0
TRCS1	9	0.3	0.3	3.2	7.9	1.2	0.8	0.8	0.6	0.7	0
Univ1	7	0.1	0.1	4	8.3	1	0.6	0.6	0.6	1	0

The Average Cycle Time by (Application, Dimension, Request Type, or Request Owner) panel displays statistics on how long on average it took for requests to complete based on the application, dimension (if one application is selected in the filter bar), request type, or request owner.

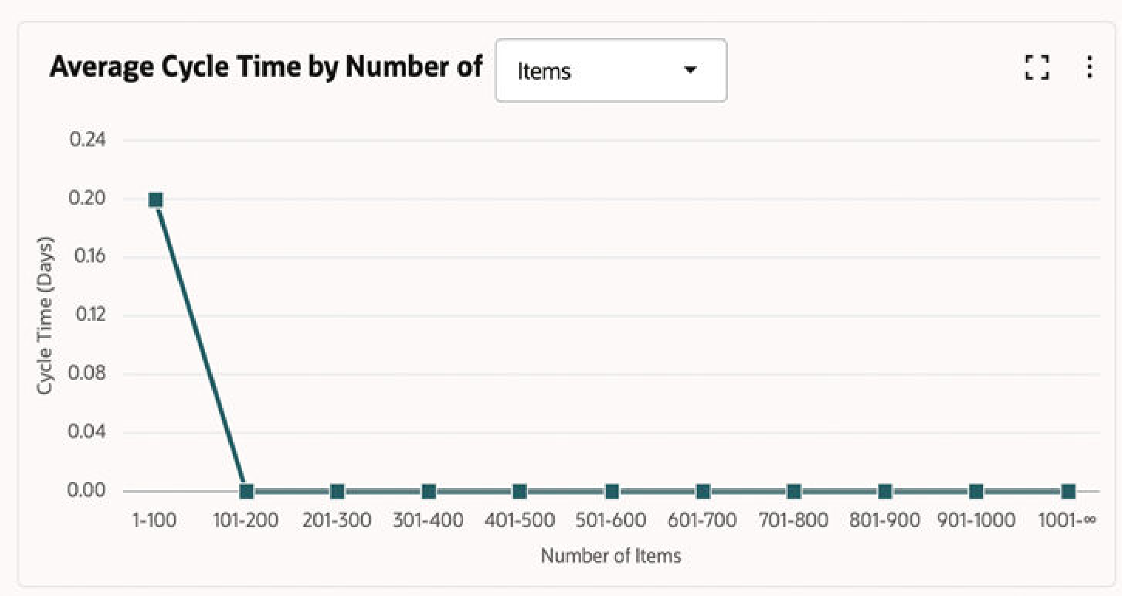
The columns in the panel display the following information:

- **Requests Closed**: The number of requests (per application, dimension, request type, or request owner) that were closed in the time frame specified in the filter bar.
- **Cycle Time From Create** is the average amount of time that a request took to close from the time a request is initially created (for example, clicking **New Request** for an interactive request).
- **Cycle Time from Submit** is the average amount of time that a request took to close from the time that a request owner clicked the **Submit** button.

- **(All other columns)**: Display various statistics about the requests, such as the average number of items and request actions that were in the requests, and the average number of approvers and request approvals per request.

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

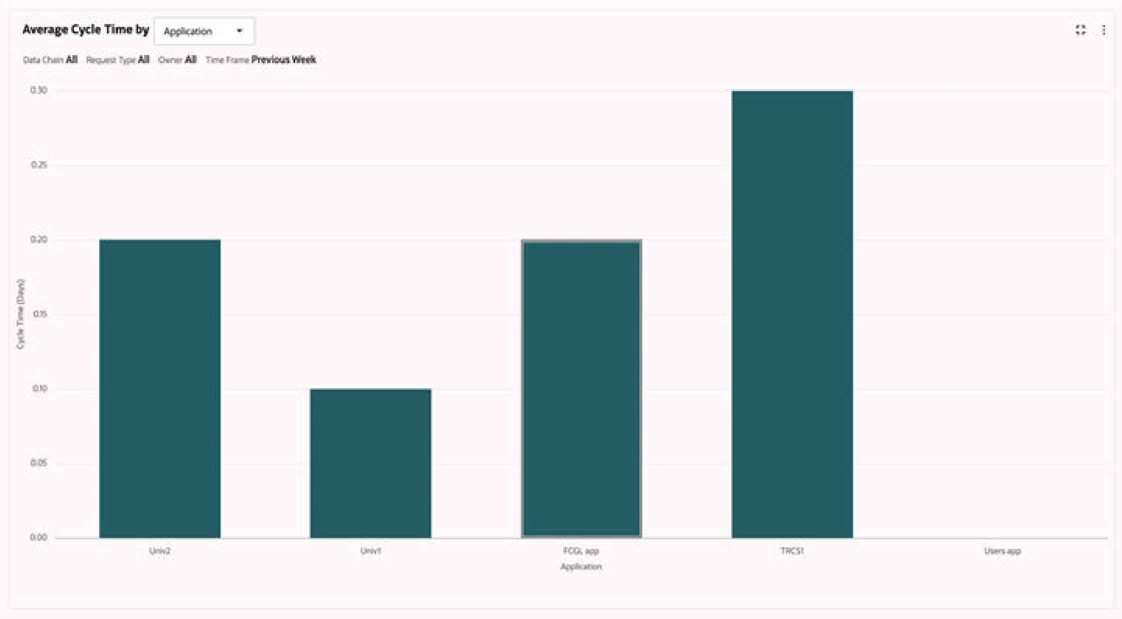
Average Cycle Time by Number of (Items or Contributors)



The Average Cycle Time by Number of (Items or Contributors) panel displays a distribution of the average number of days it took for requests to be closed by the number of items or contributors on the request.

The data elements in this chart cannot be drilled across or applied to the current filter. You can use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Average Cycle Time by (Application or Dimension)



Average Cycle Time by Application

Data Chain All Request Type All Owner All Time Frame Previous Week

The Average Cycle Time by (Application or Dimension) panel displays the average number of days it took for requests to be closed broken down by application or dimension (if a single application is selected in the filter bar).

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

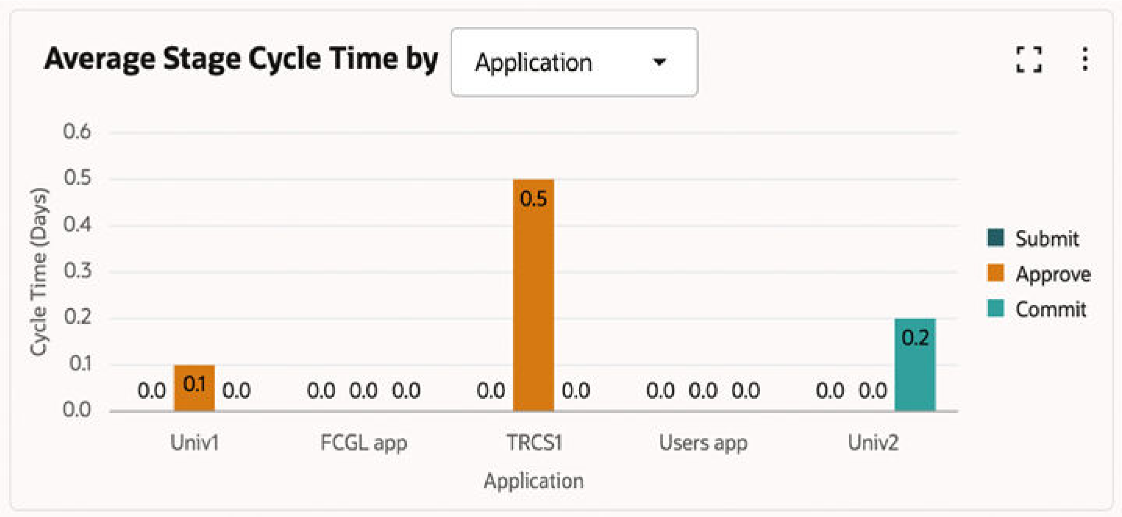
Approval Cycle Time

The Approval Cycle Time card displays information on how long requests are spending in each workflow stage and how long it takes for requests to be approved by priority or request owner.

> **Note:**
>
> Cycle time in the dashboard panels is displayed in increments of days. So a cycle time of .5 represents 50% of a day, or 12 hours.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Average Stage Cycle Time by (Application, Dimension, or Request Type)



The Average Stage Cycle Time by (Application, Dimension, or Request Type) panel displays the average number of days that requests spent in the Submit, Approve, and Commit stages by application, dimension (if there is a single application selected in the filter bar), or request type.

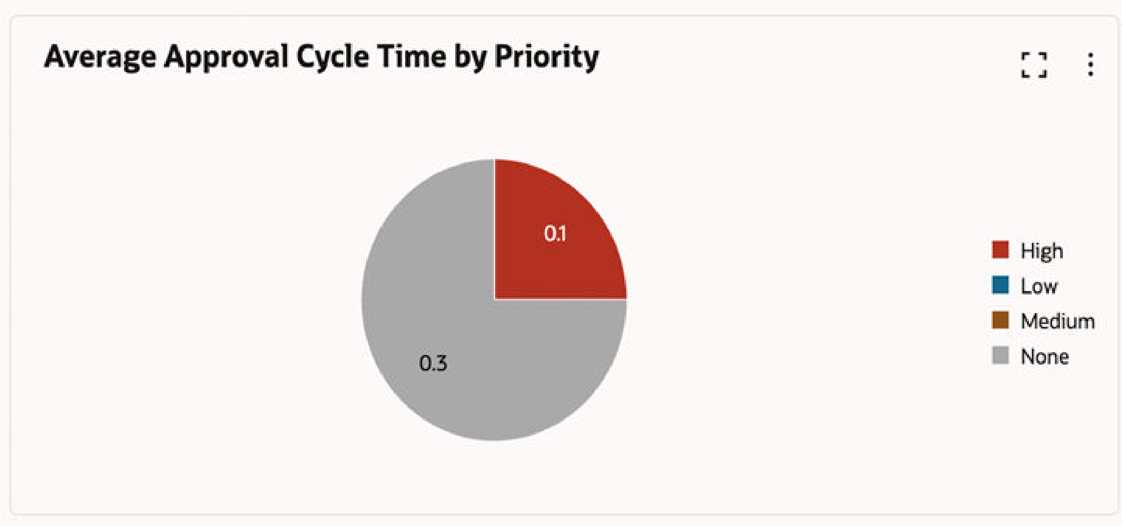
The start and end time for the stages is defined as follows:

Table 17-1 Stage Start and End Times

Stage	Start	End
Submit	A request is initially created (for example, clicking **New Request** for an interactive request).	The request is submitted.
Approve	Request is promoted to the Approve stage	Final approval is received
Commit	Request is promoted to the Commit stage	Final commit is received

For each element in the chart, you can drill down to view the request activity. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

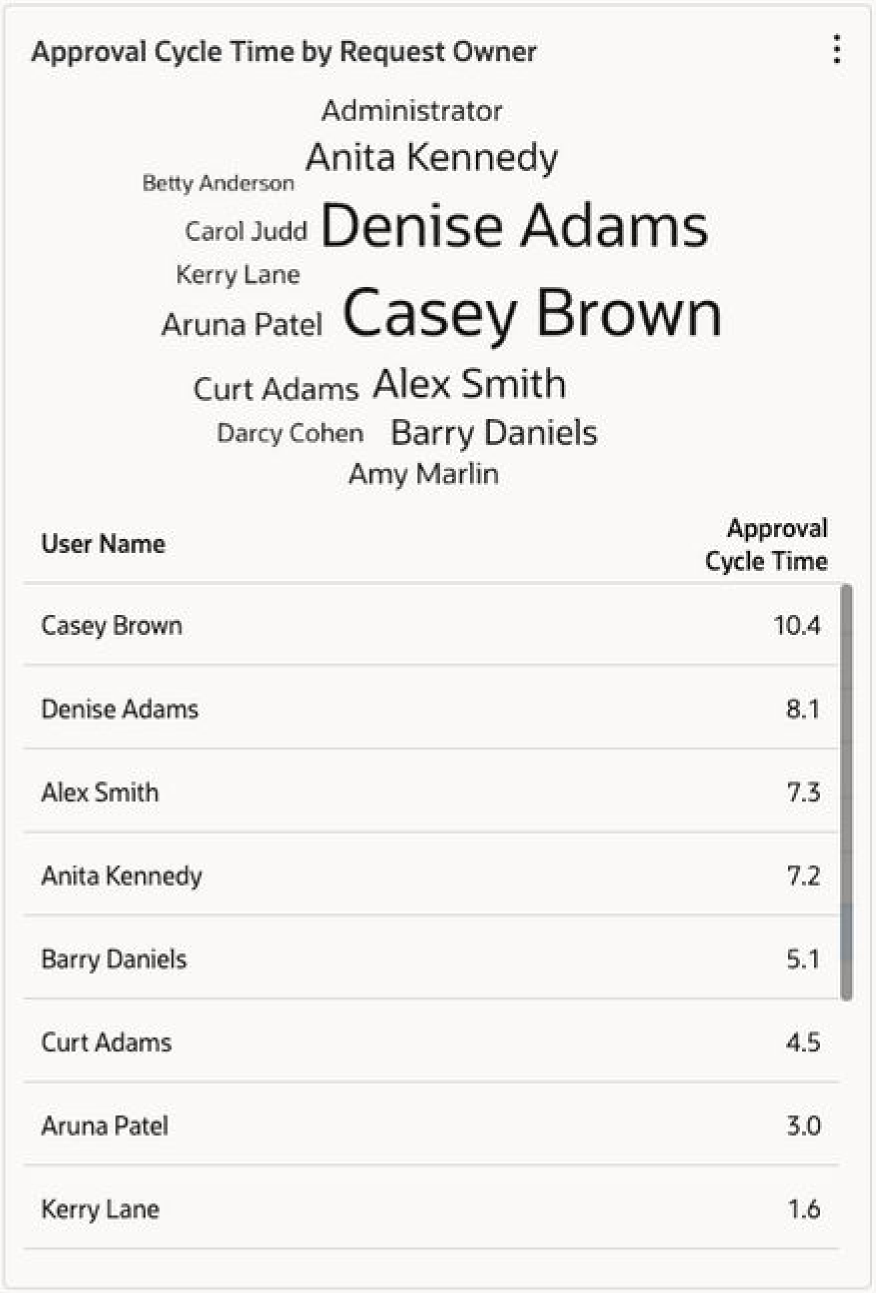
Average Approval Cycle Time by Priority



The Average Approval Cycle Time by Priority panel displays a pie chart showing the average number of days it took for requests to be approved based on their priority.

The data elements in this chart cannot be drilled across or applied to the current filter. You can use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Approval Cycle Time by Request Owner



Approval Cycle Time by Request Owner

User Name	Approval Cycle Time
Casey Brown	10.4
Denise Adams	8.1
Alex Smith	7.3
Anita Kennedy	7.2
Barry Daniels	5.1
Curt Adams	4.5
Aruna Patel	3.0
Kerry Lane	1.6

The Approval Cycle Time by Request Owner panel displays the average number of days it took for requests to be approved, grouped by request owner (that is, the person who submitted the request). The visualization is presented as a tag cloud, with request owners with longer cycle times displayed in bigger font sizes. The owners are also displayed in a table with the average approval cycle time for each owner.

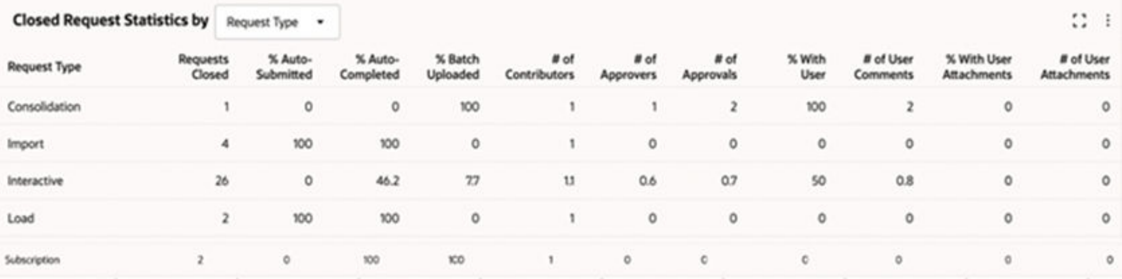
For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Auto Submit Analysis

The Auto Submit Analysis card displays information on how many requests are being submitted and completed and closed automatically without manual intervention.

For details on the actions that you can take in the dashboard panels, see Working with Dashboard Panels.

Closed Request Statistics by (Request Type or Data Chain Object)



Closed Request Statistics by Request Type

Request Type	Requests Closed	% Auto-Submitted	% Auto-Completed	% Batch Uploaded	# of Contributors	# of Approvers	# of Approvals	% With User	# of User Comments	% With User Attachments	# of User Attachments
Consolidation	1	0	0	100	1	1	2	100	2	0	0
Import	4	100	100	0	1	0	0	0	0	0	0
Interactive	26	0	46.2	7.7	1.1	0.6	0.7	50	0.8	0	0
Load	2	100	100	0	1	0	0	0	0	0	0
Subscription	2	0	100	100	1	0	0	0	0	0	0

The Closed Request Statistics by (Request Type or Data Chain Object) panel displays a visualization of percentage of requests that were submitted and completed automatically by request type, application, dimension (if a single application is selected in the filter bar), or view.

The columns in the panel display the following information:

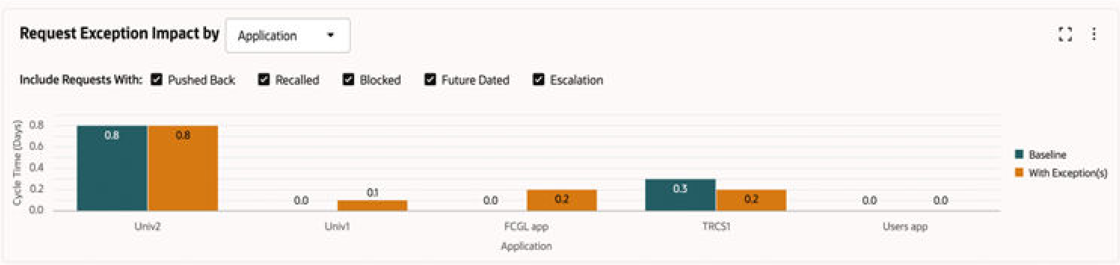
- **Requests Closed**: The number of requests (per request type or data chain object) that were closed in the time frame specified in the filter bar.
- **% Auto-Submitted**: The percentage of requests that were submitted automatically.
- **% Auto-Completed**: The percentage of requests that were completed automatically.
- **% Batch Uploaded** : The percentage of requests in which the request items were loaded from a request load file.

> **Note:**
>
> This percentage will always be 100 for Subscriptions, since they use request load files to create request items. It will always be zero for Imports and Loads, since they do not use request load files to create request items.

- **(All other columns)**: Display various statistics about the requests, such as the average number of contributors and approvers and the average number and percentage of requests that have user comments and attachments. This information can help you analyze the impact of that complexity on total cycle time.

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Request Exception Impact by (Request Type or Data Chain Object)



The Request Exception Impact by (Request Type or Data Chain Object) panel shows the impact of various request exceptions (such as push backs and recalls) on the number of days it took for requests to be completed by request type, application, dimension (if there is a single application selected in the filter bar), or view as compared to the baseline of requests with no exceptions.

Use the **Include Requests With** checkboxes to filter the exception types that are displayed in the chart.

For each element in the chart, you can drill down to view the request activity, drill across to view the requests in another visualization, or apply the specifics of that element to the page filter. You can also use the **Actions** menu ⋮ to view and download the request summary in the visualization. See Working with Dashboard Panels.

Part II
Administration

In Oracle Fusion Cloud Enterprise Data Management there are two categories of tasks: data management and administration. This section of the guide describes administration tasks.

Administration tasks include:

- Registering applications
- Importing and exporting dimensions
- Creating views, data chains, and alternate viewpoints

See Getting Started with Administration.

Data management tasks include creating viewpoints and updating data such as nodes and properties, see Getting Started with Data Management.

18

Getting Started with Administration

Administration tasks include registering applications, importing and exporting dimensions, creating views, data chains and alternate viewpoints. Service administration tasks include adding users, groups and maintaining the environment.

Videos

Your Goal	Watch This Video
Review helpful information on getting started as a Service Administrator.	▶ Getting Started with Enterprise Data Management Cloud for Service Administrators
Review helpful information on getting started as an application owner.	▶ Getting Started with Enterprise Data Management Cloud for Application Owners

Identity Domain Administration Tasks

The Identity Domain Administrator creates users within Oracle Cloud My Services and assigns predefined roles, see Users in a Traditional Cloud Account and Understanding Predefined Roles.

Service Administration Tasks

A Service Administrator is a user with the predefined role of Service Administrator, and does these tasks:

1. Assigns application roles to users within Access Control, see Access Control and Understanding Application Roles and Permissions.
2. Migrates artifacts across test and production environments. See Using Migration.
3. Performs daily maintenance, see Using the Maintenance Snapshot.

> **Note:**
>
> You must have the predefined role of Service Administrator to do these tasks.

Administration Tasks

An Administrator is a user with the predefined role of User. This table lists administration tasks and the minimum required roles and permissions needed to perform the tasks.

Task	Role	Permission	See
Assign permissions		*Owner* or *Metadata Manager* on the dimension to assign permissions to the data chain objects in that dimension *Owner* on the view to assign view permissions	See Understanding Application Roles and Permissions.
Create sample application	*You need the Service Administrator predefined role to create sample applications.*		Creating a Sample Application
Create a View	*Views - Create application role*		Working with Views
Register applications	*Application - Create application role*		Understanding Registering Applications
Modify applications		*Owner* or *Metadata Manager* on the application	Understanding Modifying Applications
Manage the lifecycle of data objects with archiving		*Owner* or *Metadata Manager* on the data object	Understanding the Lifecycle of Data Objects and the Data Chain
Import dimensions		*Data Manager* or *Metadata Manager* on the dimension that you want to import to	Importing Dimensions
Export dimensions		*Data Manager* or *Metadata Manager* on the dimension that you want to export from	Exporting Dimensions
Create alternate viewpoints		*Data Manager* or *Metadata Manager* on the dimension and *Owner* on the view	Defining Alternate Views and Viewpoints
Create data chain objects		*Owner* or *Metadata Manager* on the dimension	Creating Node Types Creating Hierarchy Sets Creating Node Sets Creating a Viewpoint

Task	Role	Permission	See
Manage data		*Participant(Read)* on the data chain object to view data *Participant(Write)* on the data chain object to make changes using requests *Owner* on an application or dimension to configure an approval policy for data objects in that application or dimension	Working with Properties Understanding Validations and Constraints Configuring Policies
Recreate service	*You need the Service Administrator predefined role to recreate the service.*		Recreating the Service

To perform some administration tasks, you need roles which are assigned by the Service Administrator:

- You need the Application - Create role to register applications. After a application is registered you are assigned Owner permission to the application, enabling you to update the application and data, create data objects, and to give users permission to use the application.
- You need the Views - Create role to create views. After a view is created you are assigned *Owner* permission to the view, enabling you to update the view, and to give users permission on the view.

> **Note:**
>
> See Understanding Application Roles and Permissions.

Creating the Enterprise Data Management Business Process

Oracle Fusion Cloud Enterprise Performance Management allows you to deploy two Enterprise Data Management options:

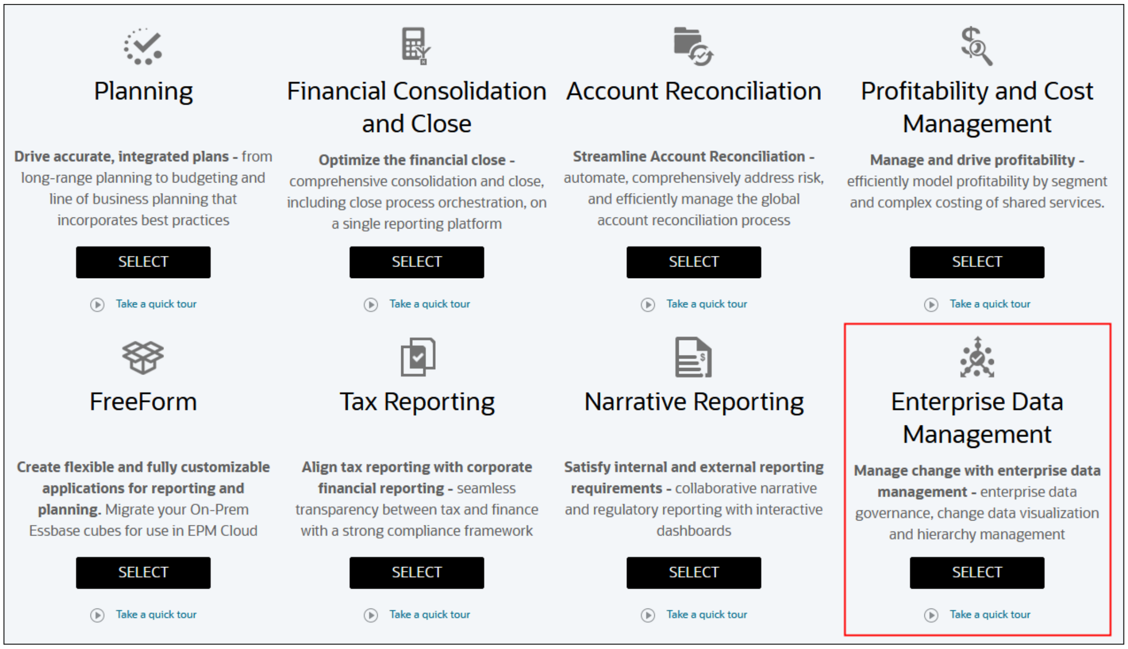
- Standalone Oracle Enterprise Data Management Cloud
- EPM Enterprise Cloud Service - Enterprise Data Management Business Process

(For more details, see Enterprise Data Management Cloud in *Getting Started Guide for Administrators*)

For the Standalone Oracle Enterprise Data Management Cloud option, your environment is already created and you can start creating the sample application, a new application, or migrating an existing snapshot. The steps for performing these tasks are described in Recreating the Service.

For the EPM Enterprise Cloud Service - Enterprise Data Management Business Process option, you must create the Enterprise Data Management business process before you access

Enterprise Data Management. On the landing page, click **SELECT** under **Enterprise Data Management**. This process takes approximately 20 minutes. The environment is not accessible while the business process creation is in process.



Planning

Drive accurate, integrated plans - from long-range planning to budgeting and line of business planning that incorporates best practices

SELECT

Take a quick tour

Financial Consolidation and Close

Optimize the financial close - comprehensive consolidation and close, including close process orchestration, on a single reporting platform

SELECT

Take a quick tour

Account Reconciliation

Streamline Account Reconciliation - automate, comprehensively address risk, and efficiently manage the global account reconciliation process

SELECT

Take a quick tour

Profitability and Cost Management

Manage and drive profitability - efficiently model profitability by segment and complex costing of shared services.

SELECT

Take a quick tour

FreeForm

Create flexible and fully customizable applications for reporting and planning. Migrate your On-Prem Essbase cubes for use in EPM Cloud

SELECT

Take a quick tour

Tax Reporting

Align tax reporting with corporate financial reporting - seamless transparency between tax and finance with a strong compliance framework

SELECT

Take a quick tour

Narrative Reporting

Satisfy internal and external reporting requirements - collaborative narrative and regulatory reporting with interactive dashboards

SELECT

Take a quick tour

Enterprise Data Management

Manage change with enterprise data management - enterprise data governance, change data visualization and hierarchy management

SELECT

Take a quick tour

After the pre-configuration is complete, use these steps:

1. Sign in to your EPM Enterprise Cloud Service environment as a Service Administrator. See Accessing EPM Cloud in *Getting Started Guide for Administrators*.
2. Select an option to create the business process.
   - **Create a sample application**: Click **CREATE** to create a sample business process with data and artifacts. You can use this ready-to-use business process for testing, exploration of functional areas and tutorials. See Creating a Sample Application
   - **Create a new application**: Click **START** then register your application. See Understanding Registering Applications
   - **Migrate**: Click **MIGRATE** to import a snapshot that you previously used on your environment. See What Applications Can I Migrate to EPM Standard Cloud Service and EPM Enterprise Cloud Service? in *Getting Started Guide for Administrators* for prerequisites and snapshot compatibility, and these topics in *Administering Migration*:
     - Backing up Artifacts and Application
     - Uploading Archives to the Service
     - Importing Artifacts and Application from a Snapshot

Creating a Sample Application

The sample application consists of pre-loaded sample data and related artifacts for Oracle Fusion Cloud Enterprise Data Management. Using this out-of-box environment, you can explore the capabilities of Cloud EDM, get familiar with its features, and use tutorials before you start working with your own data. The sample application includes examples for Planning and Planning Modules, FreeForm, Financial Consolidation and Close, Financials Cloud General Ledger, and E-Business Suite General Ledger application types.

The first time you use Cloud EDM you have an option to create a sample application. Later on, if you want to use the sample application and you are a service administrator, you need to recreate the service, which removes all applications and artifacts, then you can create a sample application. See Recreating the Service.

For example, if you want to use a tutorial with the sample application:

1. Back up your service.
2. Recreate your service.
3. Create the sample application and do the tutorial.
4. When you are done with the tutorial, you can restore your service using the backup from step 1.

> **Note:**
>
> Tutorials provide sequenced instructions that may include screenshots, sequenced videos, and documentation to help you learn a topic. See Tutorials on the Help Center.

Recreating the Service

Use the Recreate Service command to create a new business process, or to create a new application.

> **Note:**
>
> Using the Recreate Service command results in the loss of existing applications and artifacts. Perform a complete backup before you use this command.

To recreate your service, click your user name at the top-right corner of the screen, then click Recreate Service. This process takes approximately 20 minutes and your environment is not accessible until the process completes. See Recreate Service.

There are two scenarios for recreating your service.

1. If you are using the Oracle EPM Cloud Service, you can recreate your service to set up a different business process, see About the EPM Cloud Services.
2. If you are using Oracle Fusion Cloud Enterprise Data Management, you can recreate your service and then create new applications. After the recreation is complete, use one these steps :
   - **Create a sample application**: Click **CREATE** to create a sample business process with data and artifacts. You can use this ready-to-use business process for testing, exploration of functional areas and tutorials.
   - **Create a new application**: Click **START** then register your application. See Understanding Registering Applications
   - **Migrate**: Click **MIGRATE** to import a snapshot that you previously used on your environment.

     See these topics in *Administering Migration*:

- Backing up Artifacts and Application
- Uploading Archives to the Service
- Importing Artifacts and Application from a Snapshot

Customizing Your Users' Display

Service administrators can customize the display for all users by changing the theme, or adding a custom logo or background image.

Available Themes

The following themes are available to customize your users' display:

- **Oracle**: Default look and feel for the Redwood Experience. You can optionally configure your own logo, but not the home page background images or brand color.
- **Custom Dark**: Darker appearance that enables you to optionally configure your own logo and home page background images and brand color.
- **Custom Light**: Lighter appearance that enables you to optionally configure your own logo and home page background images and brand color.

> **Tip:**
>
> Click the **See Examples** link to see sample images of the Redwood themes with and without custom backgrounds.

To customize the display for all users:

1. From the Home page, click **Tools**, then **Appearance**.

   > **Note:**
   >
   > Only service administrators can see the **Appearance** icon.

2. In the **Theme** drop-down menu, select the Oracle, Custom Light, or Custom Dark theme.
3. In **Logo Image**, accept the default or select **Upload Image** and select a logo file. See Logo and Background Image Requirements.
4. For Custom Light and Custom Dark themes only:
   a. In **Home Background Image**, accept the default or select **Upload Image** and select an image file to display on the home page background. See Logo and Background Image Requirements.
   b. In **Brand Color**, click the colored circle and select from one of the predefined colors for the pattern stripe image in the page header.
5. In **Display Business Process Name**, select **Yes** to display the business process name next to the logo on the Home page and on the tab when a browser tab is opened. Select **No** to hide the business process name on the Home page and to display **Oracle Applications** on browser tabs.
6. Click **Save**.

Logo and Background Image Requirements

For the logo and background image files, select a file of type `.jpg`, `.png`, or `.gif`. The maximum file size is 5 MB.

> **Tip:**
>
> When switching to a new theme, customers using a custom background image might need to ensure that the color contrast for icons and labels is appropriate. To remedy, consider choosing a different theme or a suitable background.

Note the following about the logo and background images:

- **Logo Image**: Images smaller than 125 pixels wide and 25 pixels high are displayed without resizing them. For large image logos, Oracle recommends that you maintain a 5:1 ratio so the image can be scaled without distortion.
- **Background Image**: The default size for the background image is 1024x768. If you use a larger background image, the image is scaled to fit the resolution setting of your display. If you want your background image to fit both a browser and a mobile device, Oracle recommends that you size the image so that it fits your biggest screen (or highest resolution device). The background image is centered horizontally.

  > **Note:**
  >
  > Background images with non-standard sizes may display clipping issues. You may have to edit the image (for example, by adding extra margin area in the background) to reduce or prevent the clipping.

Working with Home Page Layouts

Service Administrators can configure and assign different home page card and cluster layouts for different users based on their predefined role or user group.

> **Tip:**
>
> A *card* on the home page is a user element that provides access to functionality. A *cluster* is a group of cards. See Cards and Clusters.

Considerations

- Service Administrators will always use the Default layout, which displays all cards and clusters on the home page, even if that user is also part of a user group that is assigned a different layout. The Default layout cannot be changed.
- You can assign a home page layout to the User predefined role or to a single user group only. You cannot assign a layout to multiple user groups.
- Making a card or cluster visible in a layout does not automatically grant view access to that card or cluster to the users of that layout. If a user does not have the required role to see a card or cluster, that card or cluster is not displayed even if it is set to display in the layout.

- If there is no layout assigned to a user's group, the layout assigned to the User role is used. If there is no layout for the User role, the Default layout is used.
- If a user belongs to more than one user group that has a layout assigned to it, that user will use the first layout that applies to their group in the list (sorted alphabetically by layout name).
- Changes to the home page layout are also reflected in the Navigator menu.

Creating a Home Page Layout by Copying an Existing Layout

You create a new home page layout by copying an existing layout and modifying it. When you first create a layout, it is inactive. You must activate the layout in order for your users to use it. See Activating, Inactivating, and Deleting Home Page Layouts.

1. From the home page, click **Tools** and then **Layouts**.
2. In the Actions column of the layout that you want to copy, click **Actions** ⋮, and then select **Copy**.
   A copy of the home page layout is displayed in the layout editor.

Editing a Home Page Layout

From the layout editor, perform the following actions:

1. To open the layout editor, click the name of the layout or click **Actions** ⋮, and then **Edit**
2. In the layout editor, edit the name and optionally the description of the layout. The name must be unique across layouts.
3. In **Applies To**, select one of the following:
   - **User Group**: Use the drop down menu to select a user group to assign to the layout. You can only choose one group.
   - **Role**: The User predefined role is selected and cannot be changed.
4. In **Layout**, perform these actions to customize the home page layout:
   - In the **Visible** column, click the check mark to toggle whether or not a card or cluster is visible on the home page.

     > **Note:**
     >
     > At least one top level card must be visible. If a cluster is visible, it must have at least one visible card in it. If a cluster is set to Visible but all of the cards in that cluster are hidden by a user's role, that cluster is not displayed.

   - In the **Order** column, click the up and down arrows to modify the sort order of cards and clusters on the home page.
   - In the **Move To** column, click **Action** ⋮ and select an option to move a card in and out of a cluster:
     - **Top Level**: Display the card directly on the home page
     - **Tools**: Display the card inside the Tools cluster
     - **Information Model**: Display the card inside the Information Model cluster

> **Note:**
>
> You cannot move a cluster inside of another cluster.

5. Click **Save**.

Activating, Inactivating, and Deleting Home Page Layouts

In the Actions column, click **Action** ⋮ and perform an action:

- **Activate**: Changes a layout to active.

  > **Note:**
  >
  > If you activate a layout for a user group or role that already has an active layout, you are prompted to confirm that you want to switch that user group or role to the layout that you are activating.

- **Inactivate**: Changes a layout to inactive.
- **Delete**: Deletes a layout.

  > **Note:**
  >
  > You are prompted to confirm that you want to delete a layout. The system indicates whether or not the layout is active.

Creating, Editing, and Deleting Announcements

Service administrators can create and publish announcements to users about upcoming events such as system maintenance, upgrades, and new features. Announcements are displayed in the Announcements area on the Activity tab on the Home page.

Creating Announcements

1. From the home page, click **Tools**, and then click **Announcements**.
2. Click **Create**.
3. Enter a subject header (maximum 100 characters), a start date and time, and optionally an end date and time for the announcement.
4. Use the rich text editor to add the content of the announcement (maximum 1,000 characters).
5. Click **Save and Close** to save the announcement.
   The announcement is displayed on the Announcements page with the most recent active announcements on top.

To edit an announcement:

Editing Announcements

1. From the Announcements page, click the subject header of the announcement that you want to edit.

2. Edit the subject header, start and end dates, and content of the announcement.
3. Click **Save and Close**.

Deleting Announcements

1. From the Announcements page click **Action** ⋮ in the Actions column of the announcement that you want to delete, and select **Delete**.
2. If the announcement is currently active, confirm that you want to delete it.

Configuring System Settings

You configure system settings to control system-level behavior such as email notifications. Use the **Settings** card in the **Tools** cluster to access system settings. You must be a Service Administrator to configure system settings.

Notification Settings

Notification settings enable you to configure the way the system sends email notifications.

1. From **Tools**, select **Settings**.
2. Perform an action:
   - In **Send Notifications**, click the check box to enable or disable sending email notifications for the entire system.
   - In **Email Subject Prefix**, enter a prefix for the subject line of system-generated emails to identify the environment from where the email was sent.
   - In **Substitute Recipient**, select a single default user to which emails should be sent instead of the primary recipient (such as request assignees, request collaborators, and request approvers). This enables one user to receive all notifications for testing purposes. Only users with a defined email address are displayed.

     > **Caution:**
     >
     > This setting is intended for use in test environments only. It should not be used in a production environment.

   - In **Attach Request File for Subscription Assignees**, click the check box to enable or disable automatically including the request file that was used to generate the request items in a subscription request as an attachment to the notification emails that get sent to subscription assignees and collaborators.
   - In **Attach Request Files for Approvers, Committers, and Notified Users**, click the check box to enable or disable automatically including a file attachment that contains all of the request items in a request to the notification emails that get sent to approvers, committers, and notified users.

     > **Caution:**
     >
     > The file attachment to the email will contain all of the request items in the request, regardless of the recipient's data permissions in Oracle Fusion Cloud Enterprise Data Management.

> **Note:**
>
> After you enable this system setting, view owners can enable or disable this setting for an individual view from the view inspector. See Inspecting a View.

3. Click **Save**.

Oracle Guided Learning Settings

Oracle Guided Learning (OGL) offers a robust framework for developing personalized, guided, and contextual training and user onboarding experiences. Use the **Oracle Guided Learning** settings to integrate an OGL application into your environment. For details, see Integrating EPM Cloud and EDM Cloud with Oracle Guided Learning in *Getting Started Guide for Administrators* .

Request Settings

Prevented Actions by Request Type

You can prevent users from performing workflow actions (such as reject, recall, or pushback) on requests of a specified request type. For example, you may not want users to have the ability to reject consolidation requests.

1. In **Prevented Action by Request Types**, click **Selected**.
2. In **Actions to Prevent**, select one or more workflow actions that you don't want users to perform for the selected request types:
   - Reject
   - Pushback
   - Recall
3. In **Select Request Types**, use the drop down box to select one or more request types that you want to prevent rejection for:
   - Interactive
   - Subscription
   - Consolidation
   - Import
   - Load

For the request types that you selected, users will no longer have the Reject action available to them.

Request Deletion

Use the **Prevent Deletion for Submitted Requests** checkbox to prevent users from deleting requests that have already been submitted. When this option is enabled, if a user attempts to delete a request in the Submit stage that has already been submitted (for example, a pushed back or recalled request), they are prompted to close the request instead. Closed requests cannot be resubmitted.

Request Purge

Request purge settings enable you to set the number of months after which request actions for completed requests are automatically purged from the system. This can reduce the size of your migration snapshot files.

- You can set the number of months after which request actions are purged from 6 to 18 months.
- After a request is purged, you can still view the request items (for example, in the Request Inspector or the Request Worklist), but you can no longer expand each request item to see the request actions for that item, and you can't download the request item details.
- The transactions for purged requests are still available in transaction history. See Auditing Transaction History.
- The purging of completed request actions takes place on a weekly basis, and after request actions are purged they cannot be recovered.

Calendar Settings

Calendar settings enable you to specify the end of the fiscal year for your organization. This setting is used by time labels to produce rolling time labeled viewpoints based on a Fiscal calendar rather than the Gregorian calendar. See Creating, Editing, and Deleting Time Labels.

Use the drop down menu to select the last month in your fiscal year.

Viewing the Record Count

You can view the record count, which represents a count of unique nodes across all applications for all users. The number of allowed records is set when you purchase your subscription and applies separately to your test and production environments. For example, if you have a subscription of 5000 records, you can use up to 5000 records on your production environment and up to 5000 records on your test environment.

> **Note:**
>
> You can purchase additional records from your sales representative.

When calculating node counts for licensing:

- Multiple instances of a node (shared nodes) are counted as one record.
- Nodes in specialty node type classes such as Lookup or User do not count towards the total record count in your environment.
- Nodes are counted as equivalent if the source and target nodes have the same base name after the qualifier is removed (default or alternate qualifier, prefix or suffix). See Working with Node Type Qualifiers.

For example, node "1000" exists in multiple node types, each with their own prefix or suffix qualifier, but all of the following are counted as a single node when determining your license count:

- Node type 1, default qualifier: "A_1000"
- Node type 2, default qualifier: "CC_1000"
- Node type 3, default qualifier: "1000-A"
- Node type 3, alternate qualifier: "1000-B"

To see the record count:

1. Click your user name in the top- right corner of the screen, and select **About**, then select **Subscription**.

2. If the total record count is greater than your subscription allows, contact your sales representative to purchase additional records.

> **Note:**
>
> The record count gets updated after the daily maintenance window or after the service is restarted. The date and time that the record count was last updated is displayed under the total nodes.

Service administrators can also view the Record Count Log report to see more details about the node counts in all applications, dimensions, and node types. See Record Count Log Report.

Embedding Enterprise Data Management Pages in Navigation Flows

You can embed Oracle Fusion Cloud Enterprise Data Management web pages within navigation flows in other Oracle Cloud EPM business processes.

A navigation flow card can point to a Enterprise Data Management URL such as the Views, Request Activity, or Applications page. A headerless option can be applied to the Enterprise Data Management URL to display the web page in an embedded mode within the Cloud EPM user interface.

For more information about navigation flows in your Cloud EPM business process, including how to create clusters and cards, see the *Viewing and Working with Navigation Flows* topic in your administering guide:

- Viewing and Working with Navigation Flows in *Administering Planning*
- Viewing and Working with Navigation Flows in *Administering Financial Consolidation and Close*
- Viewing and Working with Navigation Flows in *Administering and Working with Enterprise Profitability and Cost Management*
- Viewing and Working with Navigation Flows in *Administering Tax Reporting*

You embed Enterprise Data Management pages in a navigation flow by adding the URL for the page that you want to embed (for example, the Application, Requests, or Views list pages) in the **Content Source URL** field for the card in the navigation flow.

> **Tip:**
>
> You can navigate to the page in your Cloud EDM environment that you want to create the card for (for example, the Applications list) and then copy and paste the URL into the card.

To remove the Cloud EDM banner and tabs and make EDM headerless in the window, paste the following at the end of the URL:`&nochrome=true`. For example, to link to the Applications list without the EDM banner and tabs in the window, the URL would look similar to: `https://acme-epmidm.epm.us-somePlace-1.ocs.oraclecloud.com/epm/?root=workarea&workarea=applicationList&nochrome=true`

Note:

For details about the requirements for embedding URLs in navigation cards, see the *About Using URLs to Embed Third-Party Pages in EPM Cloud Applications* topic in your administering guide (for example, About Using URLs to Embed Third-Party Pages in EPM Cloud Applications in *Administering Planning*).

19
Working with Applications

For all application types except for Users applications, you use a Oracle Fusion Cloud Enterprise Data Management application to govern an external application's enterprise data. All work on data involves an application's viewpoints and related data objects.

There are different application types, see Understanding Application Types. The principles and features of applications that are described in this topic may not all apply to Users applications. For details, see Working with Users Applications.

> **Note:**
>
> - An *external application* is a system or application containing enterprise data that you need to work with. For example, Planning is an external application.
> - An external application can be any data source containing enterprise data. For example, you can use database tables, spreadsheets, and text files as external applications.

Videos

Your Goal	Watch This Video
Learn about working with applications.	▶ Overview: Understanding Applications in Enterprise Data Management Cloud

The following list describes the main functions of an application:

- Ensures that data conforms to the external application's requirements. For example, an application specifies which properties are available and whether they are required.
- Presents data to users as per your requirements. For example, you can configure an application to make subsets of a dimension's nodes available to different users.
- Imports enterprise data from an external application using a comma-delimited file. Planning enterprise data can also be imported from an external application by using a connection.
- Exports enterprise data to an external application using a comma-delimited file. Planning enterprise data can also exported to an external application inbox by using a connection.

Note:

All application types support importing and exporting from comma-delimited files.

For Planning applications, if you enter valid connection information to an external application during registration, you can import and export from the external application inbox by using a connection. You can then import that file into your Planning application.

For Oracle Financials Cloud General Ledger applications, if you enter valid connection information to an external application during registration, you can export to the external application server by using a connection. You can then import that file into your Oracle Financials Cloud General Ledger application.

You can register applications for various application types. For example, there is an application type for Planning, see Understanding Application Types.

When you register an application, the registration process configures settings, dimensions, and properties, and creates data objects used to work with data, see Understanding Registering Applications.

After you have registered the application, you can perform the tasks listed in the following table.

Table 19-1 Application Tasks

Task	More Information
Modify a registered application's settings. For example, you can add a dimension to an application.	Understanding Modifying Applications
Import data from an external application.	Importing Dimensions
Export data to an external application.	Exporting Dimensions
Review an application's settings. For example, you can examine a dimension's settings.	Inspecting Applications

Note:

If you are working with enterprise data for multiple external applications, you register a Cloud EDM application for each external application, see Understanding Applications and Sharing Data.

Understanding Application Types

Oracle Fusion Cloud Enterprise Data Management provides application types for commonly-used external applications. These application types enforce conformity to external applications while requiring minimal configuration. For example, the Planning application type automatically defines the properties required by dimensions.

Cloud EDM also provides a Universal application type, which can be defined and configured in any way you would like, and a Users application type, which can be used to manage a set of users who can be associated with nodes in other applications.

The following table lists the available application types.

Application Type	More Information	Considerations
Planning applications	Working with Planning and FreeForm Applications	
FreeForm applications	Working with Planning and FreeForm Applications	
Planning Modules applications	Working with Planning and FreeForm Applications	Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM. See Planning Modules Predefined Members.
Financial Consolidation and Close applications	Working with Financial Consolidation and Close Applications	Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM. See Financial Consolidation and Close Predefined Members.
Enterprise Profitability and Cost Management applications	Working with Enterprise Profitability and Cost Management Applications	Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM. See Enterprise Profitability and Cost Management Predefined Members.
Tax Reporting applications	Working with Tax Reporting Applications	Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM. See Tax Reporting Predefined Members.

Application Type	More Information	Considerations
Oracle Financials Cloud Oracle General Ledger applications	Working with Oracle Financials Cloud General Ledger Applications	Do not use inherited property values in any of the segment value properties, and avoid using Derived properties that contain positional logic (such as `ancestors`, `parent`, `children`, and `bottom`) when creating expressions to derive their default values. If your expression uses positional logic, used Derived and Stored properties instead. See Predefined Properties for Oracle Financials Cloud General Ledger Applications.
Oracle E-Business Suite General Ledger applications	Working with E-Business Suite General Ledger Applications	Do not use inherited property values in any of the segment value properties, and avoid using Derived properties that contain positional logic (such as `ancestors`, `parent`, `children`, and `bottom`) when creating expressions to derive their default values. If your expression uses positional logic, used Derived and Stored properties instead. See Predefined Properties for E-Business Suite General Ledger Applications.
Universal applications	Working with Universal Applications	
Users	Working with Users Applications	

Understanding Registering Applications

An application's registration defines the application's dimensions as well as the dimensions' properties.

Application registration also configures any other settings required by the application type. For example, when you register a Planning application, you specify the application's cubes.

Videos

Your Goal	Watch This Video
Learn about working with applications.	▶ Overview: Understanding Applications in Enterprise Data Management Cloud

For step by step instructions, see:

- Registering Planning and FreeForm Applications

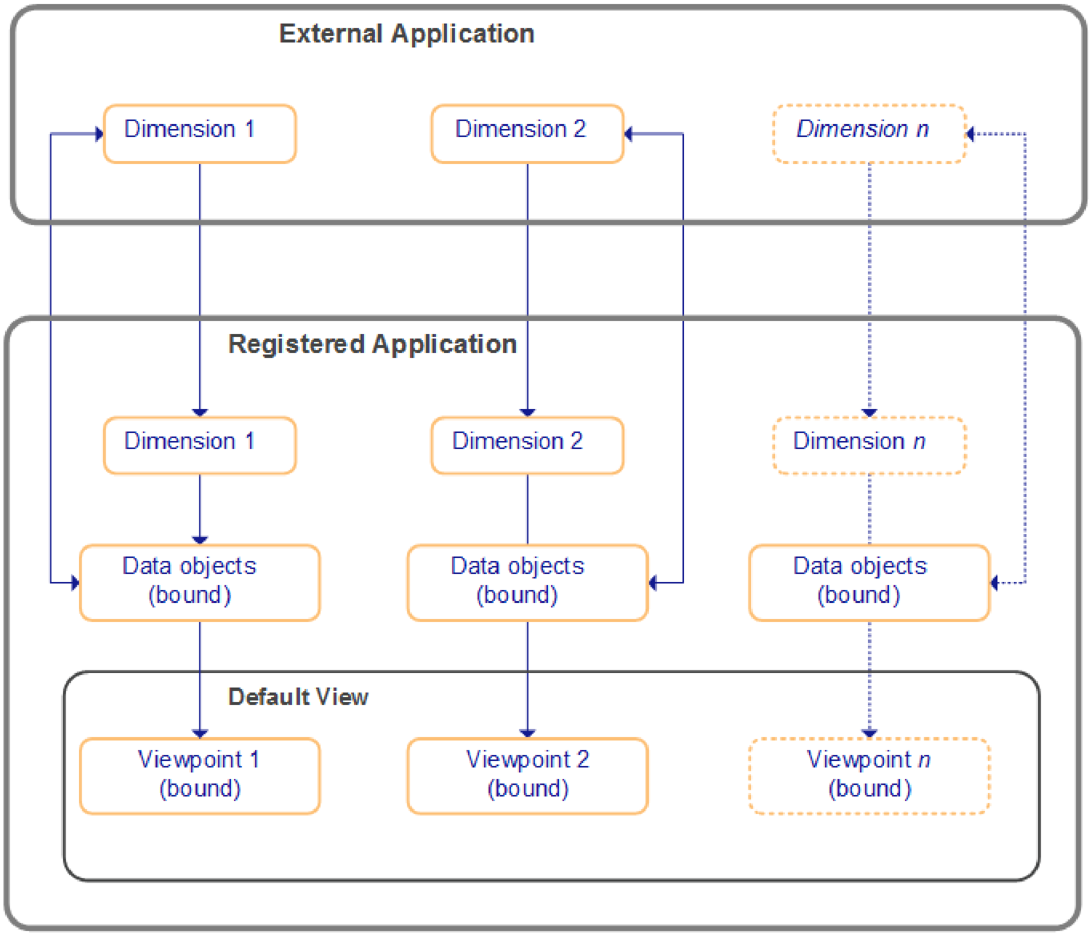
- Registering Financial Consolidation and Close Applications
- Registering Enterprise Profitability and Cost Management Applications
- Registering Tax Reporting Applications
- Registering Oracle Financials Cloud General Ledger Applications
- Registering E-Business Suite General Ledger Applications
- Registering a Universal Application
- Registering Users Applications

The registration process creates the following items:

- The application's default view
- A viewpoint and related data objects for each registered dimension

The viewpoints and data objects created by the registration process are bound to dimensions. When you import data, the data is contained by the viewpoint bound to the dimension. When you export, the bound viewpoint's data is exported, see Understanding Bindings and Bound Data Objects.

The following diagram illustrates the relationship of an external application's dimensions or segments to Oracle Fusion Cloud Enterprise Data Management dimensions:



Each of the external application's dimensions is bound to a dimension in the registered application.

> **Note:**
>
> You can create unbound data objects.

Permissions and Registering Applications

To register an application, you must be assigned the Application - Create role. When you register an application, the registration process assigns you to the following permissions:

- The *Owner* permission on the application's default view
- The *Owner* permission on the application

For more information, see:

- Understanding Dimensions
- Understanding Modifying Applications

Understanding Modifying Applications

You can modify the dimensions and other settings of registered applications. The modifications you can make depend upon the application type.

You can add, remove, or update dimensions or application settings. For example, in Planning applications you can add or remove cubes. To modify specific applications, see:

- Modifying Registered Planning and FreeForm Applications
- Modifying Registered Financial Consolidation and Close Applications
- Modifying Registered Enterprise Profitability and Cost Management Applications
- Modifying Registered Tax Reporting Applications
- Modifying Registered Oracle Financials Cloud General Ledger Applications
- Modifying Registered E-Business Suite General Ledger Applications
- Modifying a Universal Application

> **Note:**
>
> Because Users applications do not use a registration wizard, you modify dimension and application settings by inspecting the application.

When you remove a dimension or property from a registered application, the removed item's data is no longer included in exports and imports. The following table describes the considerations.

Removed Item	Description
Dimension	The dimension's bound viewpoint and data objects are unbound. The dimension and its data chain objects remain. The dimension's status is set to *Archived*.
Property in a Universal application	The property is not removed from its node type.

> **Tip:**
>
> If you need to download the data for a removed dimension or property, download the dimension's viewpoint, see Downloading a Viewpoint.

When you remove a dimension from a registered application, that dimension's status is set to *Archived*. If you add a dimension in application registration with the same name as an archived dimension, that dimension will be unarchived and set to *Active*.

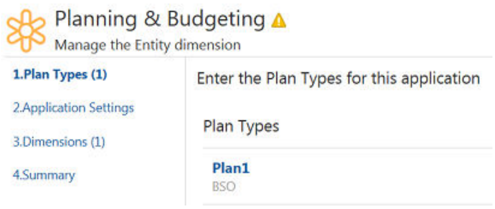
> **Note:**
>
> You cannot add a dimension in application registration that has the same name as an unbound dimension, even if that dimension is archived.

You can also archive and unarchive a dimension from the dimension inspector. See Archiving and Unarchiving Dimensions.

Modifying an Application

To modify an application, take the following steps:

1. Click **Applications**.
2. In the application's **Actions** column, click ⋮, and then select **Inspect**.
   The items that can be modified depend upon the application type. The following example indicates that for Planning applications you can modify cubes, application settings, and dimensions:



Planning & Budgeting ⚠
Manage the Entity dimension

1.Plan Types (1)
2.Application Settings
3.Dimensions (1)
4.Summary

Enter the Plan Types for this application

Plan Types

Plan1
BSO

Understanding Applications and Sharing Data

Typically you use Oracle Fusion Cloud Enterprise Data Management to govern enterprise data for multiple external applications. For example, an organization could use Cloud EDM to manage entity nodes used in general ledger, consolidation, and Planning external applications.

Videos

Your Goal	Watch This Video
Learn about sharing data across applications.	▶ Sharing Data Across Applications

To build applications, users can manage external applications' data using these steps:

1. For each external application, register a Cloud EDM application.
2. Create a new view.
3. In the view, create one viewpoint for each application dimension containing data to be shared. Each viewpoint must use the node set bound to the application's entity dimension.

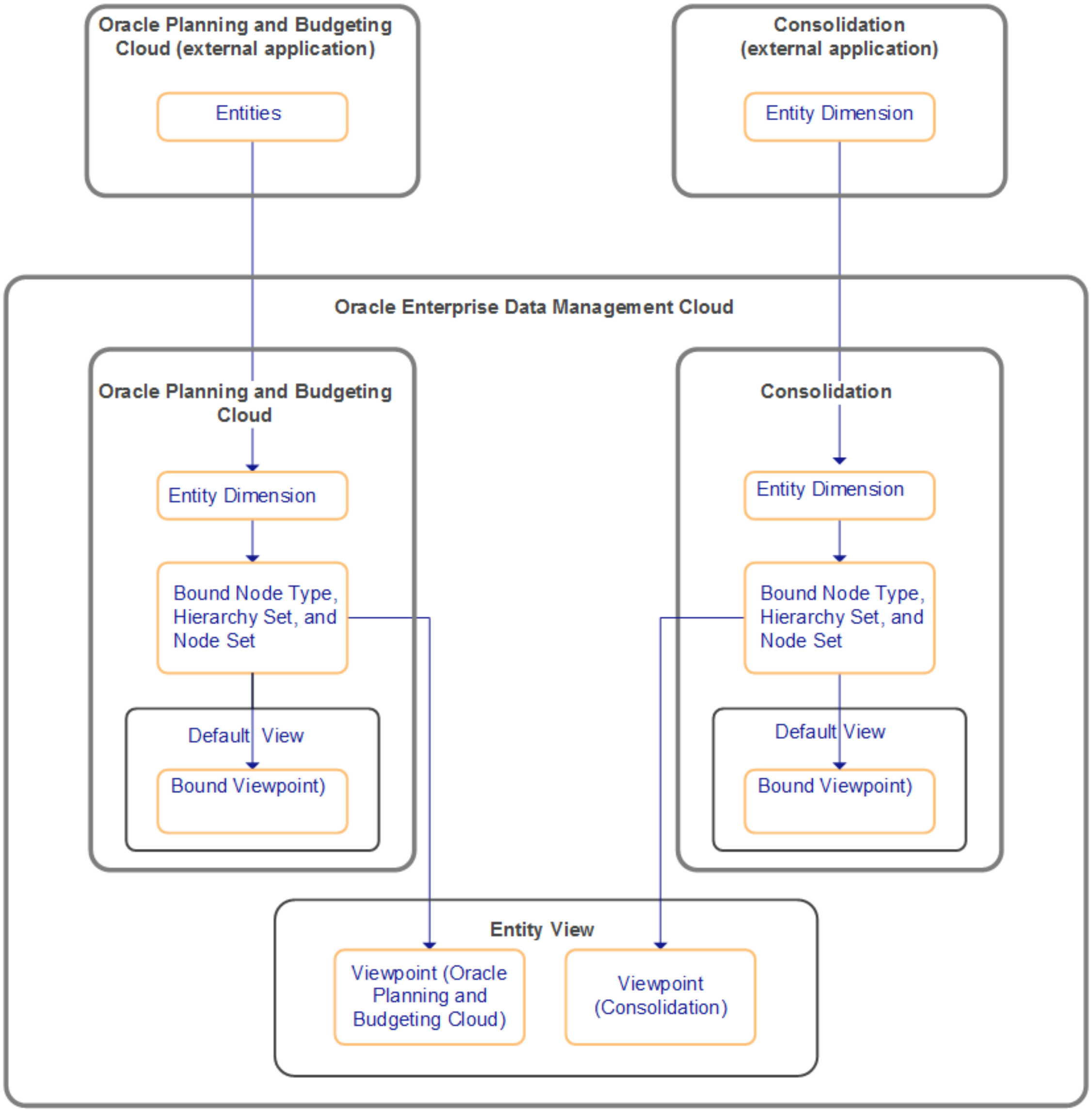
Users then use the viewpoints to manage the enterprise data. For example, a user could compare two applications' viewpoints and then copy nodes between the viewpoints. Changes can then be exported into the external applications.

> **Note:**
>
> Node type converters are needed to drag and drop, compare, and locate common nodes between viewpoints that use different node types, see Working with Node Type Converters.

Two Applications that Share Data

Let's say you need to rationalize nodes for entity dimensions contained by consolidation and Planning external applications. You register two external applications, defining an entity dimension for each. The registration process creates a default view, a bound viewpoint, and related bound data objects.

As shown in the following diagram, to enable sharing between applications, you create a view with two viewpoints; in this example the new view is named **Entity View**. Each viewpoint uses its entity dimension's bound node set.


Oracle Planning and Budgeting Cloud (external application)
Entities
Consolidation (external application)
Entity Dimension
Oracle Enterprise Data Management Cloud
Oracle Planning and Budgeting Cloud
Entity Dimension
Bound Node Type, Hierarchy Set, and Node Set
Default View
Bound Viewpoint)
Consolidation
Entity Dimension
Bound Node Type, Hierarchy Set, and Node Set
Default View
Bound Viewpoint)
Entity View
Viewpoint (Oracle Planning and Budgeting Cloud)
Viewpoint (Consolidation)

Users can use the Entity View to manage the dimensions' data; for example, users can copy nodes between the viewpoints. Since the Entity View's viewpoints use the bound node sets, any changes made in the new viewpoints are reflected in the bound viewpoints and thus can be exported.

For an example, see Comparing Enterprise Data Across Applications.

Understanding Blockout Periods

Blockout periods can be set up on applications or dimensions to enable *Owners* or *Metadata Managers* to establish a date range during which requests, imports, and viewpoint loads are held and not applied to the active view until the blockout period has ended.

For example, you may not want changes to be made to data during a financial close, a system reconciliation, or a data migration.

During the blockout period, requests can still be created and submitted, and the requests continue to follow the approval and commit policy workflow. However, after all of the approval and commit policies are fulfilled, the request is held in a "blocked" state until the blockout period is over. When the blockout period ends (either because the blockout period end date was reached or because an owner disabled the blockout period manually), all blocked requests are validated and either completed and closed if there are no issues or returned to the workflow if there are validation issues.

Considerations

- You can configure blockout periods on applications and dimensions. By default, the dimension blockout settings are the same as the application blockout settings, but you can override the application settings with dimension blockout settings. See Application and Dimension Blockout Settings.
- You configure the blockout period from the application or dimension inspector. You can set the blockout for a specified time period only or a recurring time period. For recurring time periods, you can base the period on a Gregorian period or a fiscal period. See Configuring Blockout Periods.
- By default, the blockout applies to request items for all bound or partially bound data chain objects. You can configure the blockout period to apply to unbound data chain objects as well.
- You can specify that requests submitted by users with *Owner* or *Data Manager* permission on the application or dimension are exempt from the blockout period and are completed normally.
- For requests that contain request items from different applications or dimensions, if any application or dimension in the request is currently in a blockout period the entire request is blocked. All changes from all request items are applied after the blockout period has ended for all associated applications or dimensions.

While the blockout period is active:

- You cannot perform any workflow actions on blocked requests (for example, recall, withdraw approval, or push back). However, you can still add comments or attachments to the request.
- You cannot run imports or viewpoint loads on any blocked dimensions, unless the **Allow Exceptions** setting on the affecting blockout (application or dimension) allows you to.
- You cannot bind a different viewpoint to a dimension in the blocked application.

When the blockout period is over:

- Requests are validated. If there are no errors, the requests are completed. Multiple blocked requests for the same application are completed in the order that their workflow ended and they entered Blocked status.
- If there are validation errors, the request status is changed from Blocked to its previous stage before the blockout (Draft or In Flight), and the submitter is notified.

> **Note:**
>
> When a blockout period has ended (because the blockout period end date was reached or because an owner disabled the blockout period manually), it may take up to five minutes for the blockout status to be updated and any blocked requests to be processed.

Configuring Blockout Periods

You configure blockout periods from the application or dimension inspector.

1. Inspect the application or dimension that you want to configure the blockout period for. See Inspecting Applications.
2. On the General tab, click **Edit**.
3. Perform an action:
   - For an application: Click **Enable Blockout**.
   - For a dimension: Select an option:
     - **Same as Application**: The dimension uses the same settings as the application blockout.
     - **Enabled**: The dimension overrides the application blockout settings with the settings that you specify.
     - **Disabled**: The dimension overrides the application blockout settings and disables the blockout for this dimension.
4. Select the blockout type:
   - **Specified**: Apply blockout settings for the specified time period only.
   - **Recurring**: Apply blockout settings on a recurring basis, based on the period that you specify.
5. Select the time period for the blockout, based on the blockout type:
   - **Specified**: Select the time zone for the specified blockout period, and then select the begin and end dates and times for the blockout period. You can use the date and time picker or edit the entries manually.
   - **Recurring**: Select the following:
     - **Period**: Select the recurring period from the drop down menu. You can select a Gregorian period (for example, **End of Quarter**) or a fiscal period (for example, **Fiscal Year End**).

       > **Tip:**
       >
       > You can configure your fiscal year in the system settings. See Calendar Settings.

     - **Time of Day**: Enter the time of day that the recurring blockout will begin.
     - **Time Zone**: Select the time zone for the Time of Day setting.

       > **Note:**
       >
       > The same **Time of Day** and **Time Zone** settings are used for both the begin and end dates. You cannot select different times of the day for the beginning and end of a recurring blockout.

     - **Begin**: Enter the number of days from the period date that the blockout will begin. You can enter a negative number to begin the blockout before the period date or a positive number to begin the blockout after the period date. For example, if you enter **-5** and your Period is **Fiscal Quarter End**, then the recurring blockout will occur 5 days before each fiscal quarter ends.

  - **End**: Enter the number of days from the period date that the blockout will end. You can enter a negative number to end the blockout before the period date or a positive number to end the blockout after the period date. For example, if you enter **1** and your Period is **End of Month**, then the recurring blockout will end the day after the end of each month.

6. **Optional**: In **Allow Exceptions**, click the drop down to specify that *Owners* or *Data Managers* can submit requests that are not affected by the blockout period.

   > **Note:**
   >
   > Changing this setting does not affect requests that are already in Blocked status. Setting this to *Data Managers* will include application owners as well. See Inclusive and Additive Permissions.

7. **Optional**: Select **Block Unbound Objects** to include request items for data objects in fully unbound viewpoints in the blockout period.

   > **Note:**
   >
   > Data objects from bound or partially bound viewpoints are always included in the blockout period.

8. Click **Save**.

Application and Dimension Blockout Settings

When blockout periods are configured at both the application and dimension level, the dimension blockout settings generally override the application settings, as follows:

- **Enabled** and **Disabled**:
  - If a blockout is enabled at the application level but disabled at the dimension level, request items from that dimension are processed.
  - If a blockout is disabled at the application level but enabled at the dimension level, request items from that dimension are blocked, while request items from other dimensions are processed.

  > **Note:**
  >
  > For requests that contain request items from different applications or dimensions, if any application or dimension in the request is currently in a blockout period the entire request is blocked.

- **Block Unbound Objects**: If both the dimension and application for an unbound object are in blockout, the Block Unbound Objects setting from the dimension is used.
- **Allow Exceptions**: If both the application and dimension are in blockout, the Allow Exceptions setting from the dimension is used. When multiple applications or dimensions are in blockout, the strictest settings for Allow Exceptions are used.

- **End Date**: If both a dimension and its application are in blockout, the End Date of the dimension blockout is used. For a single request with multiple viewpoints that have different blockout end dates, the end date is the furthest effective End Date in the future.

Disabling Workflow for Applications

You can disable workflow at the application level so that requests made in that application will bypass all active policies when the request is submitted.

This can be helpful, for example, when you want to change data in your test environment without having to go through the workflow process.

Considerations

- Disabling workflow for an application does not disable the individual policies for that application. When you resume workflow, all enabled policies go back into effect. To disable an individual policy, see Modifying Policies.
- The Disable Workflow setting does not affect existing requests which have already been submitted. Only newly submitted requests will bypass active policies.
- This setting does not get transferred when you transfer an application using a template. That is, if you disable workflow for an application in one environment and then transfer that application to another environment, workflow is not disabled for that application in the second environment.

To disable workflow for an application:

1. Inspect the application. See Inspecting Applications.
2. On the **General** tab, select **Disable Workflow** to have requests made in the application bypass all active workflow policies.
3. Optionally, use the **Submitted By** parameter to limit the bypassed workflow to requests submitted by the selected users only.

Inspecting Applications

After you register an application you can inspect it and edit settings.

From **Applications**, scroll to your application, click ⋮, click **Inspect**, and then select from the options on these tabs:

- **General**: Review the following general application information, or click **Edit** to change it:
  – **Summary**:
    * Name
    * Description
    * Status: **Draft**, **Active**, or **Archived**. See Archiving, Unarchiving, and Deleting Applications.

> **Note:**
>
> If an application is in Draft status, you cannot make changes to the status in the inspector. After you register the application, the status will change from Draft to Active.

    - Application Type (cannot be changed)
    - Default View (cannot be changed)
  - **Blockout Period**: Enable or disable a blockout period for an application. See Configuring Blockout Periods
  - **External System Type** (Universal application types only): Click the drop down menu to select the type of external system for your Universal application.
  - **Requests**: Select **Disable Workflow** to have requests made in the application bypass all active workflow policies. See Disabling Workflow for Applications.
  - **History**: Displays username and date information for when the application was created and last modified.
- **Dimensions**: Click a dimension, then change bindings or mapping keys, see Inspecting Dimensions.

> **Note:**
>
> By default, archived dimensions are not displayed. To display archived dimensions, click ▼, and then select **Show Archived**. You can select this option again to display active dimensions only.

- **Connections**: Edit the external application connection settings. Whether you make changes or not, you can test the connection to determine if it is correctly configured. (Not available for Universal or Users applications)

> **Note:**
>
> The **Connections** tab is available only if your application type supports connections and you have at least *Data Manager* or *Metadata Manager* permission on the application.

- **Extract Packages**: Create and edit extract packages for the applications. See Working with Extract Packages.
- **Registration**: View the registration settings. (Not available for Users applications)

> **Note:**
>
> To edit the registration settings: from **Applications**, scroll to your application, click ⋮, and then click **Modify**.

- **Constraints**: Create, edit, and view application constraints, see Working with Constraints.

- **Validations**: Create, edit, and view custom validations for the application, see Custom Validations.
- **Permissions**: Change the application security permissions, see Assigning Application Permissions.
- **Policies**: Configure an approval policy for the application, see Configuring Policies.

Assigning Application Permissions

You must have the *Owner* or *Metadata Manager* permission on a specific application to assign or remove permissions for other groups or users. Application permissions are assigned to one application at a time. If you inadvertently remove application permissions for yourself, contact a user with the *Owner* or *Metadata Manager* permission or a Service Administrator to re-assign your permissions.

> **Note:**
>
> After you register an application you are given the *Owner* permission on the application.

To assign application permissions:

1. From **Applications**, open an application.
2. Select **Permissions**, and then click **Edit**.
3. Select a user or group.
4. Select the permission to assign, click **Save**, and then click **Close**.

See Working with Permissions.

Searching for Applications

When you perform a search in the Applications list, the system searches for your text in application names and descriptions. The names and descriptions that contain the search text are displayed in the results window in bold-faced font. You can also include archived applications in your search.

To search in the Applications list:

1. From **Applications**, click 🔍.
2. (Optional) If you want your search to include archived applications, click ▼ and then select **Show Archived**.
3. Enter a text string to search for. The search is performed automatically after you stop typing.

   To close the search, click ✕.

Copying Applications

You can copy an application to create a new application of the same type with the same configuration as the original.

When you copy an application, all metadata within the original application is copied to the new application including dimensions, data chains, validations, policies, and subscriptions.

> **Note:**
>
> Application data is not copied. Use the export and import processes to transfer bound data and properties to the new application, and the extract and load processes to transfer unbound or partially bound data and properties.

Considerations

- You must have the *Application - Create* application role to copy an application. See Working with Roles and Permissions.
- For application views:
  - A new default application view is created for the new application.
  - For other views, if you have the *Owner* permission on a view in the application being copied, the view and its viewpoints are copied to the new application, as follows:
    - If all of the viewpoints in a view belong to the application being copied, then a copy of that view and all of the viewpoints is created.
    - If some but not all of the viewpoints in a view belong to the application being copied (for example, in a maintenance view), then the viewpoints for the copied application are copied and added to the existing view.
- The following objects are *not* copied to the new application:
  - Connection login details
  - Time labels and time labeled viewpoints
  - Compare profiles
  - Application specific permissions for global connections
- Both active and archived metadata objects are copied.

To copy an application:

1. From the application list, click **Actions** ⋮, and then select **Copy**.
2. In the Copy Application dialog box, enter a name and (optionally) a description for the new application. The new name must be unique.
3. **Optional**: In **Duplicate Suffix**, enter a suffix to be used to create a unique label when creating objects for the new application (such as viewpoints). If you do not enter a suffix, an incrementing numeric suffix is used.
4. On the confirmation dialog box, click **Yes**.

Archiving, Unarchiving, and Deleting Applications

You can archive an active application when you no longer need it. For example, you may want to archive an application at the end of a period of time such as the end of the quarter or end of the year. You can also return an archived application to Active status or you can delete the application.

Considerations

- A user who has the *Owner* permission on the application can change its status or delete it.
- When you archive an application, all of the application's dimensions and their data chain objects (node types, hierarchy sets, and node sets) are also archived. The default view remains active.
- When you return an archived application to Active status, all of the application's dimensions and their data chain objects are restored to the status that they had before the application was archived.
- An application must first be archived before you can delete it.
- Deleting an application permanently deletes all of its data including data chain objects, requests, transactions, match results, and the registered data source.

> **Caution:**
>
> This operation cannot be undone. You may want to make a migration back up before deleting an application in case you need to restore it in the future. See Backing up and Restoring Cloud EDM

- You cannot delete a Users application if one of its node sets is referenced by a Node data type property in another application.

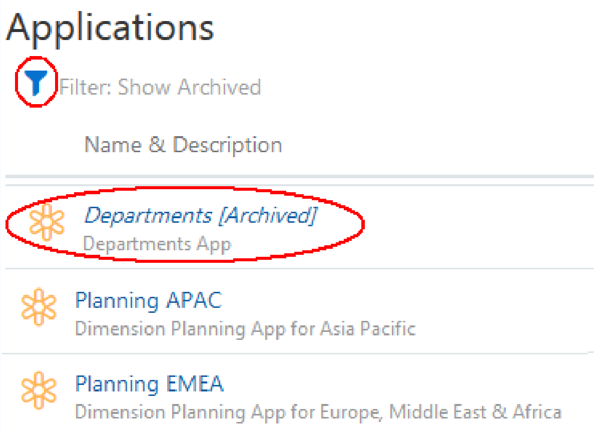
For more information, see Working with Data Objects and the Data Chain.

Archiving

To archive an application:

1. From **Applications**, inspect the application that you want to archive.
2. From **General**, click **Edit**.
3. From **Status**, select **Archived**, and then click **Save**.

The application is no longer displayed in the Applications list.

> **Note:**
>
> To view archived applications in the Applications list, click ▼.
>
> 
>
> Applications
>
> ▼ Filter: Show Archived
>
> Name & Description
>
> *Departments [Archived]*
> Departments App
>
> Planning APAC
> Dimension Planning App for Asia Pacific
>
> Planning EMEA
> Dimension Planning App for Europe, Middle East & Africa

Unarchiving

To return an archived application to Active status:

1. From **Applications**, click ▼ to show archived applications.
2. Inspect the application that you want to return to Active status.
3. From **General**, click **Edit**.
4. From **Status**, select **Active**, and then click **Save**.

The application is displayed in the Applications list.

Deleting

To delete an archived application :

1. From **Applications**, click ▼ to show archived applications.
2. In the **Actions** column for the application, click ⋮, and then select **Delete**.
3. Click **Yes** to confirm the deletion.

The application is removed from the Applications list.

Refreshing Application Indexes

When you make one of the following changes to a property's Default Type or Inheritance parameters (see Editing Property Parameters), the index for an application using that property may need to be refreshed in order for viewpoint queries to properly use the index.

- Default Type is changed from Specified to None
- Default Type is changed from Derived to None

- Inheritance is changed from Positional to None

There are two ways that an application index gets refreshed:

- Automatically, on a weekly basis
- Manually, by performing an application index refresh

To refresh an application index manually, from **Applications** find your application, click ⋮, and select **Refresh Index**. You can rebuild indexes for active applications only.

> **Note:**
>
> You must be a Service Administrator or have *Owner* or *Metadata Manager* permission on an application in order to rebuild its index.

Understanding Dimensions

A registered application contains one or more dimensions. A registered dimension represents a dimension, segment values, or other data domain in an external application.

You add dimensions when you register the application. After an application has been registered, you can add a dimension or update a dimension's setting by modifying the registered application.

A dimension is bound to a viewpoint. When you import data, the data is contained by the bound viewpoint. When you export, the bound viewpoint's data is exported, see Understanding Bindings and Bound Data Objects.

> **Note:**
>
> - Some settings in dimensions apply only to their application types. For example, the Planning application type requires you to specify at least one cube for a dimension.
> - You can create a dimension in a registered application that does not yet exist in its external application. This enables you to model a new dimension.

For more information, see:

- Understanding Registering Applications
- Understanding Data Chains

Inspecting Dimensions

- **General**: View and edit the dimension's name, description, and status. Enable or disable a blockout period for the dimension. See Configuring Blockout Periods.
- **Bindings**: (Bound dimensions only)
  - To add a mapping binding, click **Create Map Binding**, see Creating a Map Binding.
  - To change a binding, click the binding and update it, see:

  - Binding a Viewpoint to a Dimension in Planning and FreeForm Applications
  - Copying Bindings to Create Trees or Tree Versions
  - Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications
- **Import/Export**: (Bound dimensions only). Manage the import and export options for the dimension, see Managing Import Options and Managing Export Options.
- **Extracts**: Manage extracts for the dimension, see Working With Extracts.
- **Loads**: Create, edit, and view loads for the dimension, see Working with Viewpoint Loads.
- **Validations**: Manage the enforcement of predefined validations, see Predefined Validations.
- **Permissions**: Set up user and group permissions for the dimension, see Working with Permissions.
- **Policies**: Configure an approval policy for the dimension, see Configuring Policies.

Working with Unbound Dimensions

Unbound dimensions are created outside of the application registration process and do not get imported or exported. You can use unbound dimensions to manage complex data sets such as related value sets, account combinations, and multi-dimensional mappings in Oracle Fusion Cloud Enterprise Data Management.

You can load data into unbound dimension viewpoints by using request load files (see Making Changes Using a Load File) and extract data using extracts (see Working With Extracts).

Considerations

- You must be a Service Administrator or have *Owner* or *Metadata Manager* permission on an application in order to create an unbound dimension.
- Unbound dimensions are not displayed in the application registration wizard or the import and export screens. You can define extracts for unbound dimensions.
- Unbound dimensions cannot contain bound or partially bound data chains.
- You cannot convert an unbound dimension to a bound dimension.
- Unbound dimensions cannot have the same name as bound dimensions within an application. If you modify an application registration to add a bound dimension, you cannot use the names of any unbound dimensions for that application during the registration process.
- When you create an unbound dimension in a Universal application that is in Draft status, the application status is changed to Active.

To create an unbound dimension:

1. Inspect the application that you want to create the unbound dimension in.
2. On the Dimensions tab, click **Create**.
3. From **Type**, select Unbound.
4. Enter a name and (optionally) a description for the dimension, and then click **Save**.

> **Note:**
>
> The dimension cannot have the same name as any other bound, unbound, or lookup dimensions in the application.

The dimension is displayed in the dimension inspector.

After you have created an unbound dimension, you can use it just like a bound dimension when creating viewpoints and other user defined data chain objects and properties. You can also create extracts for the dimension, and add permissions, policies, and custom validations. However, you cannot create bindings for an unbound dimension, or use it for importing and exporting data.

Working with Lookup Dimensions

Lookup dimensions allow reference data sourced from an external system to be associated to data being managed in Oracle Fusion Cloud Enterprise Data Management. Nodes in lookup dimensions do not get included in the record count for the environment.

For example, suppose you had a list of cost center owners that you want to associate with an existing Cost Center dimension. You could create a lookup dimension for the cost center owners in Cloud EDM and use a node data type property to assign the owners to the cost centers in your Cost Center dimension. As a lookup dimension, you could maintain the list of owners in a structure (for example, by line of business or department), and the nodes in the lookup dimension would not count towards your total record count.

Considerations

- You must be a Service Administrator or have *Owner* or *Metadata Manager* permission on an application in order to create a lookup dimension. You can add lookup dimensions to any application type.
- Lookup dimensions use the Lookup node type class, which has a predefined set of properties that you cannot add to or remove from. You can reorder the properties, and you can set them are Required or not in the node type inspector. See Adding, Removing, and Configuring a Node Type's Properties. Nodes in a lookup dimension can use any of the predefined properties. See Predefined Properties for Lookup Node Types.
- You can add multiple bindings to a lookup dimension. The bindings can have a node or hierarchy type. See Creating a Lookup Binding.
- Lookup dimensions can be used in existing views, or you can create a view to use for the lookup dimension when you create the dimension.
- You cannot import, export, or extract data from lookup dimensions. You also cannot download a lookup dimension viewpoint to a file. You can load data into lookup dimension viewpoints by using request load files (see Making Changes Using a Load File) or viewpoint loads (see Working with Viewpoint Loads).
- Lookup dimensions are not displayed in the application registration wizard.
- When you create a lookup dimension in a Universal application that is in Draft status, the application status is changed to Active.

To create a lookup dimension:

1. Inspect the application that you want to create the lookup dimension in.
2. On the Dimensions tab, click **Create**.

3. From **Type**, select Lookup.
4. Enter a name and (optionally) a description for the dimension.

> **Note:**
>
> The dimension cannot have the same name as any other bound, unbound, or lookup dimensions in the application.

5. **Optional**: In **View**, enter a name to create a view to use for the lookup dimension viewpoints.

> **Note:**
>
> If you are not a Service Administrator, you must have the *Views - Create* role to create the view.

6. Click **Save**.
   The dimension is displayed in the dimension inspector.

After you have created a lookup dimension you can create lookup bindings for it, which create associated data chain objects if needed (see Creating a Lookup Binding), or you can create the data chain objects manually (as in an unbound dimension). Then, you can use the lookup dimension, modify data with requests, create extracts for the dimension, and add permissions, policies, and custom validations.

> **Note:**
>
> You cannot add or remove properties from the dimension, and you cannot use it for importing and exporting data

Archiving and Unarchiving Dimensions

You can archive a dimension when you no longer need it. For example, if you want to change the viewpoint bound to a dimension, you may want to archive the original dimension so that you have its history. You can also return an archived dimension to Active status.

Considerations

- You must be a Service Administrator or have *Owner* permission on the application or dimension to change a dimension's status.
- You cannot archive a dimension if there are In Flight requests for that dimension. All requests must be completed for a dimension before it can be archived.
- When you archive a dimension:
  - The dimension, binding, and associated data chain objects are set to Archived.
  - By default, the dimension is not listed on the Dimensions tab in the Application inspector. To display archived dimensions, click ▽, and select **Show Archived**.
  - The dimension is not displayed in application registration.

  - The dimension is not displayed in the Import, Export, or Extract screens for the application.

- When you return an archived dimension to Active status, all of the dimension's data chain objects and the binding are restored to the status that they had before the dimension was archived.
- When you archive a dimension, subscriptions that use that dimension are automatically ignored and will not generate subscription requests. When you unarchive a dimension, subscriptions are no longer ignored and are returned to their previous state (Enabled or Disabled).

For more information, see Working with Data Objects and the Data Chain.

To archive or unarchive a dimension:

1. Inspect the dimension. See Inspecting Dimensions.
2. On the General tab, click **Edit**.
3. In the **Status** field, select `Archived` to archive a dimension. Select `Active` to unarchive a dimension.

> **Note:**
>
> You can also archive or unarchive a dimension by modifying the application registration and adding or removing that dimension. See Understanding Modifying Applications.

Understanding Attributes

Attributes are properties on nodes in dimensions that describe the data in that dimension by using either an allowed value list or a node data type property that references another dimension.

The dimension being referenced by the node data type property can be an attribute dimension created specifically for this purpose or an existing registered dimension.

Your Goal	Watch This Video
Learn about working with attribute dimensions.	▶ Attribute Dimensions in Oracle Fusion Cloud Enterprise Data Management

> **Note:**
>
> Attribute dimensions are supported for all sparse dimensions in Planning, Planning Modules, Financial Consolidation and Close, and Enterprise Profitability and Cost Management applications. This includes the Account dimension (if it is sparse) in Financial Consolidation and Close applications.

These types of attributes are supported in Cloud EDM:

- **Simple Attribute**: a string data type property with an allowed value list that contains all of the selectable values for that attribute. This is best used for attributes with fewer values that don't change much over time. For example, a Product dimension can have a simple attribute called Color, which can contain values for Green, Blue, Yellow, and Red.
- **Attribute Dimension**: a combination of a full dimension that contains the attribute values and a node data type property that references that attribute dimension from the base dimension. The attribute dimension values can be one of these types:
  - Text
  - Numeric
  - Date

  Attribute dimensions are best used for attributes with a large number of values that you want to maintain in Cloud EDM. For example, an Entity dimension can have an attribute dimension called Region, which can contain a geographical hierarchy with parent and child members.
- **Registered Dimension**: a combination of an existing registered dimension that contains the attribute values and a Node data type property that references that dimension from the base dimension.

Both simple attributes and attribute dimensions are created during application registration. See Registering Cubes, Application Settings, and Dimensions.

Working with Simple Attributes

You create simple attributes by entering an attribute name during application registration and then entering all of the values in the allowed values list for that attribute property. To modify the list of allowed values, edit the property that was created for the simple attribute during registration. See Editing Property Parameters.

Working with Attribute Dimensions

When you register an attribute dimension during application registration, a full dimension for that attribute dimension is created with its own data chain, binding, and viewpoint. The viewpoint for the attribute dimension is added to the default view for the application and can be used to manage the nodes in the attribute dimension. A node data type property is also created that you can use to reference the attribute dimension from a base dimension.

Considerations

- Attribute dimensions of type Numeric and Date have format validations applied on bottom nodes. That is, all bottom nodes must be either numeric or in a valid date format, depending on the attribute dimension type.

  > **Note:**
  >
  > When you add a parent node to a Numeric or Date type attribute dimension, a validation message may be displayed even if you selected a parent node type. The validation error will be resolved after you add a child node under the parent and then re-validate the request.

- Attribute dimensions can be imported and exported from Planning, Planning Modules, Financial Consolidation and Close, and Enterprise Profitability and Cost Management applications.

> **Note:**
>
> When you export attribute dimensions, only the Default alias table is included in the export file.

Working with Registered Dimensions

When you add a registered dimension as an attribute dimension, a node data type property is created to reference the registered dimension.

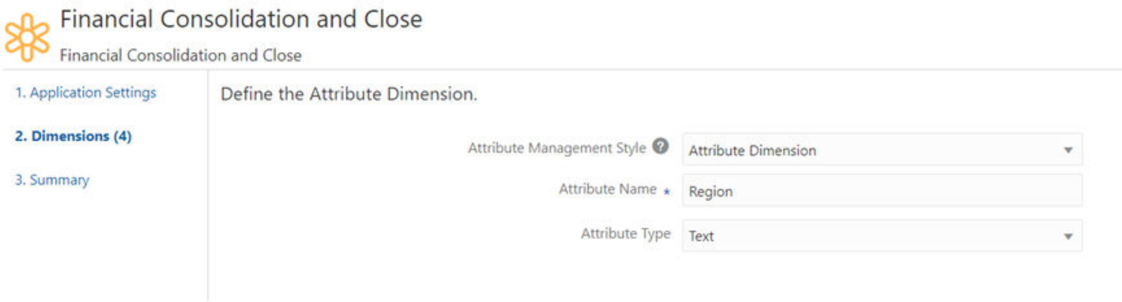
Registered dimensions that are used as attribute dimensions can be imported and exported from Planning, Planning Modules, Financial Consolidation and Close, and Enterprise Profitability and Cost Management applications.

> **Tip:**
>
> This can be used when an attribute of a dimension needs to refer to itself, for example, the triangulation currency of a Currency dimension.

Attribute Dimension Example

This topic provides an example of using an attribute dimension to reference Region values in an attribute dimension from an Entity base dimension.

In this example, we modified the registration for a Financial Consolidation and Close application to add an attribute dimension named Region to the Entity base dimension:


Financial Consolidation and Close
Financial Consolidation and Close
1. Application Settings
2. Dimensions (4)
3. Summary

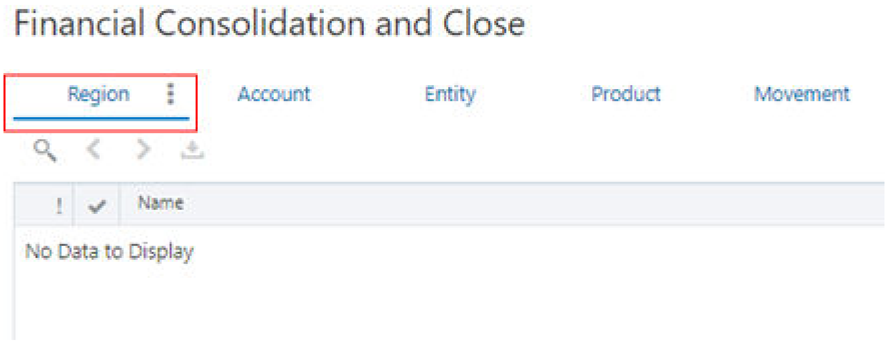
Define the Attribute Dimension.

Attribute Management Style: Attribute Dimension

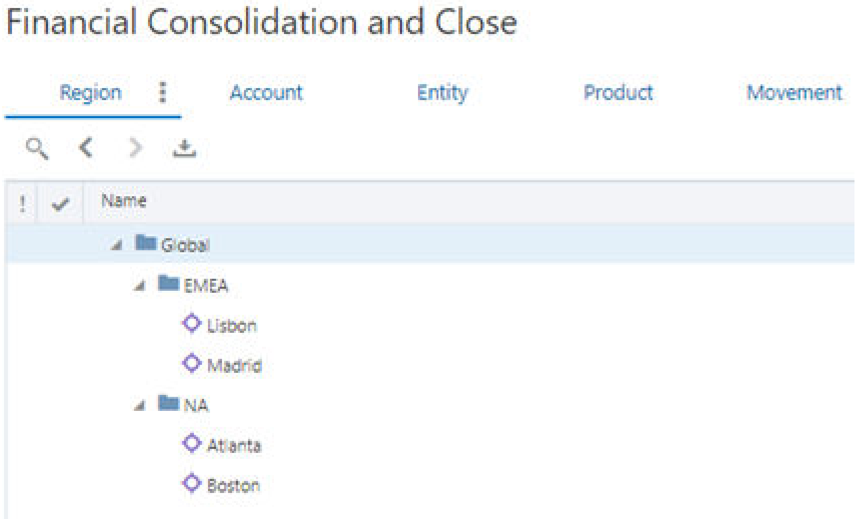
Attribute Name *: Region

Attribute Type: Text

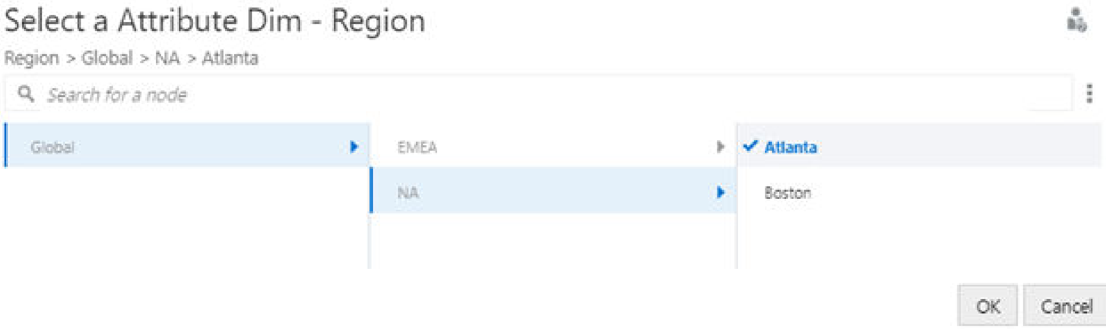
The viewpoint for the Region attribute dimension is added to the default view for the application:



Next, we add members to the attribute dimension. In this example, we will add a top node called Global and two parent nodes: NA, with Boston and Atlanta as child nodes, and EMEA, with Lisbon and Madrid as child nodes.



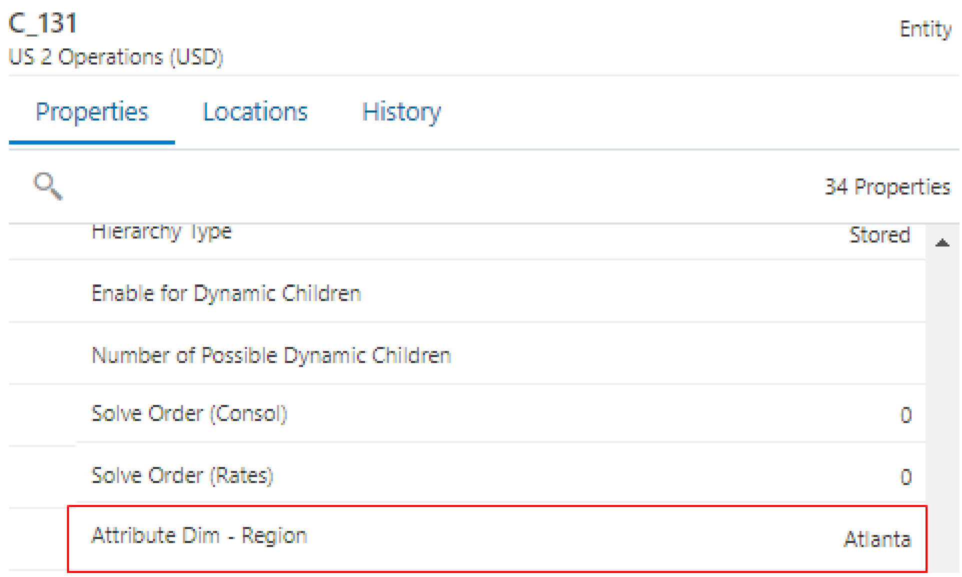
Now, when we edit a member in the Entity base dimension, in the property pane we can click ⌖ in **Attribute Dim - Region** to view the attribute dimension members that can be assigned to the base dimension member. In this example, we added the Atlanta region to the C_131 member.


Select a Attribute Dim - Region
Region > Global > NA > Atlanta
Search for a node
Global
EMEA
NA
✓ Atlanta
Boston
OK
Cancel

Tip:

By default, the node description is displayed next to the node name in the node selector. Click **Show Node Property** to select a different property to display in the node selector. The properties that you have configured to be displayed as columns in the viewpoint are available to select. See Configuring How a Viewpoint Displays Properties.

The node is updated with the attribute dimension property that we selected.



C_131
US 2 Operations (USD)

Entity

Hierarchy Type	Stored
Enable for Dynamic Children	
Number of Possible Dynamic Children	
Solve Order (Consol)	0
Solve Order (Rates)	0
Attribute Dim - Region	Atlanta

20 Working with Data Objects and the Data Chain

The basic organizational structure in Oracle Fusion Cloud Enterprise Data Management is an application, and each application contains dimensions (such as Product and Entity). Each dimension contains a series of related data objects called data chains, which consist of node types, hierarchy sets, node sets, and viewpoints. These data objects are the building blocks of the Cloud EDM solutions you will create.

The way in which a viewpoint and its related data objects are configured determine the nodes available in the viewpoint. In addition, the data chain defines characteristics of the viewpoint such as those in the following list:

- Whether the nodes are a list or a hierarchy
- Rules defining relationships between the hierarchy's nodes
- The nodes' properties
- The actions users can perform

The following table summarizes the data objects. For more information about the data objects, see Understanding Data Chains.

Data Object	Description
Node type	Contains all nodes of a given category. For example, account nodes would be contained by an Accounts node type. See Working with Node Types
Hierarchy set	Defines nodes' parent-child relationships. See Working with Hierarchy Sets.
Node set	Defines the group of nodes available in a viewpoint. See Working with Node Sets.
Viewpoint	Presents the node set to users, and defines actions users can perform and properties users can edit. See Understanding Viewpoints.

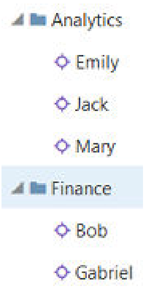
Node types, hierarchy sets, and node sets are reusable objects. For example, if you have a hierarchy that is used in different viewpoints, you typically would use its hierarchy set object in all of those viewpoints.

For more information, see:

- Understanding Data Chains
- Understanding the Lifecycle of Data Objects and the Data Chain

Example 20-1 Data Chain for a Hierarchy Example

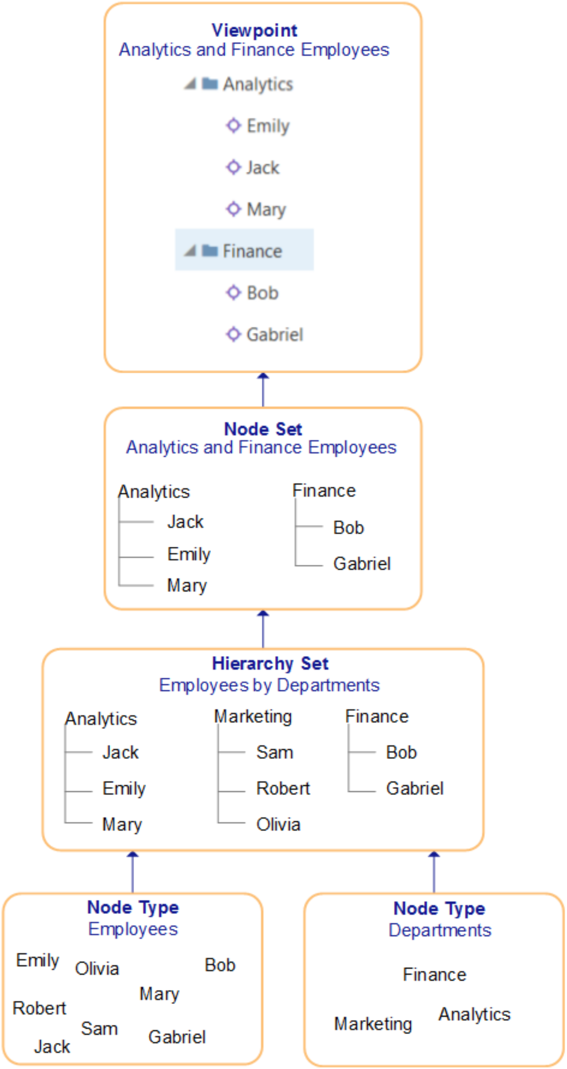
Suppose that you are working with hierarchies that list employees by departments. The viewpoint contains hierarchies that represent the employees of the Analytics and Finance departments, as shown in the following example:



The application also contains another hierarchy for the Marketing department, but that hierarchy is not included in the viewpoint. The following table describes the data chain upon which this viewpoint is built:

Data Object	Description
Node types	The nodes belong to the following node types: • The employee nodes are contained by the Employees node type. • The department nodes are contained by the Departments node type.
Hierarchy set	Defines the parent-child relationships as follows: • Departments nodes can have child Employees nodes. • Employees nodes cannot have child nodes. The hierarchy set also defines the child Employees nodes of the Departments nodes.
Node set	Includes the Analytics and Finance departments' hierarchies. The hierarchy set's Marketing hierarchy is excluded.
Viewpoint	Displays the nodes specified in the node set.

The following diagram shows the data chain used to define the viewpoint:



Videos

Your Goal	Watch This Video
Learn more about data objects and the data chain.	Understanding the Information Model in Enterprise Data Management

Understanding Data Chains

A data chain is a series of related data objects. The nodes available to users in a viewpoint are defined by its data chain.

The configurations of objects on a data chain also define a viewpoint's available properties.

The data chain consists of the following data objects:

- **Application**: a business application serving a specific purpose such as Oracle Financials Cloud General Ledger or Planning which includes enterprise data to be managed.

  - **Dimension**: a business dimension of an application such as Account, Entity, and Product.
    - **Node type**: a type of enterprise data record in a dimension such as a General Ledger segment value, a Planning dimension member, or a General Ledger to Planning mapping. Each node type defines the characteristics of a record, which is called a *node*. A dimension can consist of one or more node types. Nodes and their properties are stored in a node type.
    - **Hierarchy Set**: a set of hierarchy relationships between nodes of particular node types in a dimension. Each hierarchy set defines the node types which are related and the rules for hierarchies in that set. A hierarchy set can contain one or more Root Nodes which define the starting points for hierarchy relationships in the set. Relationships and their properties are stored in a hierarchy set.
    - **Node Set**: a set of nodes in a dimension which has a particular structure such as a List or a Hierarchy. A list node set can consist of one or more node types in a flat list. A hierarchy node set can consist of a hierarchy set and one or more top nodes. A top node of a node set can be a root node from the hierarchy set or a descendant of a root node.
  - **Viewpoint**: a business perspective of a dimension used for a particular purpose such as viewing a list of accounts or managing a product hierarchy or exporting an entity structure. Viewpoints are organized into one or more *views*. Each viewpoint uses a node set and controls how users work with data in that node set in a specific view.
- **View**: a group of viewpoints used for a particular purpose, such as managing data for a dimension across applications or integrating data from and to an external system. Each application uses a *default view* to import and export enterprise data. Each viewpoint in the default view represents the data as it exists in the external system. Users can define additional views of their own to view and manage data for specific business purposes.

> **Note:**
>
> If you are familiar with relational database concepts, node types and hierarchy sets are similar to database tables in that they hold data, node sets are similar to database predicates or "Where" clauses in that they select the nodes that you want to work with, and viewpoints are similar to database views in that they provide the user interface in which users work with data.

Data objects are modular; node types, hierarchy sets, and node sets can be used in multiple data chains. The data objects have the following relationships:

- A node type is used by one or more hierarchy sets or node sets.
- A hierarchy set uses one or more node types.
- A hierarchy set is used by one or more node sets.
- A node set uses one hierarchy set.
- A node set for a list uses one or more node types.

> **Note:**
>
> If a node set uses a hierarchy set, the node types are inherited from the hierarchy set.

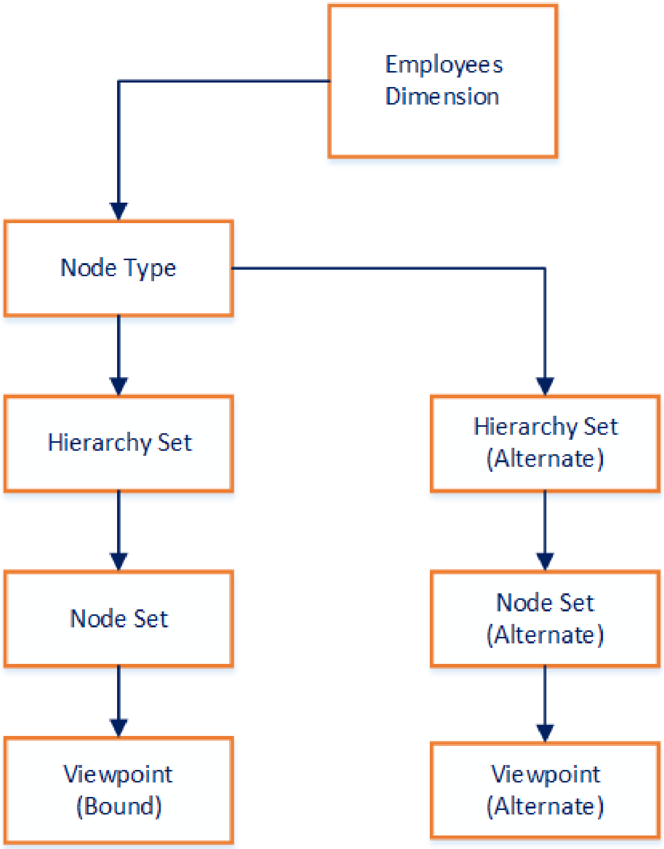
- A node set is used by one or more viewpoints.
- A viewpoint set uses one node set.

Dimension with Multiple Data Chains Example

To understand the relationship of dimensions and data objects, consider the following example:

- The user is working with an Employees dimension and needs to create an alternate viewpoint that models the results of a reorganization.
- To create the viewpoint for the alternate viewpoint, the user performs the following steps:
  1. Reuse the node type that was used in the data chain of the bound viewpoint.
  2. Create a new hierarchy set that references the node type.
  3. Create a new node set that references the hierarchy set.

The data objects used in both data chains are assigned to the Employees dimension, as shown in the following diagram:



> **Note:**
>
> The alternate viewpoint and its hierarchy set and node set are not bound to a dimension, and thus are not required to conform with the dimension's binding rules. The node type in this example complies with the dimension's binding rules even though it is part of the alternate viewpoint's viewpoint. This is because the node type is also part of the bound viewpoint's data chain.

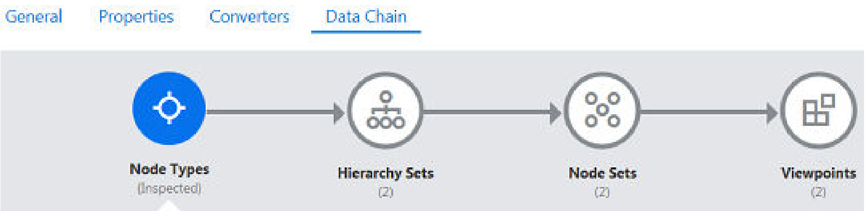
For more information, see:

- Working with Data Objects and the Data Chain
- Inspecting Data Chains
- Defining Alternate Views and Viewpoints

Inspecting Data Chains

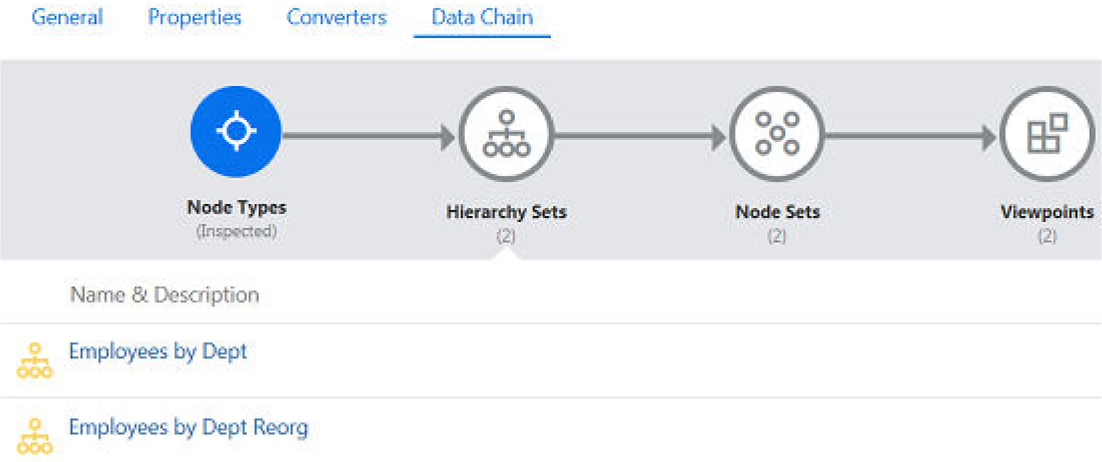
Many Oracle Fusion Cloud Enterprise Data Management screens have a **Data Chain** tab. You can use this tab to inspect the data objects on a data chain. For more information about data chains, see Understanding Data Chains.

The following example shows that a node type is used in the data chains of two hierarchy sets, node sets, and viewpoints.



You can inspect and access objects on the data chain by clicking the applicable icon. Clicking on an icon display links to data objects. If you want to inspect one of the objects, click the link and a new tab displays on the bottom of the screen.

In the following example, the user has clicked the **Hierarchy Sets** icon. Links to the definitions of both hierarchy sets that use the node type are displayed.



Name & Description

Employees by Dept

Employees by Dept Reorg

Working with Node Types

A node type is a collection of an application's nodes that share a common business purpose. Use node types to define nodes' properties and to define rules that convert a node type to another node type.

Each node is a member of a node type. For example, a node for an account typically will be a member of its application's node type for accounts.

The following list provides examples of how node types categorize nodes:

- The nodes for a Company dimension could be assigned to a node type for companies.
- The nodes of a hierarchy that organizes employees by office location could be assigned to node types for either employees or locations.

The following list describes node type features and considerations:

- Every node is assigned a node type.
- A node is uniquely identified by the combination of node name and node type.
- A node's type never varies. In other words, if a node is available in multiple viewpoints, its node type can't be changed in any of the viewpoints.
- A node type defines the properties available in any viewpoint that includes the node type in its data chain.
- Node types can have one of the following classes. You cannot change the class of a node type after it has been created:
  - **Normal**: For dimension types other than lookup dimensions. You can add or remove properties to a node type with a class of Normal.
  - **Lookup**: Specialty node type used to store reference data from an external system. Contains generic reference properties that you cannot change. Can be used in any dimension type.
  - **Legacy GL**: Specialty node type used to model chart of account values from a legacy general ledger system. Contains basic general ledger and mapping set properties that you cannot change. Can be used in any dimension type.

  > **Note:**
  >
  > Nodes with the Lookup or Legacy GL class node type are not included in your total record count. See Working with Specialty Node Type Classes.

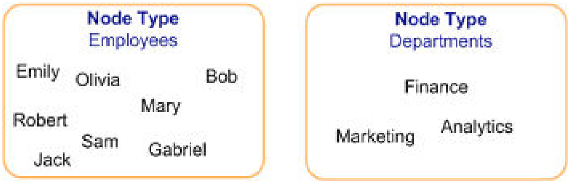
- The node type data object doesn't have dependencies on other data chain objects.
- You can specify that a node type's nodes can be converted to another node type, see Working with Node Type Converters.

For more information, see:

- Working with Data Objects and the Data Chain
- Adding, Removing, and Configuring a Node Type's Properties
- Inspecting a Node Type

Example 20-2 Node Type Example

The Data Chain for a Hierarchy Example uses node types for Employees and Departments, as shown in the following diagram:



The example uses two node types because Departments and Employees nodes would be treated differently. For example, the Employees nodes could have properties for employees' addresses, phone numbers, and e-mail addresses, while the Departments nodes could have a property for department managers.

Videos

Your Goal	Watch This Video
Learn about setting up node types.	▶ Setting Up Node Types

Creating a Node Type

Best Practice

It's a best practice to create a new node type when you want to:

- Use different properties. For example if a Location node type has 10 properties and you only want the user to see 5 of the properties, create a new Location_dm node type with the 5 properties.
- Define specific node type relationships. For example, suppose the Entity node type contains Companies and Departments as Entity nodes. If you want to set up Company as a parent of Department, create a Company node type and Department node type. Then define the hierarchy set with Company node type as a parent of Department node type.
- Classify a unique set of nodes for a business purpose. For example, suppose the Entity node type contains departments and acquired departments. If you want to distinguish the differences, create a Department node type and Acquired Department node type. Then assign the nodes to the appropriate node type.

For more information, see Working with Node Types.

Videos

Your Goal	Watch This Video
Learn more about setting up node types.	▶ Setting Up Node Types

1. From **Node Types**, click **Create**.
2. Select the application dimension.

   The dimension type is displayed below the selected dimension.
3. Select the node type class:

- **Normal** (not available in Lookup dimensions)
- **Legacy GL**
- **Lookup**

> **Note:**
>
> For considerations in working with specialty node types such as Legacy GL and Lookup, see Working with Specialty Node Type Classes.

An inspector for the node type is displayed.

4. Enter a name and (optionally) a description for the node type, and then click **Create**.
5. **Optional:** Select **General**, click **Edit**, and then define a **Default Qualifier** (and **Alternate Qualifiers**, if needed) to add a prefix or suffix to make the node type name unique. You must define a default qualifier in order to specify alternate qualifiers. See Working with Node Type Qualifiers.

> **Note:**
>
> Leading and trailing whitespace characters are trimmed from the default and alternate qualifiers.

6. **Optional:** If this node type is going to be used in cluster key-based deduplication, select the **Clustering Property** from the drop down menu. See Configuring a Clustering Property for a Node Type.
7. Select **Properties**, and then click **Edit**. You can select which properties are required and change the order in which they are displayed. You can add or remove properties from the node type by clicking **Manage** and making the necessary changes.
8. Click **Save**.

If you need to set up a node type converter for the node type, see Working with Node Type Converters.

Configuring a Clustering Property for a Node Type

You must configure a clustering property for a node type in order to use that node type in cluster key-based deduplication (see Deduplicating Using a Cluster Key). This clustering property is used to identify the set of nodes to be matched in a selected viewpoint.

Use the **Clustering Property** drop down menu to select the clustering property for this node type. You can select from properties that meet the following criteria:

- They are a String data type
- They are an indexed property
- They have an Allowed Values list for this node type

Working with Specialty Node Type Classes

Node types with a specialty node type class have a limited set of capabilities that can be used to store or model data from external systems or to manage users in Users applications.

There are four classes of specialty node types:

- **Lookup**: Used to store data from an external system that you can point to in a viewpoint by using a node data type property. See Working with Lookup Dimensions.
- **Legacy GL**: Used to model chart of account values from a legacy general ledger systems in order to support the migration to a cloud general ledger system.
- **User**: Used to manage users in Users applications only.
- **User Rollup**: Used to manage rollups in Users applications only.

> **Note:**
>
> You cannot use the User or User Rollup node type class in application types other than Users applications.

Considerations

- Nodes with a specialty node type class have a limited set of predefined properties that you cannot change. See the following topics:
  - Predefined Properties for Lookup Node Types
  - Predefined Properties for Legacy GL Node Types
  - Predefined Properties for Users Applications
- Nodes with a specialty node type class are not included in your total record count. See Record Count Log Report.
- All nodes in specialty node type classes have limited integration capabilities at the dimension and viewpoint levels:
  - **Lookup and Legacy GL**:
    - You can run viewpoint loads.
    - You cannot import, export, or viewpoint download.
    - You can extract from mapping bound viewpoints only (see Understanding Bindings and Bound Data Objects).
  - **User and User Rollup**:
    - You can run viewpoint loads.
    - You cannot import, export, extract, or viewpoint download.

Predefined Properties for Lookup Node Types

Lookup dimensions use the Lookup class node type, which has these predefined properties. You cannot create or remove properties from Lookup class node types.

Properties in the Lookup namespace are generic properties that you can use to store lookup data. You can change their labels in a viewpoint to better represent their usage in the dimension. For example, if you are using a lookup dimension to store employee lookup data, you could change the label of the `Lookup.String` property to "Employee ID". You can also hide properties that you are not using in the viewpoint. See Configuring How a Viewpoint Displays Properties.

Property	Namespace	Data Type	Level	Description
Boolean	Lookup	Boolean	Node	Generic Lookup Boolean property
Bottom Node	CoreStats	Boolean	Relationship	Determines whether node has children
Date 1	Lookup	Date	Node	Generic Lookup date 1 property
Date 2	Lookup	Date	Node	Generic Lookup date 2 property
Description	Core	String	Node	Node description
Float	Lookup	Float	Node	Generic Lookup float property
Integer	Lookup	Integer	Node	Generic Lookup integer property
Level	CoreStats	Integer	Node	Level of the node in a hierarchy node set
Name	Core	String	Node	Node name
Parent	CoreStats	Node	Relationship	Parent node name of the node in the current location
String	Lookup	String	Node	Generic Lookup string property

Predefined Properties for Legacy GL Node Types

Properties in the Legacy GL namespace can be used to store data from your legacy GL system. You can change their labels in a viewpoint to better represent their usage in the legacy GL system. You can also hide properties that you are not using in the viewpoint. See Configuring How a Viewpoint Displays Properties.

You cannot add or remove properties from Legacy GL class node types. You can rearrange the property order, and you can set the Required flag.

Property	Namespace	Data Type	Level	Description
Name	Core	String	Node	Node name
Description	Core	String	Node	Node description
Parent	CoreStats	Node	Relationship	Parent node name of the node in the current location
Bottom Node	CoreStats	Boolean	Relationship	True if the node has no children
Account Type	LegacyGL	String	Node	Legacy GL account type
Alt Description	LegacyGL	String	Node	Legacy GL alternate description

Property	Namespace	Data Type	Level	Description
Allow Posting	`LegacyGL`	Boolean	Node	Legacy GL allow posting flag
Enabled	`LegacyGL`	Boolean	Node	Legacy GL enabled flag
Target Name	`LegacyGL`	String	Relationship	Mapping target name for source
Start Date	`LegacyGL`	Date	Relationship	Effective start date
End Date	`LegacyGL`	Date	Relationship	Effective end date
Default Mapping	`LegacyGL`	Boolean	Relationship	Default mapping (Y/N)

Working with Node Type Qualifiers

A node type qualifier is a prefix or suffix that you define for a node type which allows for unique node type naming. You define node type qualifiers for external applications that use unique naming for node types or dimensions. For example, an application may use these qualifiers for entities, accounts, and cost centers: ENT_, ACCT_, and CC_.

Qualifiers are used by Oracle Fusion Cloud Enterprise Data Management when you compare nodes, display viewpoints side by side, and drag and drop from one viewpoint to another.

For example, if you have a cost center named 750 in your general ledger application and you want to add it to your Planning application, you can drag and drop the cost center from the general ledger viewpoint to the Planning viewpoint. With the node type qualifier defined and a node type converter set up, the cost center in Planning will be added as CC_750.

> **Note:**
>
> When calculating node counts, two nodes are counted as equivalent if the source and target nodes have the same base name after the qualifier is removed (default or alternate qualifier, prefix or suffix).
> See Viewing the Record Count.

Default and Alternate Qualifiers

Node types support a default qualifier as well as up to three alternate qualifiers, which are used when the node can't be found using the default qualifier. All default and alternate qualifiers for a given node type must be either a prefix or a suffix. You cannot mix prefixes and suffixes as qualifiers in a single node type.

> **Note:**
>
> You must have a default qualifier defined in order to specify alternate qualifiers. Deleting the default qualifier will also delete the alternate qualifiers.

When comparing, locating, and adding or inserting nodes across node types (using a node type converter), default and alternate qualifiers are handled as follows:

- When comparing, locating, and aligning nodes:
  - If the source node name begins with the default qualifier, remove the qualifier and use the result as the base name. If the default qualifier is not found, look for the node using each of the alternate qualifiers. If one is found, remove the qualifier and use the result as the base name.
  - If the target node type has a default qualifier defined, add it to the beginning of the name and look for a node with that node type and name in the target viewpoint. If the default qualifier isn't found, look for the node using each of the alternate qualifiers.
- When adding or inserting nodes across node types:
  - If the source node name begins with the default qualifier, remove the qualifier and use the result as the base name. If the default qualifier is not found, look for the node using each of the alternate qualifiers. If one is found, remove the qualifier and use the result as the base name.
  - For target nodes, the default qualifier is always used for the new node name.

> **Note:**
>
> You can also define property name transformations in the node type converter in order to transform the node name when adding or inserting nodes. See *Property Transformations* in Working with Node Type Converters.

Working with Node Type Converters

You can create node type converters to convert one node type to another node type.

> **Note:**
>
> Node type converters are needed to compare, locate, align, and drag and drop nodes of *different node types* across two viewpoints in side by side layout.

You start with the node type that you want to convert to and create a converter by selecting which node type you want to convert the node type from. Let's say that you want to compare nodes in a viewpoint that uses a node type called Acquired Depts. to nodes in another viewpoint that uses a node type called All Depts. You'll need to create a node type converter that converts the Acquired Depts node type to the All Depts node type.

Videos

Your Goal	Watch This Video
Learn about converting node types.	Converting Node Types

Tutorials

Tutorials provide instructions with sequenced videos and documentation to help you learn a topic.

Your Goal	Learn How
Build an expression in a node type converter to derive the parent of a node	Finding a parent node using a node type converter expression

With a node type converter set up between two applications:

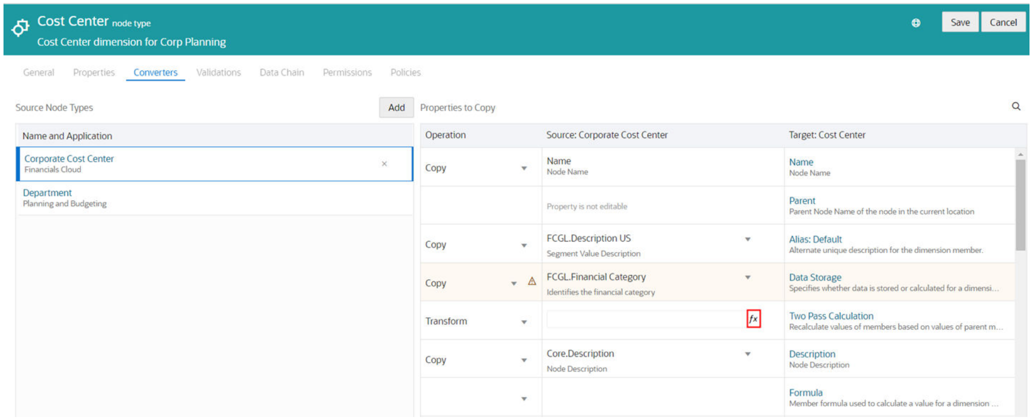
- If the node does not exist in the target node type used by the viewpoint, the node is converted and added to that node type. If the node is added to the target node type, the properties specified in the node type converter are copied to the target node.
- If a property that is specified to be copied in the node type converter is set to read-only in the target viewpoint, then no value for that property is copied to the target node.

> **Tip:**
>
> If you need to set up a node type qualifier for the converter, see Working with Node Type Qualifiers.

Creating Node Type Converters

To create a node type converter:

1. Inspect a node type that will be your "to" node type.
2. Select the **Converters** tab.
3. Click **Edit** and then click **Add**.
4. Select the node type that will be converted. This will be the "from" node type.
5. On the **Properties to Copy** screen, properties that are the same or similar automatically populate. In the **Operation** column, select from these options for properties:
   - **Copy**: Copy an existing property from the source. Select the property from the **Source** column drop-down list. Properties that are the same or similar are automatically selected but can be changed.
   - **Transform**: Define an expression to calculate a value for the target property. In the **Source** column, click **Define Expression** *fx*, define an expression, and then click **Apply**. For information on how to define expressions, see Using Expressions to Define Custom Business Logic.



Name and Application
Corporate Cost Center Financials Cloud
Department Planning and Budgeting

Operation	Source: Corporate Cost Center	Target: Cost Center
Copy	Name Node Name	Name Node Name
	Property is not editable	Parent Parent Node Name of the node in the current location
Copy	FCGL.Description US Segment Value Description	Alias: Default Alternate unique description for the dimension member.
Copy ⚠	FCGL.Financial Category Identifies the financial category	Data Storage Specifies whether data is stored or calculated for a dimensi...
Transform	*fx*	Two Pass Calculation Recalculate values of members based on values of parent m...
Copy	Core.Description Node Description	Description Node Description
		Formula Member formula used to calculate a value for a dimension ...

Tip:

Click 🔍 and enter text to search within the property list. The search is performed on property names and descriptions.

Note:

- If the **Operation** column for a property is blank, the property is not editable.
- If the **Operation** column is blank with a drop down list, you can select from the options above.
- If the row for a property has a **Caution** ⚠ icon, the source property has been mapped to a read only target property. Any changes made to the source property will not be propagated to the target property.

6. **Optional**: To remove a property transformation from the node type converter, change the Operation for the property to Copy or blank.
7. Click **Save**.

For more information, see:

- Working with Node Types
- Inserting Nodes and Related Nodes

Property Transformations

Property transformations calculate a defined value for a target property in a node type converter and only run when sharing, mapping, or synchronizing data across viewpoints. Transformations are configured for node type converter properties using expressions.

> **Note:**
>
> You can transform properties with these data types.
>
> - Boolean
> - Date
> - Integer
> - List
> - Node
> - String

Property transformations are calculated during the following operations:

- Add or insert nodes across viewpoints
- Create request items from compare differences
- Generate subscription requests

> **Note:**
>
> Property name transformations cannot be used when locating or aligning properties. Instead, you can add alternate qualifiers to the node type so that if a node can't be located using the default qualifier, they can be searched for using the alternates. See *Default and Alternate Qualifiers* in Working with Node Type Qualifiers.

Considerations

When using expressions to configure property transformations keep this information in mind:

- The SourceNode object is available to access information from a source node.
- Information cannot be accessed from the target node.
- Derived properties for a source node are accessible.

Transforming Node Parents

By default, when you set up a transformation for the Name property, that transformation is also used for the Parent in interactive requests, viewpoint comparisons, and when generating subscription requests. For example, if you are concatenating a suffix to the Name in the transformation expression, the suffix is also added to any Parent references.

However, you can set up a transformation for the Parent of a source node that is separate from the transformation on the Name property. The Parent transformation runs on the source parent and is also used for the old parent in cases where the source node was moved to a different parent.

If there are separate transformations set up for the node Name and Parent, when a node is manually inserted across viewpoints and you are also inserting the node's descendents, when request items are created from compare results, or when a subscription generates request items, the following takes place for the Parent:

- If a transformation operation is set up for the Parent in the node type converter, the *Parent* transformation is run for each source parent and the resulting value is used as the Parent for each hierarchy action. For subscription requests, the Parent transformation is also used for the Old Parent when applicable.

  > **Note:**
  >
  > If the source viewpoint is a list viewpoint, then the parent transform is not run (because there is no source parent). Instead, the `[unknown]` keyword is used for the parent in the subscription request item.

- If a transformation operation is set up for the Name but not the Parent in the node type converter, the *Name* transformation is run for each source name and source parent, and the resulting value for the source parent is used as the Parent for each hierarchy action. For subscription requests, the Name transformation is also used for the Old Parent when applicable.
- If a Copy operation is used for the Parent in the node type converter, the source parent (with a node type qualifier if applicable) is used as the Parent for each hierarchy action. For subscription requests, the source parent (with a node type qualifier if applicable) is also used for the Old Parent when applicable.

> **Note:**
>
> For subscriptions, parent transformations in node type converters are run for top nodes in source hierarchy viewpoints when request items for a subscription request are generated. This enables top nodes which were added in a source viewpoint to be placed under a different parent in a target hierarchy viewpoint.

> **Tip:**
>
> When testing expressions for transforming node parents, make sure you are evaluating the expression on the parent node and not the child node.

Filtering Target Node Types

For target viewpoints with multiple node types, you can filter the list of available node types to only those for which you want to allow changes. Node type filtering is applied when:

- You drag and drop a node across viewpoints
- You insert a node from another viewpoint
- Subscription requests are generated

To filter node types out of the available node types list, configure the node type so that it has no allowed actions or editable properties:

1. Inspect the target viewpoint.
2. On the **Definition** tab, click **Edit**.
3. Select a node type and clear all the allowed actions and editable properties.
4. Click **Save**.

Inspecting a Node Type

From the node type inspector, you can view and edit information about a node type. For more information, see Working with Node Types.

Videos

Your Goal	Watch This Video
Learn more about setting up node types.	Setting Up Node Types.

1. From **Node Types** scroll to the node type that you want to inspect, click ⋮ in the **Actions** columns, and then select **Inspect**.
2. Select from the options available on these tabs:
   - **General**—Displays information on the application, dimension, status, class, and binding status of the node type. See Binding Status.

     Click **Edit** to perform any of these actions on the node type:
     - Change the name, description and status, see Understanding the Lifecycle of Data Objects and the Data Chain.
     - Add or modify a node type qualifier, see Working with Node Type Qualifiers.
     - Select **Combination Node Type** to designate a node type that represents a combination of values from multiple other node types. Combination node types are noted in the Record Count Log report, see Record Count Log Report.
   - **Properties**—Add or remove properties from the node type, set properties as required or not, and reorder the properties in the displayed list, see Adding, Removing, and Configuring a Node Type's Properties.
   - **Converters**—Create node type converters to convert one node type to another node type, see Working with Node Type Converters.

     > **Note:**
     >
     > For help with building expressions in node type converters, see Using Expressions to Define Custom Business Logic.

   - **Validations**: Add custom business logic to enforce data rules for the node type by creating custom validations. See Custom Validations
   - **Data Chain**—Provides a clickable graphic so that you can see the hierarchy sets (if applicable), node sets, and viewpoints referenced by the node type.
   - **Permissions**—Set up user and group permissions for the node type, see Working with Permissions.
   - **Policies**—Configure an approval policy for the node type, see Configuring Policies.

Searching for Node Types

When you perform a search in the Node Types list, the system searches for your text in node type name and description, application name and type, dimension name and binding status, and class. The fields that contain the search text are displayed in the results window in bold-faced font. You can also include archived node types in your search.

To search in the Node Types list:

1. From **Node Types**, click 🔍.
2. **Optional**: If you want your search to include archived node types, click **Filter** ▽ and then select **Show Archived**.
3. Enter a text string to search for. The search is performed automatically after you stop typing.

   To close the search, click ✕ .

Adding, Removing, and Configuring a Node Type's Properties

Node types define the properties available in viewpoints and whether they are required.

> **Note:**
>
> If you add a property to a node type, and the property has not been added to a dimension through the application registration process, the property will not be included in imports and exports. See Working with Properties.

> **Note:**
>
> You cannot add or remove properties from a Legacy GL or Lookup class node type.

For more information, see:

- Working with Properties
- Working with Node Types
- Configuring How a Viewpoint Displays Properties

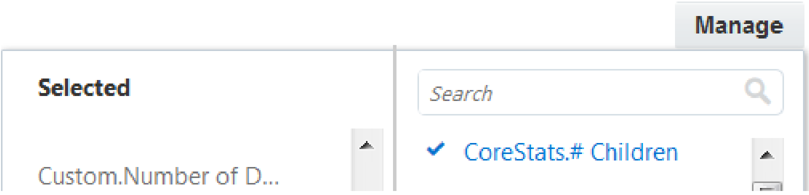
To configure a node type's properties, take the following steps:

1. From Node Types, find the node type, click ⋮, and then select **Inspect**.
2. From **Properties**, click **Edit**.
3. (Optional) Click **Search** 🔍 and enter text to search the property name or description for. Click **X** to clear the search filter and display all properties.
4. To make a property required or optional, select or clear the property's check box in the **Required** column.

> **Note:**
>
> Setting a property as required at the node type level marks it as required in all viewpoints that contain the property. If you want to make the property required in some viewpoints but not others, you can specify that the property is required for specific viewpoints only. See Configuring How a Viewpoint Displays Properties.

5. To change the order of the properties, in the **Actions** column click **Actions** ⋮, and then select First, Up, Down, or Last.
6. To add or remove a property, take the following steps.
   a. Click **Manage**.

      A dialog box displays. The **Selected** column shows the properties that have already been added to the node type. The second column displays all properties available; the properties that already have been added to the node type have a check mark next to the property name.
   b. In the second column, find and select the property to add or remove, then take one of the following steps:
      - To add a property, select it. A check mark displays next to the property you selected, as shown below:



      - To remove a property, select it. The check mark next to the property you selected is removed.
   c. When you are finished adding and removing properties, click **OK**.

      The grid listing the node type's properties displays.
   d. If you added properties, use the check boxes in the **Required** column to specify whether the properties are required or optional.
7. Click **Save**.

Working with Hierarchy Sets

Hierarchy sets define nodes' parent-child relationships. For example, you would use a hierarchy set to define a hierarchy in which products roll up to product categories.

Hierarchy sets define the following rules for a hierarchy:

- The node types that the hierarchy consists of.
- The hierarchy's levels, which are the relationships between node types. The following list provides examples of hierarchy levels:
  – An organization chart's hierarchy set would specify that employee nodes can be parent nodes of other employee nodes.

  - A hierarchy set that groups employees by office buildings would specify that building nodes can be parents of employee nodes but not other building nodes, and that employee nodes cannot have child nodes.
  - A hierarchy set for rollups by department would specify that rollup nodes can be parents of both department and rollup nodes, and that department nodes can be parents of department nodes.
- Whether the hierarchy allows shared nodes, which are nodes that can be included in different branches of the hierarchy.
- Whether the hierarchy allows custom sort orders in viewpoints, imports, and exports.

The following list describes hierarchy set features and considerations:

- Hierarchy sets also define the relationships of the nodes themselves. Within a hierarchy set, a node's parent-child relationships are the same even if the node occurs in different branches.

> **Note:**
>
> If you need to define a hierarchy where nodes have parents that differ from those in a hierarchy set, you would need to create a new hierarchy set. For example, you would create a new hierarchy set to evaluate a reorganization that breaks a business unit into multiple business units,

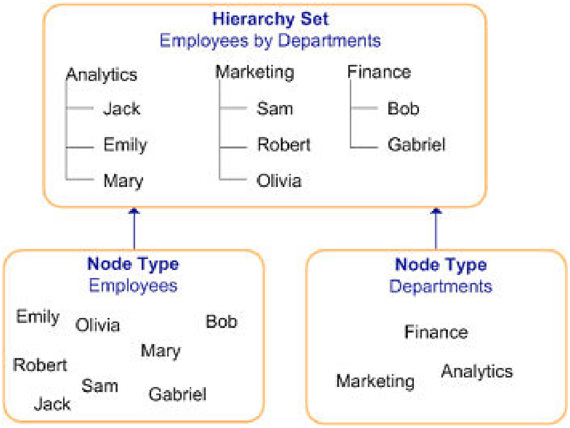
- Hierarchy set data objects require one or more node types, and are referenced in node set objects.

For more information, see:

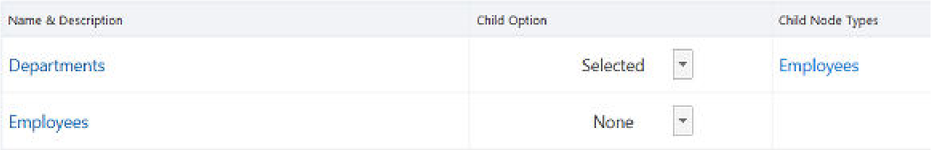
- Working with Data Objects and the Data Chain
- Inspecting a Hierarchy Set

Example 20-3 Hierarchy Set Example

The following diagram represents a hierarchy set that contains hierarchies for the Analytics, Finance, and Marketing hierarchies:

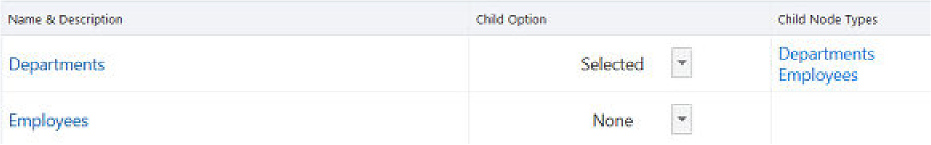


The example's hierarchy set specifies that Departments nodes can have Employees child nodes and Employees nodes cannot have child nodes, as shown in the following example:



Name & Description	Child Option	Child Node Types
Departments	Selected	Employees
Employees	None	

Suppose you need to create a viewpoint in which Departments nodes can contain both Departments and Employees nodes. You would create a hierarchy set specifying that Departments nodes can have **Child Node Types** of Employees and Departments, as shown in the following example:



Name & Description	Child Option	Child Node Types
Departments	Selected	Departments Employees
Employees	None	

You would include the hierarchy set in the viewpoint's data chain.

Videos

Your Goal	Watch This Video
Learn about creating hierarchy sets.	▶ Creating Hierarchy Sets

Creating Hierarchy Sets

When you create a hierarchy set, you'll select the node types that you want to work with in the set and whether the node type allows children. If the node type has children then you'll also need to select the node types for the child. For example, you may have an Account Rollup node type that allows children of the Account node type. For more information, see Working with Hierarchy Sets.

Best Practice

It's a best practice to create a new hierarchy set when you want to:

- Define an alternate hierarchy using different relationships between the same node types, see Alternate Viewpoints Using Different Parent-Child Relationships.
- Define an alternate hierarchy using different node types than another hierarchy set, see Alternate Viewpoints Using Different Node Types
- Version hierarchies. For example, if you are using a hierarchy set, and know that it will change in the next year due to a reorganization, create a new hierarchy set to represent the reorganization and use it for planning purposes.

Considerations

- For normal and unbound dimensions:
  - For unbound hierarchy sets:
    - **All unbound hierarchy sets except those in Users applications**: You can add Normal, Legacy GL, or Lookup class node types. All of the node types must be of the same class.
    - **Unbound hierarchy sets in Users applications only**: You can add both User and User Rollup class node types, but no other node type classes.
  - For mapping bound hierarchy sets (see Understanding Bindings and Bound Data Objects): You can add all of the same class node types, or a mix of normal and one

specialty class (Legacy GL or Lookup) node types. You cannot add both specialty types together.

  - For all other bound hierarchy sets other than mapping bound: You can add only normal class node types.

- For lookup dimensions: You can add any Legacy GL or Lookup class node types to any hierarchy set (bound or not). All of the node types must be of the same class.

To create a hierarchy set:

1. From **Hierarchy Sets**, click **Create**.
2. Select the application dimension, define the name and description for the hierarchy set, and then click **Create**.
   An inspector for the hierarchy set opens. You'll need to select and configure node types for the hierarchy set before it's ready to use.
3. Select **Definition** tab, click **Edit**, click **Manage**, select one or more node types to use in the hierarchy set, and then click **OK**.

   > **Note:**
   >
   > See **Considerations**, above, for the class of node types that you can select. For specialty node type classes, the class is displayed after the node type name in the node type selector.

4. For each node type, make selections for **Child Option**:
   - **Any** - Any node types are allowed as children
   - **None** - No node types are allowed as children
   - **Selected** - The selected node type is the only type allowed for children. For **Child Node Types**, select node types for children, and then click **OK**.
5. Make selections for these optional settings:
   - **Allow Shared Nodes**: Configure shared nodes for the hierarchy set. Shared nodes allow the same nodes to exist multiple times in the hierarchy set.

     > **Note:**
     >
     > By default, shared nodes are allowed within a hierarchy set.

   - **Use Custom Order**: Select to allow nodes to be reordered in a hierarchy viewpoint. If this is not selected, nodes are sorted alphanumerically. If this is selected, nodes can be sorted using a custom order.
   - **Group Parent Nodes First**: Select to have parent nodes placed at the beginning of the siblings when there is a mix of parent and bottom nodes.
   - **Sort Bottom Nodes By**: When **Group Parent Nodes First** is enabled, select the sorting order (Alphanumeric or Custom) for the bottom nodes under the parents.

     > **Note:**
     >
     > This field is displayed only when **Group Parent Nodes First** is enabled.

The following table describes the way these settings work together to control the sort order of parent and bottom nodes:

Scenario	Use Custom Sort	Group Parent Nodes First	Sort Bottom Nodes By	Reorder Action
All nodes grouped together, sorted alphanumerically	False	False	(Not available)	Disabled on all nodes
All nodes grouped together, custom sort order	True	False	(Not available)	Enabled on all nodes
Group siblings by parent and bottom nodes, sort all alphanumerically	False	True	Alphanumeric	Disabled on all nodes
Group siblings by parent and bottom nodes, sort parent nodes alphanumerically and bottom nodes custom	False	True	Custom	Disabled on parent nodes. Enabled on bottom nodes.
Group siblings by parent and bottom nodes, sort parent nodes custom and bottom nodes alphanumerically	True	True	Alphanumeric	Enabled on parent nodes. Disabled on bottom nodes.
Group siblings by parent and bottom nodes, sort both custom	True	True	Custom	Enabled on all nodes.

6. Click **Save**.

Inspecting a Hierarchy Set

From the hierarchy set inspector, you can view and edit information about a hierarchy set. For more information, see Working with Hierarchy Sets.

Videos

Your Goal	Watch This Video
Learn about creating hierarchy sets.	▶ Creating Hierarchy Sets

To inspect a hierarchy set:

1. Click **Hierarchy Sets.**
2. For the hierarchy set that you want to inspect, click ⋮ in the **Actions** columns, and then select **Inspect**.
3. Select from the options available on these tabs:

- **General**: Displays information on the application, dimension, status, and binding status of the hierarchy set. See Binding Status.

  Click **Edit** to change the name, description, and status. See Understanding the Lifecycle of Data Objects and the Data Chain.
- **Definition**: Configure node types for this hierarchy set, including whether node types allow children and if so, what the allowed child node types are, whether shared nodes are allowed in the hierarchy set, and whether a custom order for nodes is allowed. See Inspecting a Node Type
- **Validations**: Add custom business logic to enforce data rules for the hierarchy set by creating custom validations. See Custom Validations
- **Data Chain**—Provides a clickable graphic so that you can see the node types, node sets, and viewpoints referenced by the hierarchy set.
- **Permissions**—Set up user and group permissions for the hierarchy set, see Working with Permissions.
- **Policies**—Configure an approval policy for the hierarchy set, see Configuring Policies.

Searching for Hierarchy Sets

When you perform a search in the Hierarchy Sets list, the system searches for your text in hierarchy set names and descriptions, application names, and dimension names. The names and descriptions that contain the search text are displayed in the results window in bold-faced font. You can also include archived hierarchy sets in your search.

To search in the Hierarchy Sets list:

1. From **Hierarchy Sets**, click 🔍.
2. (Optional) If you want your search to include archived hierarchy sets, click ▼ and then select **Show Archived**.
3. Enter a text string to search for. The search is performed automatically after you stop typing.

   To close the search, click ✕ .

Working with Node Sets

Node sets define the group of nodes available in viewpoints. A node set can include all the hierarchies in a hierarchy set or can exclude some hierarchies by only using a subset of top nodes. For example, to work with only the US branch of a geographical hierarchy, you would define a node set that includes only the US hierarchy.

> **Tip:**
>
> Top nodes for a node set can be defined by adding or inserting them into a viewpoint using a request, see Making Changes Using Requests.

Node sets can consist of either hierarchies or lists. For example, a node set could contain a list of country codes, or a hierarchy of cost center relationships. The data object in which a node set's node types are defined depends upon whether a node set is a list or hierarchy:

- If the node set represents a hierarchy, the node types are specified by the hierarchy set used by the node set.
- If the node set represents a list, the node types are specified by the node set.

The relationships between node sets and the other data objects depends upon whether the node set is for a list or a hierarchy:

- Hierarchy: The node set references one hierarchy set. The hierarchy set defines the node set's node types.
- List: The node set references one or more node types.
  Use multiple node types to create lists that contain nodes of multiple types. For example, suppose you have node types for Cost Centers and for Adjustment Cost Centers, and want to create a list that contains both types of cost centers. To do so, you would define a node set that references both of the node types.

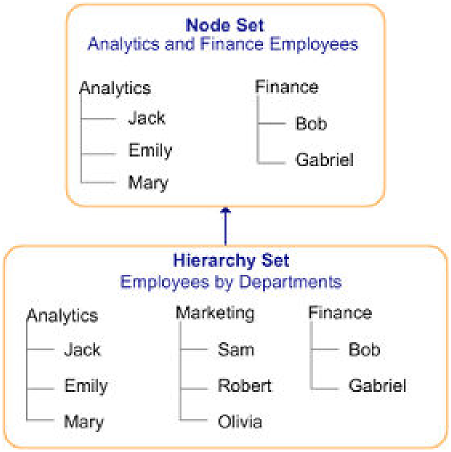
> **Note:**
>
> A node set can be used in more than one viewpoint.

For more information, see:

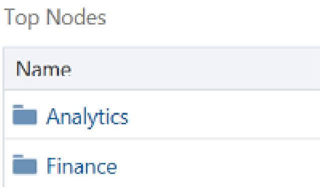
- Working with Data Objects and the Data Chain
- Inspecting a Node Set

Example 20-4 Node Set Example

The Data Chain for a Hierarchy Example uses a node set that includes the hierarchy set's Analytics and Finance hierarchies and excludes the Marketing hierarchy, as shown in the following diagram:



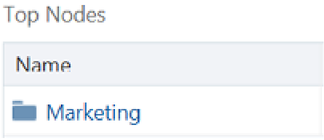
The node set defines the Analytics and Finance departments as its top nodes, as shown in the following example:



Top Nodes

Name
Analytics
Finance

Suppose you need to create a viewpoint that includes only the Marketing hierarchy. You would create a node set that uses the same hierarchy set as used in the previous example. The Marketing node would be the node set's only top node, as shown in the following example:



Top Nodes

Name
Marketing

Videos

Your Goal	Watch This Video
Learn about creating node sets.	Creating Node Sets

Creating Node Sets

Best Practice

It's a best practice to create a new node set when you want to:

- Point to a different hierarchy set for a hierarchy.
- Point to different node types for a list.
- Display viewpoints with different top nodes.

For more information, see Working with Node Sets.

Videos

Your Goal	Watch This Video
Learn about creating node sets.	Creating Node Sets

Considerations

- For unbound list based node sets:
  - **All unbound node sets except those in Users applications**: You can add Normal, Legacy GL, or Lookup class node types. All of the node types must be of the same class.
  - **Unbound node sets in Users applications only**: You can add both User and User Rollup class node types, but no other node type classes.
- For bound or partially bound list based node sets: You can add only Normal class node types.
- For lookup dimensions: You can add Legacy GL or Lookup class node types only. All of the node types must be of the same class.

To create a node set:

1. From **Node Sets**, click **Create**.
2. Select the application dimension, define the name and description for the node set, and then click **Create**.

   An inspector for the node set opens.

3. Select **General**, and from **Source Type** select whether you want the node set to display as a hierarchy or a list. If you select hierarchy, then you'll also need to select the Hierarchy Set to use.
4. Click **Save**.
5. Do one of the following:
   - If you are using a list, select **Definition**, click **Edit**, click **Manage**, and then select the node types to include in the node set. Click **Save**

     > **Note:**
     >
     > See **Considerations**, above, for the class of node types that you can select. For specialty node type classes, the class is displayed after the node type name in the node type selector.

   - If you are using a hierarchy, node types are already assigned based on the hierarchy set that you selected.
6. **Optional:** From **Definition**, click the name of a node type to edit properties for the node type.

Inspecting a Node Set

From the node set inspector, you can view and edit information about a node set. For more information, see Working with Node Sets.

Videos

Your Goal	Watch This Video
Learn about creating node sets.	▶ Creating Node Sets.

1. Click **Node Sets.**
2. For the node set that you want to inspect, click ⋮ in the **Actions** columns, and then select **Inspect**.
3. Select from the options available on these tabs:
   - **General**—Displays information on the application, dimension, status, and binding status of the node set. See Binding Status.

     Click **Edit** to change the name, description, and status. See Understanding the Lifecycle of Data Objects and the Data Chain.
   - **Definition**—View node types in a hierarchy or add node types to a list. To add a top node, see Adding a Top Node.
   - **Data Chain**—Provides a clickable graphic so that you can see the node types, hierarchy sets (if applicable), and viewpoints referenced by the node set.

Searching for Node Sets

When you perform a search in the Node Sets list, the system searches for your text in node set names and descriptions, application names, and dimension names. The names and descriptions that contain the search text are displayed in the results window in bold-faced font. You can also include archived node sets in your search.

To search in the Node Sets list:

1. From **Node Sets**, click 🔍.
2. (Optional) If you want your search to include archived node sets, click ▼ and then select **Show Archived**.
3. Enter a text string to search for. The search is performed automatically after you stop typing.

   To close the search, click ✕ .

Understanding Viewpoints

A viewpoint provides a user interface in which users can work with nodes. A viewpoint's available nodes and hierarchical relationships are defined by its data chain.

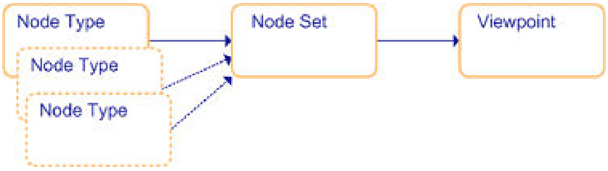
You can configure the following items in a viewpoint:

- The properties that can be edited.
- Actions that users can perform in the viewpoint. The following actions can be enabled:
  - Add
  - Insert
  - Move
  - Remove
  - Delete
  - Reorder
    When **Reorder** is selected, users can change the order of nodes. You can select **Reorder** only if the hierarchy set on the data chain is configured to allow a custom sort order.
- Labels for properties. For example, you might want a property named "Country" to display as "Country Name".

A viewpoint references one node set, and the objects on the viewpoint's data chain depend upon whether the viewpoint represents a list or a hierarchy.

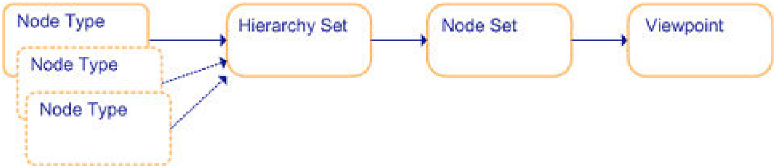
Data Chain for Lists

A viewpoint representing a list references one node set. The node set references one or more node types, as shown in the following diagram:



Data Chain for Hierarchies

A viewpoint representing a hierarchy references one node set. The node set references one hierarchy set, which in turn references one or more node types. The following diagram shows the data chain for a viewpoint that contains a hierarchy:

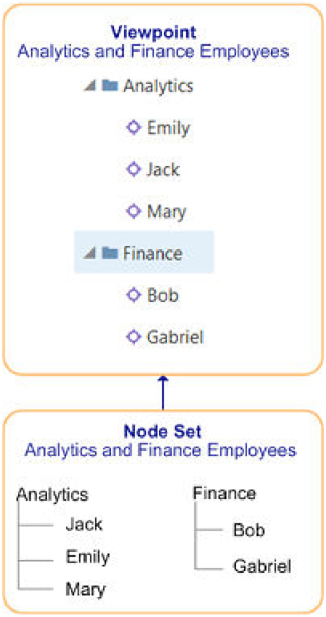


For more information, see:

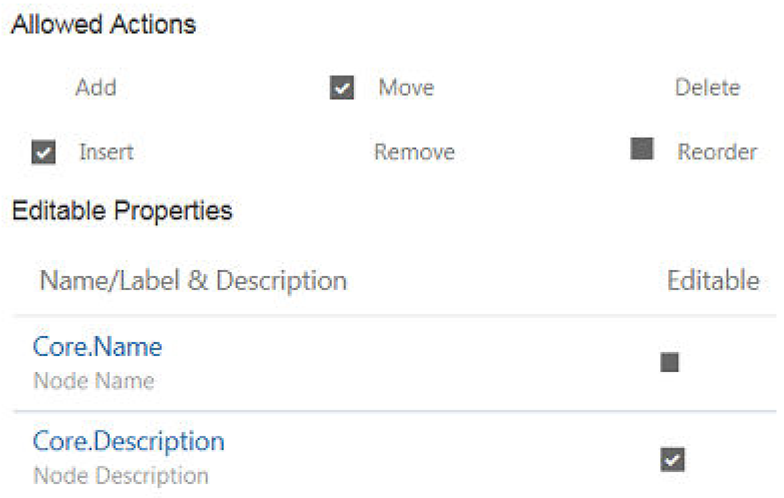
- Working with Data Objects and the Data Chain
- Working with Viewpoints

Example 20-5 Viewpoint Example

The viewpoint in the Data Chain for a Hierarchy Example uses a node set that includes the Analytics and Finance hierarchies, as shown in the following diagram:



You can specify the actions users can perform and the properties they can edit in the viewpoint. In the following example, users can only insert and move nodes, and can edit the Description property but not the Name property:



Allowed Actions

Add ☑ Move Delete

☑ Insert Remove ■ Reorder

Editable Properties

Name/Label & Description	Editable
Core.Name Node Name	■
Core.Description Node Description	☑

Videos

Your Goal	Watch This Video
Learn about creating viewpoints.	Creating Views and Viewpoints

Archiving, Unarchiving, and Deleting Viewpoints

You can archive an active viewpoint if you no longer need it. After a viewpoint is archived, you can return it to an Active status or you can delete it.

Considerations

- You must have *Owner* permission on a view in order to archive, unarchive, or delete a viewpoint in that view.
- A viewpoint must be archived before you can delete it.
- When you archive a viewpoint that is the source or target of a subscription:
  - If the viewpoint that you are archiving is the source viewpoint in a subscription, the system displays a warning before you archive it.
  - Subscriptions that use that viewpoint are automatically ignored and will not generate subscription requests. When you unarchive the viewpoint, subscriptions are no longer ignored and are returned to their previous state (Enabled or Disabled).
- In order for an archived viewpoint to be deleted, it cannot not have any of these dependencies:
  - It cannot be bound to a dimension.
  - It cannot have any active subscriptions (as either the source or target viewpoint), compare profiles, loads, or public extracts defined for it. Private extracts do not prevent a viewpoint from being deleted; they are deleted along with the viewpoint.
  - It cannot have any requests that are not in Completed status.

To archive or unarchive a viewpoint:

1. From the Viewpoints tab of the View inspector, click **Edit**.
2. Perform an action:

- To archive: In **Status**, select **Archived**, and click **Save**.
- To unarchive: In **Status**, select **Active**, and click **Save**.

To delete a viewpoint:

1. From the Viewpoints tab of the View inspector, click **Filter** ▽ and select **Show Archived** to display archived viewpoints.
2. Click **Edit**.
3. In the Actions column for an archived viewpoint, click **Action** ⋮, select **Delete**, and then click **Yes**.
   If the viewpoint still has an active dependency (see the Considerations section, above), a message is displayed that identifies the issue. Take the appropriate action to resolve the dependency (for example, delete extract or subscription definitions, delete or complete active requests, or bind the dimension to a different viewpoint) and try your action again.

After you delete a viewpoint, it is no longer displayed on the Views page or the Viewpoints tab of the View inspector. You can create a new viewpoint with the same name as the deleted viewpoint. Request and audit history will continue to display history for the deleted viewpoint.

Understanding Bindings and Bound Data Objects

Bindings are created between dimensions and viewpoints in an Oracle Fusion Cloud Enterprise Data Management application to ensure that the bound data objects conform to an external application's requirements. If a viewpoint and its related data chain objects are bound to a dimension, it means that those objects are used by the external application.

Changes made to the bound objects are imported from and exported back to the external application, with the exception of bound lookup dimensions. For example, if you add 10 new nodes to a bound viewpoint in Cloud EDM, the next time the dimension is exported, the 10 new nodes are part of the export to the external application.

There are three types of bindings in Cloud EDM:

- **Dimension binding**: Automatically created during application registration. When a dimension is registered for an application, Cloud EDM automatically creates one or more viewpoints (based on the application type and information provided during registration) and a series of related data chain objects. This viewpoint and the associated data chain objects are bound to the external application at the dimension level.

  > **Note:**
  >
  > – Oracle Financials Cloud General Ledger dimensions can be bound to multiple viewpoints to support multiple trees and tree versions. See Copying Bindings to Create Trees or Tree Versions.
  >
  > – All other application types support a single dimension binding. That is, for all application types except Oracle Financials Cloud General Ledger, you can change the viewpoint that a dimension is bound to, but you cannot bind a dimension to multiple viewpoints.

- **Map binding**: Manually created to enable you to map node types from dimensions in different source applications to a dimension in a target application. When you create a map binding, a mapping hierarchy set, node set, and viewpoint are also created. You can create only one map binding for a dimension, but you can map multiple source systems in your mapping. See Creating Mapping Viewpoints.

- **Lookup binding**: Manually created for lookup dimensions. When you create a lookup binding, the data chain objects needed to support the binding are also created if they do not already exist. This includes a viewpoint, hierarchy set (if needed), node set, and Lookup class node type. Lookup bindings are not used to import and export data. You can create multiple lookup bindings for a lookup dimension. See Creating a Lookup Binding.

The data imported and exported for a dimension is stored and displayed by the data objects bound to the dimension. If a dimension is related to other data objects, those objects are unbound. Only data contained by the dimension's bound viewpoints is imported and exported.

The following list describes considerations for bindings and bound data objects:

- To ensure that a dimension's data conforms to the requirements of the external application, some binding settings cannot be overridden by settings on related data objects. For example, if a binding does not enable shared nodes, the hierarchy set bound to the dimension cannot enable shared nodes.
  Binding rules protect such dimension settings from being overridden, see Understanding Binding Rules.
- You can change the viewpoint and related data objects bound to a dimension, see Binding a Viewpoint to a Dimension in Planning and FreeForm Applications or, for Universal applications, see Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications.
- You cannot manually delete bindings that were created as part of the application registration process (such as dimension bindings). Bindings that were created manually, such as map bindings and lookup bindings, can be deleted.

> **Note:**
>
> Only data objects bound to a dimension must conform with the binding rules. Unbound data objects enable you to model and evaluate potential changes to dimensions in cases where the changes would not conform with a dimension's current binding rules.

Binding Status

The binding status of a data chain object identifies whether its data is used by an external application. Data objects may be bound because they are used in a binding themselves, or because they contain a data chain object that is bound. For example, if a viewpoint that is not used in a binding contains a node set that is bound to a dimension, the viewpoint has a status of **Node Set Bound**. To ensure data integrity during import and export operations, predefined validation rules are run against data objects with any binding status except **Unbound**. See Understanding Validations and Constraints.

The following table describes the binding status for data objects.

Table 20-1 Binding Statuses for Data Objects

Binding Status	Viewpoint	Node Set	Hierarchy Set	Node Types
Bound	The viewpoint is used in a binding.	The node set is used by a bound viewpoint.	The hierarchy set is used by a bound viewpoint.	The node types are used by a bound viewpoint.
Node Set Bound	The viewpoint itself is not bound, but it uses a bound node set.	The node set is used by a bound viewpoint	Not applicable for hierarchy sets	Not applicable for node types
Hierarchy Set Bound	The viewpoint itself is not bound, but it uses a bound hierarchy set.	The node set is not bound, but it uses a bound hierarchy set.	The hierarchy set is used by a bound viewpoint.	Not applicable for node types
Node Type Bound	The viewpoint itself is not bound, but it uses a bound node type.	The node set is not bound, but it uses a bound node type.	The hierarchy set is not bound, but it uses a bound node type.	May contain bound and unbound node types.
Mapping Bound	The viewpoint is used in a map binding. It can be used in extracts that contain specialty node types such as Lookup and Legacy GL class node types.	The node set is used in a map binding.	The hierarchy set is used in a map binding.	May contain bound and unbound node types.
Unbound	The viewpoint is not used in a binding, and all of the data chain objects that it contains have a binding status of Unbound.	The node set is not used in a binding, and all of the data chain objects that it contains have a binding status of Unbound.	The hierarchy set is not used in a binding, and all of the data chain objects that it contains have a binding status of Unbound.	The node type has a binding status of Unbound.

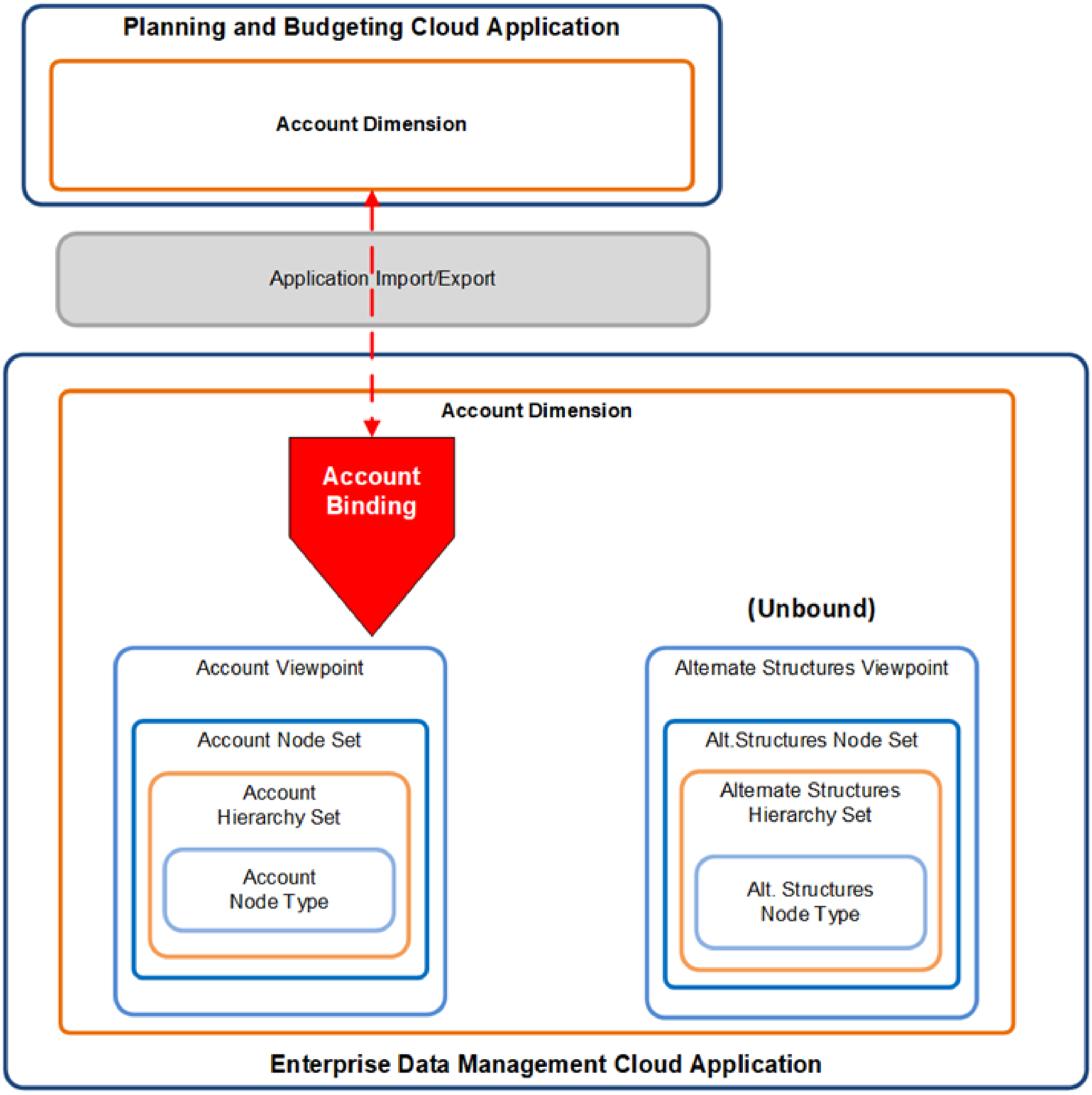
For more information, see:

- Understanding Dimensions
- Identifying Objects Bound to Dimensions
- Understanding Applications and Sharing Data

Dimension Binding Examples

Binding Examples

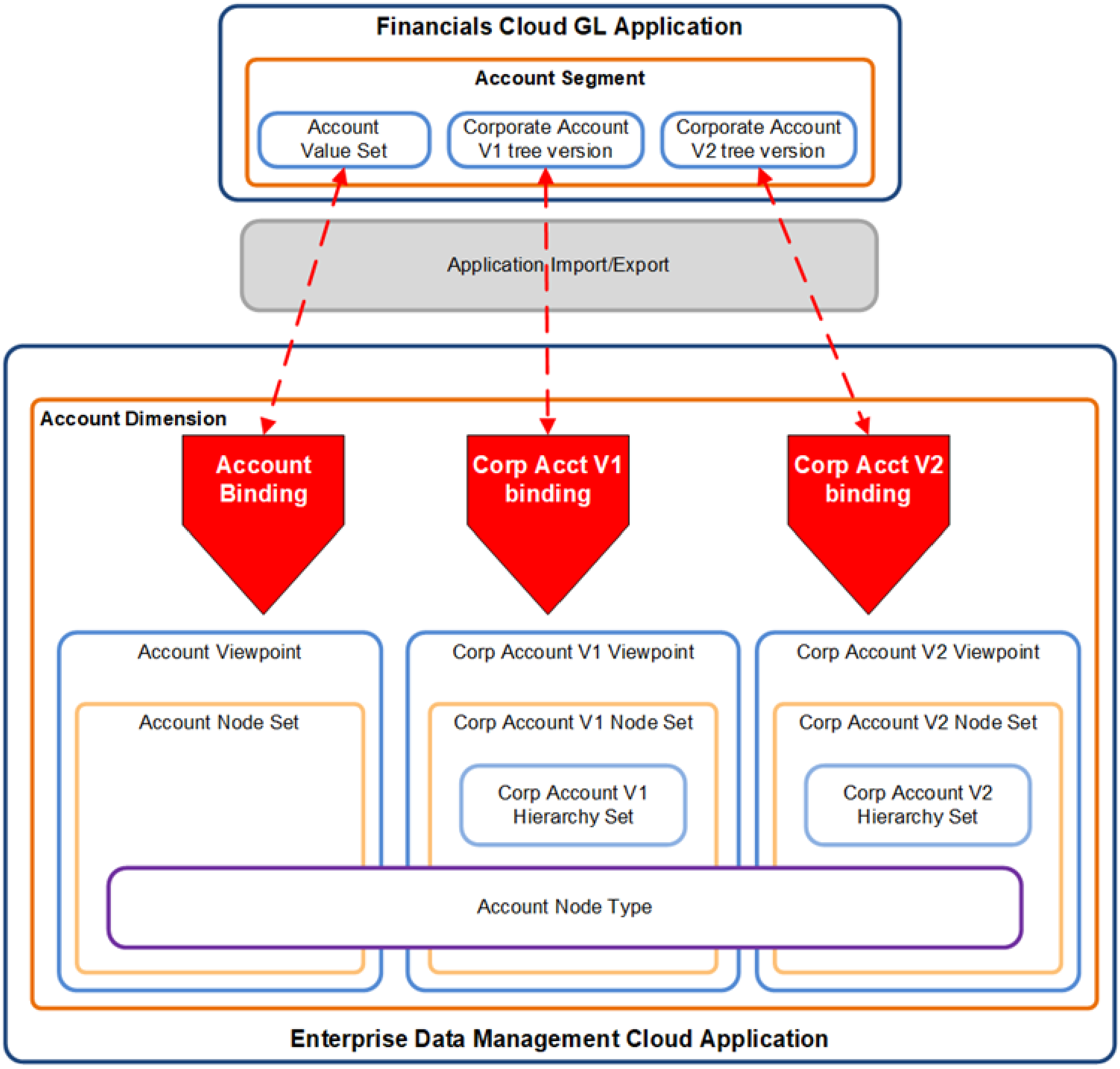
Let's look at a binding for a Planning application to understand some binding principles:


Planning and Budgeting Cloud Application
Account Dimension
Application Import/Export
Account Dimension
Account Binding
(Unbound)
Account Viewpoint
Account Node Set
Account Hierarchy Set
Account Node Type
Alternate Structures Viewpoint
Alt.Structures Node Set
Alternate Structures Hierarchy Set
Alt. Structures Node Type
Enterprise Data Management Cloud Application

Note the following about bindings in this diagram:

- In Planning applications, dimensions can be bound only to a single viewpoint. In this example, the **Account** dimension is bound to the Account viewpoint, while the Alternate Structures viewpoint is unbound. Since data is imported and exported only to bound viewpoints, import and export operations will not affect the Alternate Structures viewpoint. See Importing Dimensions.
- The Account binding is between the Account dimension and the Account viewpoint in Oracle Fusion Cloud Enterprise Data Management. The binding controls the import and export to the Account dimension in the external application, as well as enforcing binding rules within Cloud EDM.
- All of the data chain objects contained within the Account viewpoint are also bound to the Account dimension.

Now let's look at a binding for an Oracle Financials Cloud General Ledger application:


Financials Cloud GL Application
Account Segment
Account Value Set
Corporate Account V1 tree version
Corporate Account V2 tree version
Application Import/Export
Account Dimension
Account Binding
Corp Acct V1 binding
Corp Acct V2 binding
Account Viewpoint
Corp Account V1 Viewpoint
Corp Account V2 Viewpoint
Account Node Set
Corp Account V1 Node Set
Corp Account V2 Node Set
Corp Account V1 Hierarchy Set
Corp Account V2 Hierarchy Set
Account Node Type
Enterprise Data Management Cloud Application

Oracle Financials Cloud General Ledger applications support binding a dimension to multiple viewpoints. In this example, the Account dimension in Cloud EDM is bound to three viewpoints. Each binding controls the import and export to a specific data structure in the external application and enforces binding rules within Cloud EDM:

- **Account Viewpoint binding**— controls the import and export to the Account value set in the external application.
- **Corporate Account V1 viewpoint binding**— controls the import and export to the Corporate Account V1 tree version in the external application.
- **Corporate Account V2 viewpoint binding**— controls the import and export to the Corporate Account V2 tree version in the external application.

See Working with Oracle Financials Cloud General Ledger Applications for more information about the Oracle Financials Cloud General Ledger information model.

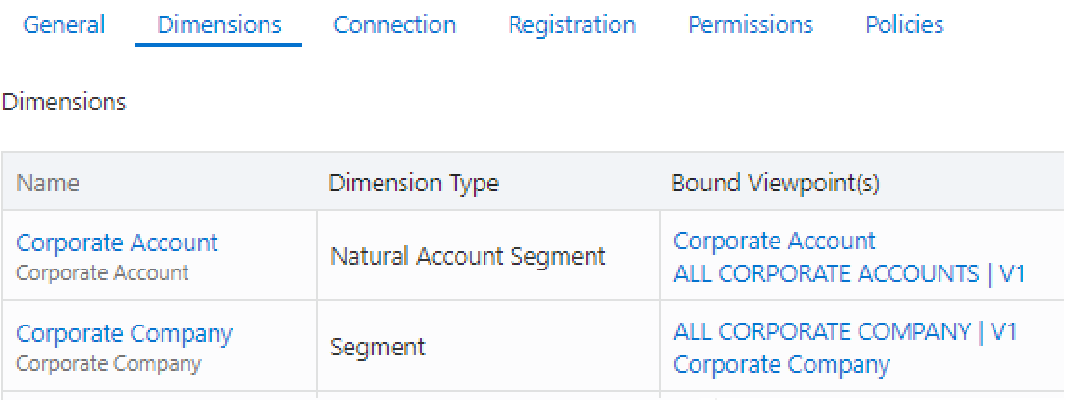
Identifying Objects Bound to Dimensions

This topic describes the various ways to identify objects that are bound to dimensions.

Identifying a Dimension's Bound Viewpoints

To find the viewpoints bound to a dimension, take the following steps:

1. Click Applications.
2. In the application's **Actions** column, click ⋮, and then select **Inspect**.
3. Click **Dimensions**.
   The Dimensions tab displays the registered application's dimensions and the viewpoints that the dimensions are bound to. In the following example, the Corporate Account dimension is bound to the Corporate Account and ALL CORPORATE ACCOUNTS| V1 viewpoints, and the Corporate Company dimension is bound to the ALL CORPORATE COMPANY | V1 and Corporate Company viewpoints.



Dimensions

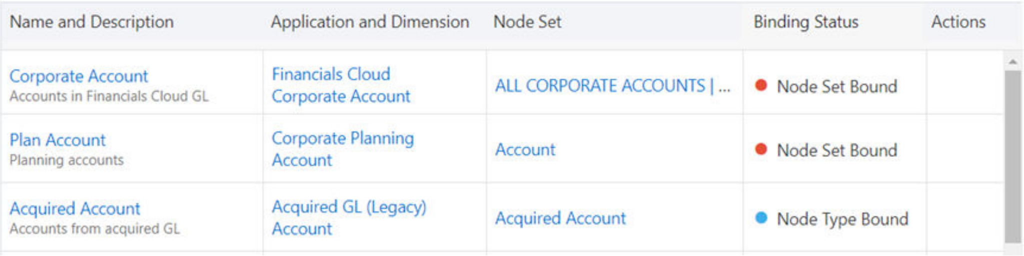
Name	Dimension Type	Bound Viewpoint(s)
Corporate Account Corporate Account	Natural Account Segment	Corporate Account ALL CORPORATE ACCOUNTS | V1
Corporate Company Corporate Company	Segment	ALL CORPORATE COMPANY | V1 Corporate Company

You can inspect the bindings and change the viewpoint that a dimension is bound to. See Inspecting and Editing a Binding.

Identifying Bound and Unbound Viewpoints in a View

To identify the bound and unbound viewpoints contained by a view, take the following steps:

1. Click **Views**.
2. In the view's **Actions** column, click ⋮, then select **Inspect**.
3. Click **Viewpoints**.
   The Viewpoints tab displays the following information:
   - The registered application's viewpoints.
   - Each viewpoint's dimension and application.
   - The **Binding Status** column indicates the binding status of the viewpoints. See Binding Status.
     The following example shows three viewpoints in an Account Maintenance view. The Corporate Account viewpoint contains a node set that is bound to the Corporate Account dimension, the Plan Account viewpoint contains a node set that is bound to the Account dimension, and the Acquired Account viewpoint contains a node type that is bound to the Account dimension.



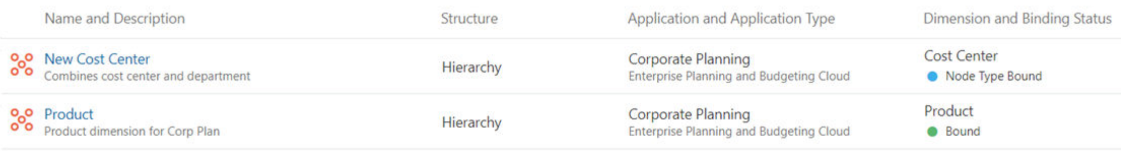
Name and Description	Application and Dimension	Node Set	Binding Status	Actions
Corporate Account Accounts in Financials Cloud GL	Financials Cloud Corporate Account	ALL CORPORATE ACCOUNTS | ...	● Node Set Bound	
Plan Account Planning accounts	Corporate Planning Account	Account	● Node Set Bound	
Acquired Account Accounts from acquired GL	Acquired GL (Legacy) Account	Acquired Account	● Node Type Bound	

Identifying Bound Data Objects

The Node Type, Hierarchy Set, and Node Set pages indicate the binding status of those data chain objects. Each page contains a **Dimension and Binding Status** column that indicates the binding status of the data chain objects.

For more information, see Understanding Bindings and Bound Data Objects.

In the following example, there are two nodes sets for the Corporate Planning application. The New Cost Center node set contains a node type that is bound to the Cost Center dimension and the Product node set is bound to the Product dimension:



Name and Description	Structure	Application and Application Type	Dimension and Binding Status
New Cost Center Combines cost center and department	Hierarchy	Corporate Planning Enterprise Planning and Budgeting Cloud	Cost Center ● Node Type Bound
Product Product dimension for Corp Plan	Hierarchy	Corporate Planning Enterprise Planning and Budgeting Cloud	Product ● Bound

Creating Bindings

While dimension bindings are automatically created during application registration, you create map and lookup bindings manually in the Dimension inspector. For lookup dimensions, you create lookup bindings only. For other dimension types, you create map bindings.

Related links:

- Creating a Map Binding
- Creating a Lookup Binding

Creating a Map Binding

A map binding allows node types to be mapped from dimensions in different source applications to a dimension in a target application.

Creating Map Bindings

To create a map binding:

1. Inspect the application that contains your target dimension. See Inspecting Applications.
2. On the Dimensions tab, click the target dimension.
3. On the Bindings tab, click **Create Binding**.

> **Note:**
>
> The **Create Binding** button is not visible if a map binding already exists.

4. Enter a name, a description (optional), and at least one target node type for the map binding, and then click **Create**.

   > **Note:**
   >
   > You can add all of the same class node types, or a mix of normal and one specialty class (Legacy GL or Lookup) node types. You cannot add both specialty types together.

   The map binding is created, along with these data chain objects:

   - A new hierarchy set is created, and the target node types that you identified when you created the map binding are added to the hierarchy set.
   - A new node set is created and pointed to the new hierarchy set.
   - A new mapping viewpoint is created in the default view for the target application, and it is set to use the new node set.

> **Note:**
>
> You can add additional source systems to a map binding without having to recreate the map binding. Edit the hierarchy set for the map binding to include the new source node types, and then add the new mapping keys for the new source node types. See Mapping Source Nodes to Target Nodes.

Defining Mapping Keys

You create mapping keys to specify the source node types mapped to target node types and to define a location name to export the mapping data. These are considerations:

- Mapping keys are used for mapping exports for Enterprise Performance Management applications only. See Exporting Mapping Data.
- Define one mapping key, one for each source node type mapped to a target node type. For example, if you map two source applications to one target application, define two mapping keys defining the mapping relationship from each source node type to the target node type. A source node type can be mapped to one target node type only.
- Each mapping key is identified by the unique location name you enter. The location name is used to export the mapping data and by the consuming or external application to import the mapping data. For example, use the location name in Data Management to import mappings in Planning.

Before you define the mapping key, set up the maintenance view for mapping. See Creating Mapping Viewpoints.

To define a mapping key:

1. From **Applications**, find the application containing the target dimension.

2. In the application's **Action** column, click ⋮
3. Click **Inspect**, and from the **Dimensions** tab, select the target dimension.
   The dimension's inspector dialog displays.
4. On the Bindings tab, select the mapping binding or click **Create Map Binding** to create one.

   > **Note:**
   >
   > The **Create Map Binding** button is not visible if a mapping binding already exists.

5. In the binding inspector, click **Mapping**, and then click **Edit**.
   The Mapping tab displays all mapping viewpoints that meet this criteria:
   - The viewpoint is active.
   - The viewpoint uses the same dimension and application you are inspecting.
   - The viewpoint is not bound to the target dimension.
   - The viewpoint is in the same view as the viewpoint bound to the target dimension.
6. Select the mapping viewpoint.
7. Click **Create**, and then enter:
   - Location name to be used when exporting mappings to an external application.

     > **Note:**
     >
     > For Planning and Financial Consolidation and Close applications, use the Location name to import the mapping data using Data Management.

   - Source node type (can be Normal, Lookup, or Legacy GL class)
   - Target node type (can be Normal, Lookup, or Legacy GL class)

> **Note:**
>
> You do not need to set the Default check box; it is not used at this time.

Creating a Lookup Binding

You can create lookup bindings for lookup dimensions only. A lookup dimension can have multiple lookup bindings, and they can have the node or hierarchy type.

Creating Lookup Bindings

To create a lookup binding:

1. Inspect the application that contains your target dimension. See Inspecting Applications.

2. On the Dimensions tab, click the target dimension. For lookup bindings, you must pick a lookup dimension.
3. On the Bindings tab, click **Create Binding**.
4. Select the binding type (node or hierarchy).
5. Enter a name and (optionally) a description.
6. Select the node type class (Lookup or Legacy GL) for the binding.
7. Select a view to contain the viewpoint for the new binding.

   > **Note:**
   >
   > If you are not a Service Administrator, you must have the *Owner* permission on a view in order to create a binding in that view.

8. Click **Create**.
   The lookup binding is created, along with these data chain objects if they do not already exist:
   - A new node type of the class that you selected is created.
   - If the binding type is Hierarchy, a new hierarchy set is created and the node type is added to it.
   - A new node set is created and pointed to either the node type (if the binding type is Node) or the new hierarchy set (if the binding type is Hierarchy).
   - A new viewpoint is created in the view that you selected and connected to the new node set.

Inspecting and Editing a Binding

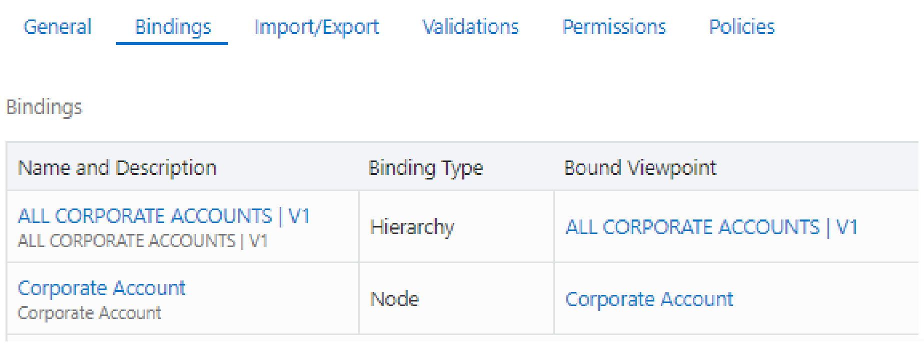
You can inspect a dimension's bindings, and you can edit the binding to change the viewpoint the dimension is bound to in that binding. Depending on your application type, a dimension can be bound to a single viewpoint (for example, in Planning applications) or multiple viewpoints (for example, in Oracle Financials Cloud General Ledger applications.)

To inspect the bindings for a dimension and change the viewpoint that a dimension is bound to:

1. Click Applications.
2. In the application's **Actions** column, click ⋮, and then select **Inspect**.
3. Click **Dimensions**.
   The Dimensions tab displays the registered application's dimensions and the viewpoints that the dimensions are bound to.
4. Click the dimension that you want to view the bindings for.
   The General tab of the **Dimension Inspector** dialog is displayed.
5. Click **Bindings**.
   The bindings for the dimension are displayed, along with the binding type (Hierarchy or Node) and the bound viewpoints. In the following example, the ALL CORPORATE ACCOUNTS | V1 dimension has a hierarchy-type binding to the ALL CORPORATE ACCOUNTS | V1 viewpoint and a node-type binding to the Corporate Account viewpoint.

Note:

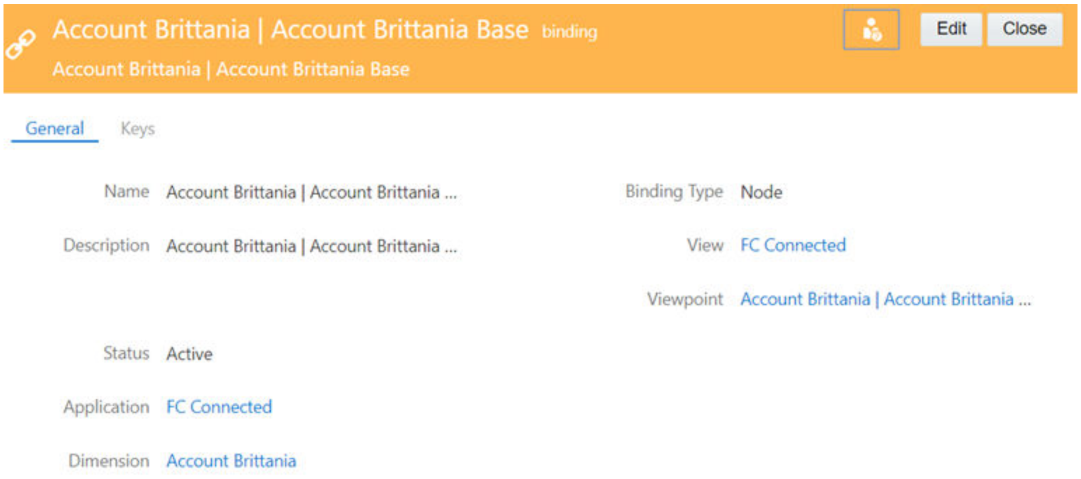
Node-type bindings represent viewpoints with a flat list of nodes that use list node sets. Hierarchy-type bindings represent viewpoints with a structured set of nodes that use hierarchy node sets.



Bindings

Name and Description	Binding Type	Bound Viewpoint
ALL CORPORATE ACCOUNTS | V1 ALL CORPORATE ACCOUNTS | V1	Hierarchy	ALL CORPORATE ACCOUNTS | V1
Corporate Account Corporate Account	Node	Corporate Account

6. Click the name of the binding that you want to inspect.
   The General tab of the **Binding Inspector** displays the following information about a binding:
   - Name and Description
   - Status (Draft, Active, or Archived)
   - The application that contains the binding
   - The dimension that the binding is attached to
   - The binding type (Node or Hierarchy)
   - The viewpoint that the dimension is bound to, as well as the view that contains the viewpoint



Account Brittania | Account Brittania Base binding
Account Brittania | Account Brittania Base

Name Account Brittania | Account Brittania ...
Description Account Brittania | Account Brittania ...
Status Active
Application FC Connected
Dimension Account Brittania

Binding Type Node
View FC Connected
Viewpoint Account Brittania | Account Brittania ...

7. **Optional:** To change the viewpoint that the dimension is bound to, perform these actions:
   a. From the Binding Inspector, click **Edit**.

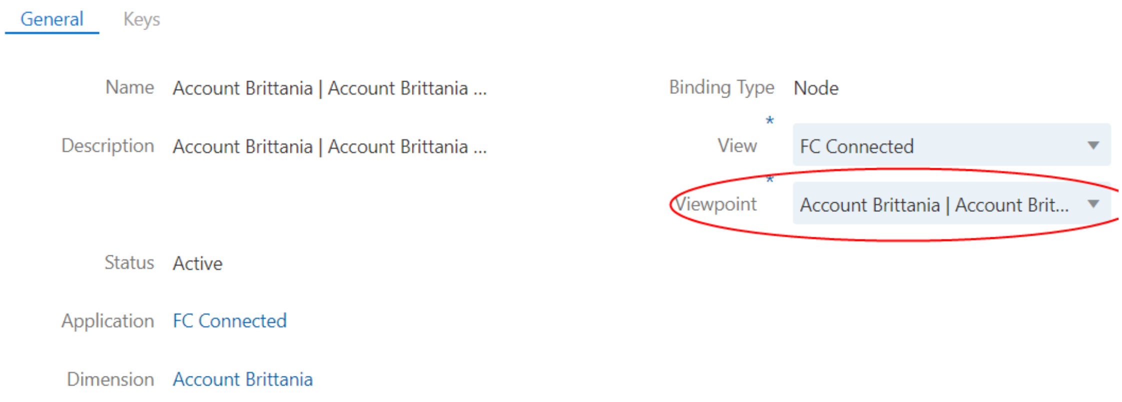
b. In **View**, select the view that contains the viewpoint that you want to bind the dimension to.

c. In **Viewpoint**, select the viewpoint that you want to bind the dimension to.

d. Click **Save**.
predefined binding rules are enforced when you click **Save**. For more information, see these topics:

   - Binding Rules for Planning and FreeForm Applications
   - Binding Rules for Oracle Financials Cloud General Ledger Applications
   - Binding Rules for E-Business Suite General Ledger Applications
   - Binding Rules for Financial Consolidation and Close Applications
   - Binding Rules for Universal Applications

> **Note:**
>
> You cannot change the binding to point to a viewpoint of a different type. That is, if the binding is pointing to a Node-type viewpoint, you cannot switch the binding to a Hierarchy-type viewpoint.



Name Account Brittania | Account Brittania ...

Description Account Brittania | Account Brittania ...

Status Active

Application FC Connected

Dimension Account Brittania

Binding Type Node

* View FC Connected

* Viewpoint Account Brittania | Account Brit...

Editing Binding Settings

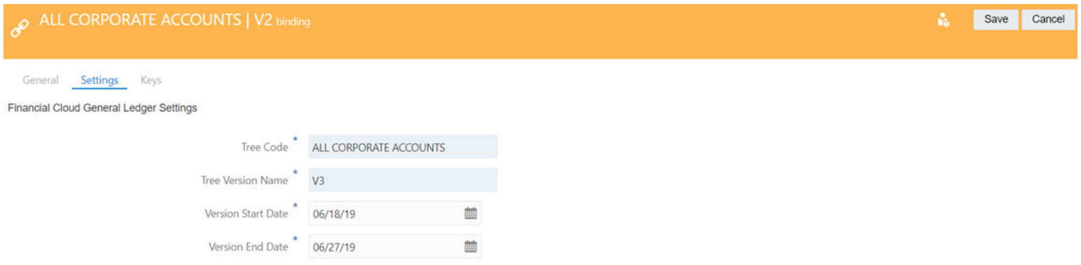
For hierarchy-type bindings in Oracle Financials Cloud General Ledger dimensions only, you can edit the binding settings to change the tree code and tree version that the binding points to.

To edit the binding settings:

1. Inspect the application that contains the binding that you want to edit.
2. On the **Dimensions** tab, click the dimension that you want to edit the binding for.
The dimension is displayed in the dimension inspector.
3. On the **Bindings** tab, click the binding that you want to edit the settings for. You can edit the settings for hierarchy-type bindings only.
4. In the binding inspector, click **Settings**.

> **Note:**
>
> The Settings tab is enabled only for hierarchy-type bindings in Oracle Financials Cloud General Ledger dimensions.

5. Click **Edit**.
6. Enter a new tree code, tree version name, and tree version start and end dates for the binding, and then click **Save**.



Financial Cloud General Ledger Settings

Tree Code * ALL CORPORATE ACCOUNTS

Tree Version Name * V3

Version Start Date * 06/18/19

Version End Date * 06/27/19

The service validates the binding settings and then updates the binding with the new settings.

Editing Binding Keys

Binding keys are created during application registration to control the binding rules and the import and export constraints on a binding for a dimension. You can edit the binding keys to change the direction (for example, to specify that a binding key is valid for exports only) and the column order in your export file.

> **Note:**
>
> The binding keys for Oracle Financials Cloud General Ledger applications are view-only. You cannot edit the column order or direction in the export file for these applications.

To edit the binding keys:

1. Inspect the binding that you want to edit the keys for. See Inspecting and Editing a Binding.
2. Click the Keys tab. The following information is displayed for the binding keys:

Field	Definition
Header	The column headers in the import and export files for all application types except for Oracle Financials Cloud General Ledger applications (which do not have column headers in their import and export files).
Key	Determines what type of information the binding key contains. For example, the `PARENT` key contains the parent for a node, and the `VALIDCONSOL` key contains information on whether or not the node is valid for consolidations.

Field	Definition
Sub Key	Works with the Key to determine the type of information the binding key contains. For example, for the key `ALIAS`, the sub key `DEFAULT` contains the default alias table for a node, and the sub key `ENGLISH` contains the English alias table.
Key Type	The usage for the key: • `Property`: contains information about a node property. • `Constant`: contains a value for a constant. • `Editable Constant`: contains a value for a constant in an Oracle Financials Cloud General Ledger application that can be edited during application registration or on the Settings tab of the binding inspector. See *Editing Binding Settings* in Inspecting and Editing a Binding. • `Node Type`: identifies the column for the node type and the parent node type in a Universal application if you have selected **Use Node Type Column** or **Use Parent Node Type Column**. See User Defined Dimension Import and Export Settings.
Source Value	• For property key types, contains the fully-qualified property name. • For constant and editable constant key types, contains the constant value.
Direction	Determines the importing and exporting of the binding key: • `Import`: binding key is not processed during exports. • `Export`: binding key is not processed during imports. • `Both`: binding key is processed during imports and exports. • `None`: binding key is not processed during either imports or exports. Select this option if a bound property in Oracle Fusion Cloud Enterprise Data Management is not needed in your external application. **Note:** Constant and Editable Constant binding keys can be set to `Export` or `None` only.
Action	Enables you to move the column order for the key up or down in the export file, or to delete the key for constant columns in Universal applications.

3. **Optional**: To change the direction for a binding key:
   a. Click **Edit**.
   b. In the **Direction** column for a binding key, select whether the binding key should be processed for exports, imports, both, or neither.
4. **Optional**: To change the column order for your export file:
   a. Click **Edit**.
   b. In the **Actions** column, click ⋮, and then select an option to move the binding key up, down, first, or last in the file.
5. **Optional**: To delete a binding key:
   a. Click **Edit**.

   b. In the **Actions** column, click ⋮, and then select **Delete**.

6. Click **Save**.

Managing Constant Keys for Universal Applications

For Universal applications only, you can create, edit, move, and delete constant keys:

1. Inspect the binding that you want to manage a constant key for.
2. Click **Edit**.
3. Perform an action:
   - To add a constant key, click **Create**, and then enter a header, source value, and direction.
   - To edit a constant key, modify the header, source value, or direction of a constant key.
   - To move a constant key, in the **Actions** column click ⋮, and then select an option to move the column up or down in the export file.
   - To delete a constant key, in the **Actions** column click ⋮, and then select **Delete**.
4. Click **Save**.

> **Note:**
>
> By default, when you create a constant key for a Universal application it is sorted to the end of the export file.

Deleting a Binding

You can delete bindings that you created, such as map bindings or lookup bindings. You cannot delete bindings that were created as part of the application registration process.

You must have at least *Metadata Manager* or *Owner* permission on the dimension in order to delete a binding.

When you delete a binding, the binding status is updated for the data chain objects that are associated with the binding but the objects themselves are not deleted. These objects can be separately archived and deleted if you no longer need them.

To delete a binding:

1. Inspect the dimension that contains the binding that you want to delete. See Inspecting Dimensions.
2. On the Bindings tab, click **Edit**, and then click the **Action** menu ⋮ next to the binding that you want to delete.
3. Click **Delete**, and then confirm that you want to delete the binding and click **Save**.

Understanding Binding Rules

Binding rules ensure that users cannot make changes that would prevent an application's data from being imported or exported. In other words, binding rules ensure that a registered application conforms to the external application's requirements. Binding rules vary by application type. For example, Oracle Financials Cloud General Ledger applications support multiple bindings per dimension, while Planning applications support a single binding per dimension.

Binding rules are enforced on the viewpoint and related data objects that are bound to a dimension. When you register an application, Oracle Fusion Cloud Enterprise Data Management creates the bound viewpoint and related data chain objects. The bound data objects store and display the data that is managed, imported, and exported, see Understanding Bindings and Bound Data Objects.

> **Tip:**
>
> If you change the bound viewpoint, the dimension will be bound to the data objects on the viewpoint's data chain.

Binding rules are enforced when users perform the following tasks:

- Updating a bound data object. If a change would violate a binding rule, then Cloud EDM prevents the change from being made.
- Updating the viewpoint bound to a dimension.

For example, suppose a dimension does not allow shared nodes. A binding rule prevents users from enabling shared nodes in the hierarchy set bound to the dimension.

> **Note:**
>
> Only data objects bound to a dimension must conform with the binding rules. Unbound data objects enable you to model and evaluate potential changes to dimensions in cases where the changes would not conform with a dimension's current binding rules.

There are basic binding rules that apply to all application types. Some application types have additional binding rules.

For more information, see:

- Binding Rules for All Application Types
- Binding Rules for Universal Applications
- Binding Rules for Planning and FreeForm Applications for Planning.
- Binding Rules for Oracle Financials Cloud General Ledger Applications
- Binding Rules for E-Business Suite General Ledger Applications
- Binding Rules for Financial Consolidation and Close Applications

Binding Rules for All Application Types

The following table lists the binding rules that apply to all application types.

Bound Object	Binding Rules
Viewpoint	• A property required by the binding cannot be hidden or read-only. • The viewpoint cannot be archived.
Node set	• If the binding specifies the nodes are a hierarchy, the node set cannot specify the nodes are a list. • If the binding specifies the nodes are a list, the node set cannot specify the nodes are a hierarchy. • The node set cannot be archived.
Hierarchy set	• If a binding does not allow shared nodes, the hierarchy set cannot be configured to allow shared nodes. • The hierarchy set cannot be archived.
Node type	The node type cannot be archived.

For more information, see:

- Understanding Binding Rules
- Binding Rules for Universal Applications

Understanding the Lifecycle of Data Objects and the Data Chain

Applications, dimensions, views, and data chain objects exist in one of three statuses: Draft, Active, and Archived.

- **Draft**—Applications and most data chain objects are set to Draft status when you create them. You can't assign draft objects to other objects in the system.

  > **Note:**
  >
  > Views and node types do not use Draft status. They are set to Active status as soon as you create them.

  You can:

  – Edit draft objects with the exception of Application and Status

  – Delete draft objects

- **Active**—In Active status, objects are available for you to use in the system. You can edit active data chain objects with the exception of Application and Dimension. Objects in an Active status can be archived but can't be deleted. You can change the status from Active to Archived only if dependent objects are set to Archived status.

  > **Caution:**
  >
  > When you archive data chain objects, they must be archived in this order: viewpoint, node set, hierarchy set (if used), and then node type.

  For example, you have Data Chain ABCD which uses Node Type A, Hierarchy Set B, Node Set C, and Viewpoint D which are all in Active status. If you want to set Hierarchy

Set B to Archived, you must first set Viewpoint D to Archived, then set Node Set C to Archived, and then Hierarchy Set B to Archived. Node Type A is free to remain in Active status.

- **Archived**—Objects set to Archived status can't be used by other objects in the system. You can edit Name, Description, and Owner only. You can access archived objects in the list for the object.

> **Caution:**
>
> When you unarchive data chain objects, they must be unarchived in this order: node type, hierarchy set (if used), node set, and then viewpoint.

> **Note:**
>
> Only archived applications can be deleted. Dimensions, views, and data chain objects cannot be deleted.

The following table shows lifecycle status for applications, dimensions, views, and data chain objects.

Object	Draft	Active	Archived
Application	Applications start in draft status. Draft applications can be deleted. After you register an application, the status changes to Active.	An application is in Active status after you register it. To archive an application, change its status to Archived in the application's inspector.	Archived applications can be inspected in the Applications list. They can also be unarchived or deleted. To unarchive an application, change its status to Active in the application's inspector.
Dimension	Not Applicable	A dimension is in Active status after you register an application. To archive a dimension, remove the dimension from the application's registration.	You can see dimensions that have been archived for an application by viewing them on the Application inspector **Dimension** tab. Archived dimensions can be unarchived. To unarchive a dimension, add the dimension to the application's registration. The dimension must use the same name as previously used.

Object	Draft	Active	Archived
View	Not Applicable	Newly created views start in Active status. To archive a view, archive all of its active viewpoints, delete any draft viewpoints, and then change its status to Archived in the view's inspector. You can delete draft views and active views which contain no viewpoints.	Archived views can be inspected in the Views list. You cannot unarchive a view.
Viewpoint	Viewpoints start in draft status. You move a viewpoint to the Active status by assigning a node set and saving.	A viewpoint is in Active status after you assign a node set to it. To archive a viewpoint, change its status to Archived in the viewpoint's inspector. You cannot archive a viewpoint that is bound to a dimension. If you want to archive the viewpoint bound to a dimension, you must first change the dimension's bound viewpoint.	Archived viewpoints can be inspected from the view that used the viewpoint. They can also be unarchived. To unarchive a viewpoint, from the view's inspector, select the Definition tab, click the archived viewpoint, and then change its status to Active.
Node Type	Not Applicable	Newly created node types start in Active status. To archive a node type, change its status to Archived in the node type's inspector.	Archived node types can be inspected in the Node Types list. They can also be unarchived. To unarchive a node type, change its status to Active in the node type's inspector.
Hierarchy Set	Hierarchy sets start in draft status. Draft hierarchy sets can be deleted. You move a hierarchy set to the Active status by assigning a node type and saving.	After a node type is selected, a hierarchy set moves to Active status. To archive a hierarchy set, change its status to Archived in the hierarchy set's inspector.	Archived hierarchy sets can be inspected in the Hierarchy Sets list. They can also be unarchived. To unarchive a hierarchy set, change its status to Active in the hierarchy set's inspector.

Object	Draft	Active	Archived
Node Set	Node sets start in draft status. Draft node sets can be deleted. You move a node set to the Active status by assigning a hierarchy set and saving.	After a hierarchy set is assigned, a node set moves to Active status. To archive a node set, change its status to Archived in the node set's inspector.	Archived node sets can be inspected in the Node Sets list. They can also be unarchived. To unarchive a node set, change its status to Active in the node set's inspector.
Binding	Not Applicable	After a dimension is created (in the application registration or modification process), the binding status is set to Active status When you archive a dimension, the associated binding is automatically archived.	Archived bindings can be viewed in the dimension inspector. When you unarchive a dimension, the associated binding is automatically unarchived.

Archiving and Unarchiving Order of Data Chain Objects

When you archive data chain objects, they must be archived in this order: viewpoint, node set, hierarchy set (if used), and then node type.

When you unarchive data chain objects, they must be unarchived in this order: node type, hierarchy set (if used), node set, and then viewpoint.

For more information, see:

- Working with Data Objects and the Data Chain
- Archiving, Unarchiving, and Deleting Applications
- Archiving and Unarchiving Dimensions
- Archiving and Deleting Views
- Archiving and Unarchiving Data Chain Objects

Archiving and Unarchiving Data Chain Objects

You can archive active data chain objects (viewpoints, node sets, hierarchy sets, and node types) when you no longer need them. You can also return archived data chain objects to Active status.

Considerations

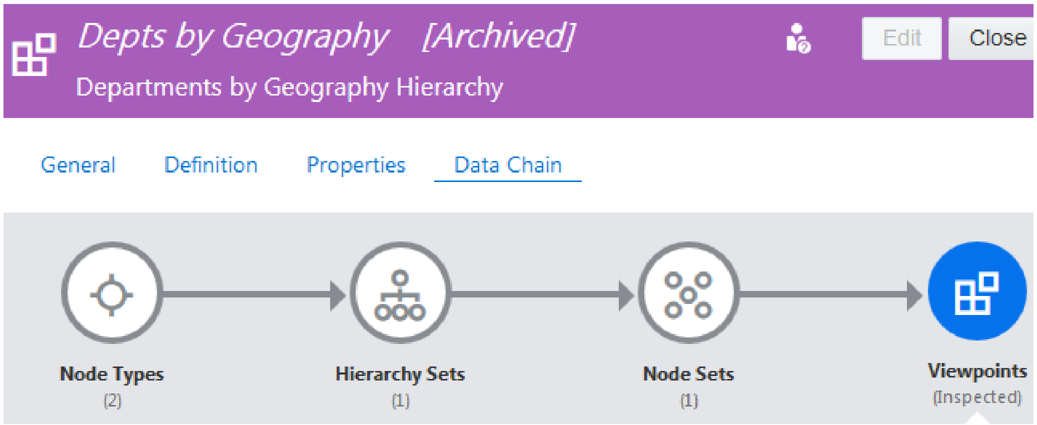
- A user who has the *Owner* or *Metadata Manager* permission on the application or dimension that contains the data chain object can change the object’s status.
- To be able to return a data chain object’s status to Active, the dimension that contains the data chain object must be Active.
- If you remove a dimension from the application registration, the dimension and its data chain objects are set to Archived.

- You cannot archive a viewpoint that is bound to a dimension. If you want to archive the viewpoint bound to a dimension, you must first change the dimension's bound viewpoint. See Binding a Viewpoint to a Dimension in Planning and FreeForm Applications or for Universal application dimensions, see Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications.
- You can set a viewpoint to archived at any time. However, if you validate a request with item details that reference an archived viewpoint, you'll receive a validation message that the viewpoint could not be found. You can delete the request item details that reference the archived viewpoint and then try to validate the request again or you can delete the entire request if it's in Draft status.

For more information, see Working with Data Objects and the Data Chain.

Archiving Order

You can archive an Active data chain object as long as no other Active data chain object depends on it. You need to archive data chain objects in this order: viewpoint, node set, hierarchy set (if used), and then node type. In the example below, we've archived the viewpoint. We are not able to archive the hierarchy set until the node set has been archived.



Archiving a Data Chain Object

To archive a data chain object:

> **Note:**
>
> Remember to archive data chain objects in this order: viewpoint, node set, hierarchy set, and then node type.

1. Inspect a view that contains the data chain object.
2. Select **Definition** and then click the name of the viewpoint.

   The viewpoint inspector opens to the General tab.
3. Follow these steps:
   - To archive a viewpoint:
     a. Click **Edit**.
     b. From **Status**, select **Archived** and then click **Save**.

- To archive a node set, hierarchy set, or node type:
    a. Select **Data Chain**.
    b. Click the data chain object.
    c. Select the object to archive.
    d. Click **Edit**.
    e. From **Status**, select **Archived** and then click **Save**.

Unarchiving Order

You can return an archived object to Active status as long as the objects that it depends on are also Active. You need to set data chain objects to active status in this order: node type, hierarchy set (if used), node set, and then viewpoint. In the example above, we've returned the node type, hierarchy set, and node set to Active status. Now we can set the viewpoint to Active status also.

Unarchiving a Data Chain Object

To return a data chain object to Active status:

> **Note:**
>
> Remember to return status to Active in this order: node type, hierarchy set, node set, and then viewpoint.

1. Inspect a view that contains the data chain object.
2. Select **Definition** and then click the name of the viewpoint.

   The viewpoint inspector opens to the General tab.
3. Follow these steps:
   - To return a viewpoint to Active status:
       a. Click **Edit**.
       b. From **Status**, select **Active** and then click **Save**.
   - To return a node set, hierarchy set, or node type to Active status:
       a. Select **Data Chain**.
       b. Click the data chain object.
       c. Select the object to return to Active status.
       d. Click **Edit**.
       e. From **Status**, select **Active** and then click **Save**.

21

Defining Alternate Views and Viewpoints

Alternate views and viewpoints let you see enterprise data organized in a different ways.

When you register an application, a default view is created automatically. This default view contains viewpoints for each bound dimension in your application, and therefore the default view enables you to work with all of the data in all of the dimensions in a hierarchy structure that matches the structure of your external application.

But what if you want to see this hierarchy structure organized in different ways? Or you want certain users to work with a subset of data in an application? Or perhaps you want to work with data from more than one application at the same time? For these use cases, you can create alternate views and viewpoints.

Videos

Your Goal	Watch This Video
Learn about creating alternate viewpoints.	▶ Creating Alternate Viewpoints

Defining Alternate Views

There are two main use cases for creating a view that is separate from the default application view:

- You want to create a dedicated view to have specific users maintain specific data sets without having full access to all of the data in your application.
- You want to create a master view that contains viewpoints from multiple applications. Then, you can make changes in your master view and use subscriptions to push those changes out to the subscribing viewpoints (see Subscribing to Viewpoints).

This alternate view provides several more flexible options to work with your enterprise data:

- You can add viewpoints for only the dimensions that you want to maintain in that view (see Creating a Viewpoint).
- You can set specific permissions on that view that are separate from the view permissions on your default view (see Security for Views and Viewpoints).

Defining Alternate Viewpoints

With alternate viewpoints, nodes can be aggregated, rolled up, or summarized using a hierarchy that differs from the hierarchy in the bound viewpoint. In other words, alternate viewpoints let you group lower-level nodes under top nodes other than the top nodes in the bound viewpoint.

The following list describes some common reasons for creating alternate viewpoints:

- Group nodes under different parent nodes while preserving existing parent-child relationships. For example, you can create hierarchies that group cost centers by geography, function, and legal entity.

- Examine how a hierarchy and its associated data would be impacted if nodes are rolled up using parent-child relationships that differ from those in the bound viewpoint.
  For example, you could use an alternate viewpoint to analyze the impact of splitting a business unit into multiple units. You could then export the alternate viewpoint to an external application in order to evaluate the impact to the associated data.
- Define a hierarchy to be used in a report.

Best Practices

There are various types of alternate viewpoints. The type of alternate viewpoint you use, and the data chain objects you use to build the alternate viewpoint, depend upon your objective. The following questions determine which type of alternate viewpoint you need to define and which data chain objects you need to create:

- Will the alternate viewpoint create relationships between node types that are not used in the bound viewpoint? In other words, will the alternate viewpoint use a category of nodes that is not in the bound viewpoint?
- Will the alternate viewpoint use parent-child relationships that differ from the relationships in the bound viewpoint?
- Is the alternate viewpoint's purpose to group the bound viewpoint's bottom-level nodes under new top nodes while preserving the bound viewpoint's parent-child relationships? If so, will you have users work on the alternate viewpoint in the same viewpoint as the bound viewpoint or in a new viewpoint?

The following table summarizes the types of alternate viewpoints you can create.

Table 21-1   Types of Alternate Viewpoints

Type	Example	More Information
Group nodes under new top nodes while preserving the bound viewpoint's parent-child relationships.	Cost centers rolled up by geography and by function.	Alternate Viewpoints Using Different Top Nodes
Define parent-child relationships that differ from those defined in the bound viewpoint. This type of hierarchy is often used for what-if or future planning scenarios.	Analyze the impact of a reorganization of business units.	Alternate Viewpoints Using Different Parent-Child Relationships
Create relationships using node types that are not in the bound viewpoint's data chain.	A hierarchy listing products organized by supplier, starting with a bound viewpoint of products.	Alternate Viewpoints Using Different Node Types
Define relationships between nodes in different applications that share information with each other.	A mapping hierarchy maps source nodes from one application to target nodes in another application. The mapping hierarchy segregates the dimension mapping relationships from the hierarchy relationships in the dimension.	Creating Mapping Viewpoints

If you need to create new node types, node sets, or hierarchy sets, see:

- Creating a Node Type
- Creating Node Sets
- Creating Hierarchy Sets

Exporting an Alternate Viewpoint's Data

To export an alternate viewpoint's data to an external application, bind the dimension to the alternate viewpoint, see Binding a Viewpoint to a Dimension in Planning and FreeForm Applications.

> **Note:**
>
> You can't change the bound viewpoint for a Universal application's dimension. You can however download the alternate viewpoint's data and then load the data into the external application, see Downloading a Viewpoint.

Creating a View

Create alternate views to organize and manage data differently from the application's default view.

> **Note:**
>
> Before you create a view, ensure that you have the data chain defined for the task you want to accomplish.

To create a view:

1. From Views, click **Create**.
2. Define the name and description for the view, and then click **Create**.
   An inspector for the view opens.

After you create the view, you create viewpoints to work with your application's data. See Creating a Viewpoint.

Creating a Viewpoint

You can create a viewpoint to point to a different node set or to set the ability to edit properties or change permissible actions. For example, suppose you want to enable a data steward to view data but not make changes. You could turn off the ability to perform permissible actions, such as add, insert, and move and disable the ability to edit properties.

To create a viewpoint:

1. Inspect the view that you want to add a viewpoint to.
2. On the Viewpoints tab, click **Create**.
3. From **Application Dimension**, select the application to use for the viewpoint.
4. Enter a name, and optionally a description for the viewpoint.
5. In **Viewpoint Class**, select an option:

- **Normal**: displays current application data. See Working with Viewpoints.
- **Time Labeled**: displays application data from a historical point in time. See Working with Time Labeled Viewpoints

6. For Time Labeled viewpoints, select a time label for that view. The time label must already have been created in order to use it for a viewpoint. To create a new time label, see Creating, Editing, and Deleting Time Labels.
7. Click **Create**.
   An inspector for the viewpoint opens. The viewpoint is in a draft state until you select a node set.
8. Click **Edit**, select a **Node Set**, and then click **Save**.

   > **Tip:**
   >
   > For time labeled viewpoints, select a node set that has the historical data that you want to access.

   The viewpoint status is changed to Active.

Alternate Viewpoints Using Different Top Nodes

You can create an alternate viewpoint in which lower-level nodes are grouped under different top nodes than the primary viewpoint's top nodes but that also preserve the lower-level nodes' parent-child relationships.

For example, let's say there is a hierarchy of cost centers that includes a node named 110 with child nodes named 115, 116, and 117. The 110 node and its child nodes can be placed under a different top node in an alternate viewpoint than in the primary viewpoint.

This type of alternate viewpoint must include the hierarchy that is in the primary viewpoint's data chain. That hierarchy set must allow shared nodes, see Understanding Shared Nodes.

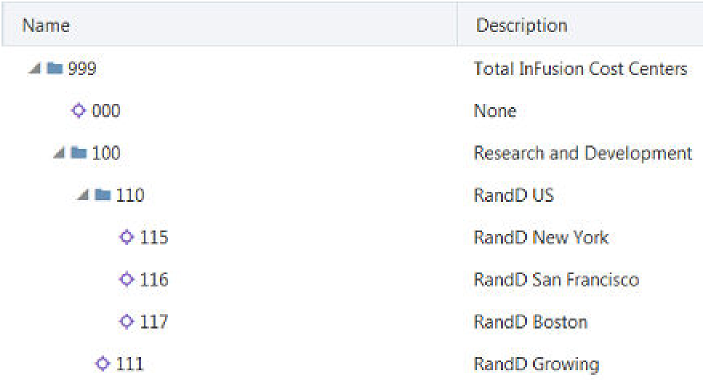
To create this type of alternate viewpoint in a new viewpoint, take the following steps. For more information, see, Defining Alternate Views and Viewpoints.

1. Create a new node set using the same hierarchy set used for the primary viewpoint.
2. Create the alternate viewpoint using the node set you just created.

> **Note:**
>
> If you prefer, you can instead use the primary viewpoint to group nodes under different top nodes. To do so, you would add new top nodes and then insert the lower-level nodes.

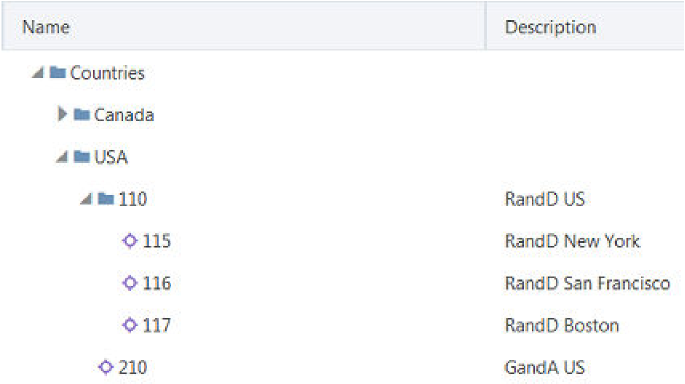
Example 21-1 Group Nodes Under Top Nodes for Countries

Suppose there is a primary viewpoint in which cost centers are grouped by lines of business. The lines of business include cost centers for research and development in the United States. The hierarchy of United States cost centers in the following example has a parent node named 110 with child nodes named 115, 116, and 117:



Name	Description
999	Total InFusion Cost Centers
000	None
100	Research and Development
110	RandD US
115	RandD New York
116	RandD San Francisco
117	RandD Boston
111	RandD Growing

Now suppose you need to create an alternate viewpoint that groups cost centers by the countries in which they are located. In the following example, the alternate viewpoint's top node is Countries, which includes a node for the USA. In the following example, the USA node includes the hierarchy of United States research and development cost centers:



Name	Description
Countries	
Canada	
USA	
110	RandD US
115	RandD New York
116	RandD San Francisco
117	RandD Boston
210	GandA US

Alternate Viewpoints Using Different Parent-Child Relationships

You can create an alternate viewpoint in which nodes' parent-child relationships differ from the relationships defined in the primary viewpoint. For example, you would use this type of hierarchy to analyze the impact of a reorganization that combines multiple business units into one unit.

Tip:

Use this type of alternate viewpoint for what-if and future scenario planning.

Note:

For Oracle Financials Cloud General Ledger applications, each tree (tree code/tree version name) must have a separate hierarchy set.

To define a hierarchy in which parent-child relationships differ from those of the primary viewpoint, you create new instances of the following data chain objects:

- Hierarchy set

> **Note:**
>
> A new hierarchy set is needed because you would not be changing the primary viewpoint's parent-child relationships

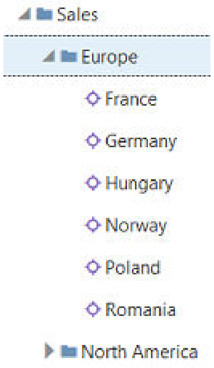
- Node set
- Viewpoint

The following steps describe how to create a hierarchy with parent-child relationships that differ from those of the primary viewpoint. For more information, see, Defining Alternate Views and Viewpoints.

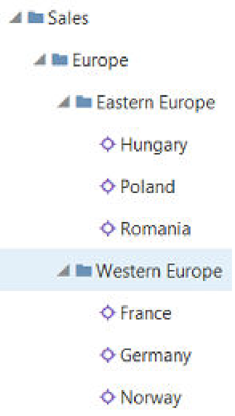
1. Create a new hierarchy set that uses the same node types as those used in the primary viewpoint.
2. Create a new node set using the same hierarchy set you just created.
3. Create the alternate viewpoint using the node set you just created.

Example 21-2 Evaluating a Reorganization

Suppose a company needs to evaluate the impact of reorganizing the Sales department. Currently the Sales organization has top nodes for continents, and the continent nodes have child nodes for countries. The following example shows the Europe branch of the hierarchy:



In the alternate viewpoint that models the reorganization, the branches for continents have child nodes for regions, and the regional nodes have child nodes for countries. The following example shows that the Europe node in the alternate viewpoint has child nodes for the Eastern Europe and Western Europe regions, with the regional nodes containing nodes for countries:



- Sales
  - Europe
    - Eastern Europe
      - Hungary
      - Poland
      - Romania
    - Western Europe
      - France
      - Germany
      - Norway

Alternate Viewpoints Using Different Node Types

You can create an alternate viewpoint that uses node types other than those used by the primary viewpoint. For example, suppose you have a primary viewpoint consisting of nodes for products and are tasked with creating an alternate viewpoint of products by supplier. The registered application does not yet have nodes for suppliers, so you would create a node type for suppliers, then create a hierarchy in which the product nodes are contained by supplier nodes.

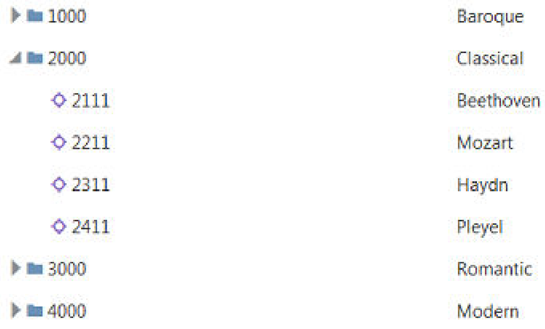
To create this type of alternate viewpoint, you create a new hierarchy set and node set. If you are defining an alternate viewpoint that uses node types that do not exist in the application, you also create new node types.

The following steps describe how to create an alternate viewpoint for creating relationships between node types other than those in the primary viewpoint. For more information, see, Defining Alternate Views and Viewpoints.

1. If the alternate viewpoint will include node types that do not currently exist in the application, create a new node type for each new category of nodes.
2. Create a hierarchy set that defines the parent-child relationship between the node types.
3. Create a new node set using the same hierarchy set you just created.
4. Create the alternate viewpoint using the node set you just created.

Example 21-3 Products by Store

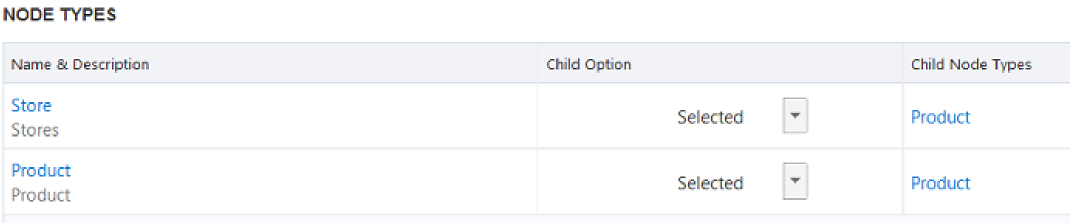
Suppose that you are working with a primary viewpoint of products that includes nodes for musical products, as shown in the following example:



1000	Baroque
2000	Classical
2111	Beethoven
2211	Mozart
2311	Haydn
2411	Pleyel
3000	Romantic
4000	Modern

Now suppose you are tasked with creating an alternate viewpoint that lists products by store. The application currently does not have a node type for stores. To create this alternate viewpoint you would do the following:

1. Create a node type for stores.
2. Create a hierarchy set that defines the parent-child relationship between the node types. For this example, stores can have child nodes for products but not other stores, and products can only have child nodes of other products:

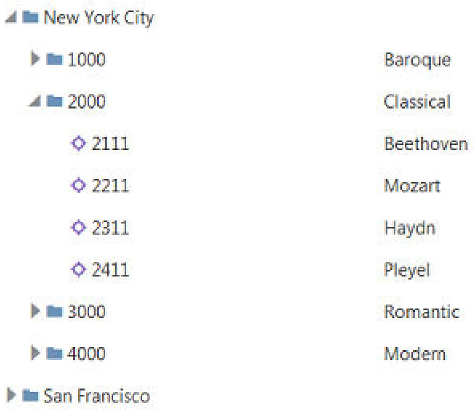


NODE TYPES

Name & Description	Child Option	Child Node Types
Store Stores	Selected	Product
Product Product	Selected	Product

3. Create a new node set using the hierarchy set you just created.
4. Create the alternate viewpoint using the node set you just created.

The following example shows the new hierarchy. The top nodes are for stores:



```
◢ New York City
    ▶ 1000              Baroque
    ◢ 2000              Classical
        2111            Beethoven
        2211            Mozart
        2311            Haydn
        2411            Pleyel
    ▶ 3000              Romantic
    ▶ 4000              Modern
▶ San Francisco
```

Creating Mapping Viewpoints

Mapping ensures nodes and their relationships are represented accurately across applications. Mappings are used to define relationships between nodes in dimensions of different applications that share information with each other. In some cases, the dimensionality of source data does not match the target application, and mappings must be defined to determine how the shared data will be transformed in order to be consumed by the target application.

For an example, see Mapping Source Dimensions to Target Dimensions.

Videos

Your Goal	**Watch This Video**
Learn about mapping data across applications.	Mapping Data Across Applications

These are common scenarios for mapping:

- Map bottom level nodes from one application to a different application.
  For example, suppose your corporation acquires a subsidiary company and you want to map the subsidiary local accounts to your corporate accounts. You can use a mapping hierarchy to perform the mapping. This places the bottom level nodes in the mapping hierarchy instead of your corporate account dimension, keeping the corporate account dimension intact and less cluttered.
- Map general ledger transaction accounts to financial summarized accounts used for planning purposes.

> **Note:**
>
> In all scenarios, a mapping hierarchy set holds the mapping relationships separate from the hierarchy relationships in the dimension.

Considerations:

- Mapping is not supported for Oracle Financials Cloud General Ledger applications.
- For all other applications:
  - You can map bottom level source nodes to bottom level target nodes only.
  - You can map one or more source node types to one target node type only.
  - To export mapping data:
    - Use a view containing both the mapping viewpoint and a viewpoint bound to the target dimension.
    - For Planning and Financial Consolidation and Close applications, you can export the mapping data to a file or the inbox of an external application.
    - For Universal applications, you can export the mapping data to a file only.
  - Optionally, use the `Core.Change.Sign` property if you are mapping nodes of different types and need to reverse the sign on the target nodes. For example if you are mapping assets to liabilities add the `Core.Change.Sign` property and set it to `True` . When you export the mapping data:
    - If the target nodes are in a Planning or Financial Consolidation and Close application, a negative sign is added to the target node.
    - If the target nodes are in a Universal application, a column containing the negative sign is exported.

    When an external application imports the mapping data, it applies the change sign to the target nodes. For example, if your target is a Planning application and you are mapping asset nodes to liability nodes add the `Core.Change.Sign` on the source node type and set it to `True`. On export the change sign is added to the target node. Within Planning the Data Management module imports the data and applies the change sign to the target nodes.

Mapping Source Nodes to Target Nodes

You can map nodes from one or more dimensions in source applications to a nodes in a dimension in a target application. First, you create at least one source node type for each source dimension. Then, you create a map binding that includes the source and target node types, and mapping keys that define locations for each source node type. Creating a maintenance view with the source, target, and mapping viewpoints enables you to map source to target nodes by dragging and dropping nodes from the source and target viewpoints into the mapping viewpoints. When you have finished mapping nodes, you can export the mapping.

To map source nodes to target nodes:

1. **Create source node types for the map binding.** You must create at least one node type for each source dimension. You can create multiple node types for a source dimension if necessary.
   a. Create node types, using the target dimension, to hold the source nodes in the mapping viewpoint, see Creating a Node Type.

   > **Note:**
   >
   > You must create new source node types because you cannot use the source node types from the source applications. Node types are specific to dimension and applications.

   b. Create node type converters to convert the node types from each source application to the node types you created in step 1a above. See Working with Node Type Converters.
   c. Optionally, add the `Core.Change.Sign` property to the source node types if you are mapping nodes which require a sign change when transferring numerical data across applications (for example, if you are mapping an asset to a liability).
2. **Create the map binding.** See Creating a Map Binding.

   A map binding enables you to map node types to one another for different source and target systems.

   > **Note:**
   >
   > When you add node types to the map binding, add all of the target node types that you want to map to as well as the source node types that you created for the mapping viewpoint.

   When you create the map binding, the following data chain objects are also created:
   - A new hierarchy set is created, and the target node types that you identified when you created the map binding are added to the hierarchy set.
   - A new node set is created and pointed to the new hierarchy set.
   - A new mapping viewpoint is created in the default view for the target application, and it is set to use the new node set.
3. **(Optional) Configure the mapping hierarchy set.** See Creating Hierarchy Sets.
   The hierarchy set that gets created when you create the map binding contains all of the source and target node types that you specified when you created the map binding. You can edit this hierarchy set if necessary:
   - If you have added additional source node types after you created the map binding, add them to your hierarchy set so that they can be used when you define the mapping keys.

- Add relevant rollup rules for the nodes in the hierarchy set. For example, the target node type can have source nodes and itself as children, but the source node types cannot have any children.

4. **Define the mapping keys.** See Defining Mapping Keys.

   Mapping keys enable you to set up locations, which are combinations of the source and target node types that you select, in order to export the mapping data.

   > **Note:**
   >
   > When defining the mapping keys, select the mapping viewpoint that was created when you created the map binding.

5. **Create a maintenance view containing the source, target, and mapping viewpoints**

   Set up a maintenance view to contain the source viewpoint, target viewpoint and mapping viewpoint that was created when you created the map binding. This will enable you to drag and drop nodes from the source viewpoint and target viewpoint into the mapping viewpoint.

   a. Create a view, see Creating a View.

   b. Create a viewpoint containing the source dimension, see Creating a Viewpoint.

   c. Create a viewpoint containing the target dimension.

   d. Create the mapping viewpoint using the node set that was created with the map binding.

6. **Map source nodes to target nodes**

   a. Open the maintenance view and click the **Side by Side** button to work with the viewpoints.

   b. Display the mapping viewpoint on the right, and toggle between the target and source viewpoints on the left.

   c. Within the mapping viewpoint:

      i. Create a top node using the target node type. For example, name it Mapping Account, if you are mapping account nodes, see Making Changes Interactively.

      > **Note:**
      >
      > Use this top node to group the mapped target and source nodes. This enables mappings to be organized and keeps the number of top nodes at a minimum.

      ii. Drag and drop a node from the target application. In the mapping viewpoint this will be a parent node.

      iii. Drag and drop a node from the source application. In the mapping viewpoint this will be a child node.

> **Note:**
>
> You can map bottom level source nodes to bottom level target nodes only.

d. **Optional**: If you are mapping nodes which require a sign change when transferring numerical data across applications, set the `Core.Change.Sign` property on the source node type to True, which will flip the sign on the target node. Otherwise leave the property set to False.

e. Repeat these steps to map as many nodes as desired.

f. To create the mapping key and export mapping data you need a view that contains both the mapping viewpoint and a viewpoint bound to the target dimension. To set this up, you can use one of these views:

   - Use the application default view which has a viewpoint bound to the target dimension created during application registration. Add the mapping viewpoint using the mapping hierarchy node set to the default view. Now the default view can be used to export both mapping data and dimension data.
   - Use the maintenance view and bind the target viewpoint to the target dimension.

> **Note:**
>
> In the maintenance view, the target viewpoint was created outside of application registration and by default is not bound to the target dimension. To bind the target viewpoint, edit the target dimension's existing binding and change it to point to the target viewpoint. See Identifying Objects Bound to Dimensions.

7. **Export the mapping data.** See Exporting Mapping Data.

Optionally, you can import the mapping data to Planning or Financial Consolidation and Close, using Data Management.

> **Tip:**
>
> If you add source node types after you have already created the map binding, follow these steps to add them to your mapping viewpoint:
>
> 1. Add the new source node types to the hierarchy set that was created when you created the map binding.
> 2. Add rollup rules for the source node types in the hierarchy set.
> 3. Add the new source node types in your mapping keys.

Working with Time Labeled Viewpoints

Time labeled viewpoints enable you to access data in a node set from a historical point in time.

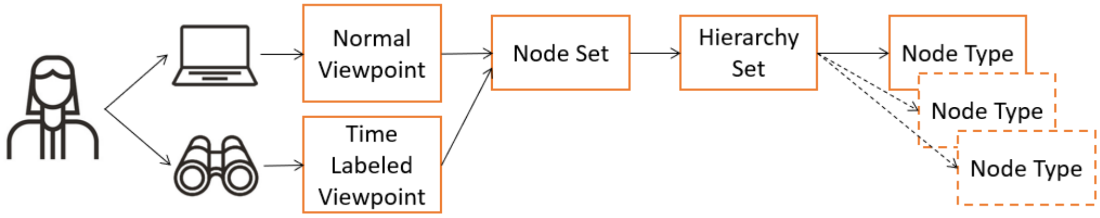
Videos

Your Goal	Watch This Video
Learn how to use time labeled viewpoints to access data in a node set from a historical point in time to assist in comparing, reporting, and analyzing data.	▶ Working with Time Labeled Viewpoints

Each time labeled viewpoint uses a single time label, and you can create different types of time labeled viewpoints based on different business needs. For example, you could create a rolling time labeled viewpoint that compares Entity data to history on a quarterly basis, or you could create a fixed time label for a Cost Center hierarchy to use as a reference before a reorganization.

Before you create a time labeled viewpoint, you must create the time label itself. See Creating, Editing, and Deleting Time Labels.

When you create a time labeled viewpoint, you should use a node set that contains the historical data that you want to view. The following diagram demonstrates that you can use the same node set for both a normal and a time labeled viewpoint. You can view and edit data in the normal viewpoint, but the time labeled viewpoint is read only.



Considerations

- You must have the *Owner* permission on a view as well as the *Data Manager* or *Metadata Manager* permission on the data objects in that view in order to create a time labeled viewpoint for it.
- Time labeled viewpoints are **read only**. You cannot modify any data in a time labeled viewpoint. That means the following operations are not available:
  - Requests
  - Imports and Viewpoint Loads
  - Subscriptions

  You can perform read only operations such as searching, comparing, extracting (Full extracts only), and downloading from time labeled viewpoints.
- Time labeled viewpoints cannot be bound to an external application and thus cannot be used in imports or exports.

- You cannot use time labeled viewpoints in viewpoint queries, and you cannot validate a time labeled viewpoint.
- A user's current viewpoint permissions determine their data access in a time labeled viewpoint. That means that if a user currently has access to viewpoint data they will be able to view that data, even if they did not have access to that viewpoint data in the associated time period for the time label.
- Time labeled viewpoints can access data only for time periods after the creation date of the node set for the viewpoint.
- The properties that are displayed in time labeled viewpoints are based on the current properties for each node type in the node set:
  - If a property for a node type exists now that did not exist at the associated time period for the time label, the property is displayed with no value in the viewpoint.
  - If a property for a node type does not exist now but did exist at the associated time period for the time label, the property is not displayed in the viewpoint.
- Time labeled viewpoints use historical property parameters as they existed at the associated time period for the time label in order to determine property values. These include:
  - Allowed or invalid values
  - Inherited values
  - Derived values based on expressions

  However, if you inspect a property in a time labeled viewpoint, the inspector uses the current property parameters to determine the property value.

  For example, if the Alias property was derived from the node name at the time of the historical time label but was later changed so that now it is derived from the node name plus the description, a time labeled viewpoint using that time label will display the node name as the Alias. But if you inspect the Alias property, the node name plus the description will be displayed in the inspector.
- Node history in time labeled viewpoints is displayed up to the specified date and time that is configured in the time label.

> **Note:**
>
> Users with at least *Participant (Read)* access to a time-labeled viewpoint can also create private time labels that enable them to override the time value of the existing time label without needing a View Owner to create new time labels for them. See Working with Private Time Labels.

Working with Private Time Labels

Users with at least *Participant (Read)* access to a time-labeled viewpoint can create private time labels that enable them to override the date and time of the existing time label.

This enables them to view historical data for a time other than the time period in the original time label without needing a View Owner to create a new time label for them.

Considerations

- You must have at least *Participant (Read)* access to a time labeled viewpoint in order to create, edit, and remove private time labels
- You can create private time labels only for time labeled viewpoints. You cannot create time labels for normal viewpoints.
- You can create multiple private time labels for a time labeled viewpoint, and multiple viewpoints in a view can share the same time label.
- Private time labels can be viewed and used only by the user who created them.
- With the exception of the permission needed to create a private time label, all other considerations for time labels also apply to private time labels. See Considerations for Time Labeled Viewpoints.

Creating a Private Time Label

You create private time labels from the viewpoint menu in an open view.

1. For a time labeled viewpoint in an open view, click **Viewpoint Menu** ⋮ next to a viewpoint name and select **Change Time Label**.
2. Select **Create**.
3. Enter a name and description for the private time label.
4. Enter or use the date selector to specify a date and time for the time label.
5. Select the time zone for the time label, and click **OK**.

The viewpoint is reloaded with data from the private time label that you created. The **Time Label** icon in the viewpoint header is also replaced by a **Private Time Label** icon.

> **Note:**
>
> If you create another private time label for the viewpoint, the old private time label is removed. If it is not being used by any other viewpoints, the old private time label is deleted.

Editing, Selecting, and Removing Private Time Labels

1. In a time labeled viewpoint in an open view, click **Viewpoint Menu** ⋮ next to a viewpoint name and select **Change Time Label**.
2. Perform an action:
   - Click **Edit** to edit the name, description, dates, or time zone of the existing private time label.
   - Click **Select** and then use the **Time Label** drop down menu to select a private time label that is being used by another viewpoint.
   - Click **Remove** to remove the private time label and restore the time value from the original time labeled viewpoint. If this private time label is not being used by any other viewpoints, it is deleted.

> **Note:**
>
> You can also delete private time labels from the **Time Labels** tab in the view inspector. See Inspecting a View.

Copying a Viewpoint

You can copy an active viewpoint to create a new viewpoint. You can copy the nodes from the viewpoint as well as the relationships between node types if copying from a hierarchy viewpoint. If you want to retain a version of a viewpoint you can copy it and make it read only.

> **Note:**
>
> You cannot copy a time labeled viewpoint.

Considerations

- When a viewpoint is copied, any associated data grants and approval policies for data objects are also copied. However, saved viewpoint queries (both public and private) are not copied to the new viewpoint.
- If you just want to limit node and property actions for a specific user with the same hierarchy you have in a viewpoint, instead of copying the viewpoint, you could make a new viewpoint using the same node set but set the permissible actions and editable properties to limit the node and property actions in the new viewpoint.
- If you open a viewpoint in a duplicate tab, you cannot copy the viewpoint from the duplicate tab. You can copy the viewpoint from the original tab only. See Using Duplicate Viewpoint Tabs.

Copying Nodes

If you copy nodes:

- Node types from the original viewpoint are copied to create new node types to be used in the new viewpoint's node set.
- Nodes in the original viewpoint's node types are copied to the new node types in the new viewpoint.

If you copy nodes for a hierarchy viewpoint, you must also copy relationships.

> **Note:**
>
> If you do not copy nodes, the node types from the original viewpoint are used in the new viewpoint's node set.

Copying Relationships

If you copy relationships from a hierarchy viewpoint:

- The hierarchy set from the original viewpoint is copied to create a new hierarchy set for the new viewpoint. All relationships in the original hierarchy set are copied to the new hierarchy set.
- The node types based on the copy nodes option are used by the new hierarchy set.

Creating a New Viewpoint Based on Copy Options

The following tables show how the data chain for a new viewpoint is created depending on the copy options that you select when copying a viewpoint.

For example, if you select to copy relationships only from a hierarchy viewpoint, then:

- A new node set is created.
- A new hierarchy set is created.
- The original node types from the copied viewpoint are used.

Table 21-2 Copy Hierarchy Viewpoint

Copy Nodes	Copy Relationships	Node Set	Hierarchy Set	Node Type
Yes	Yes	New	New	New
No	Yes	New	New	Original
No	No	New	Original	Original

Note:

When you copy a hierarchy viewpoint, you cannot copy nodes only.

Table 21-3 Copy List Viewpoint

Copy Nodes	Node Set	Node Type
Yes	New	New
No	New	Original

Binding Considerations When Copying Viewpoints

When you copy a bound viewpoint, the options that you select for copying nodes and relationships affect the binding status of the data objects in the new viewpoint, as follows:

- If you do not select copy nodes or copy relationships, the new viewpoint is fully bound. You can use this to create a filtered view using a new node set. Any updates that are made in the copied viewpoint will affect your production data.
- If you copy relationships but not nodes, the viewpoint is node type bound. This is useful for creating what-if scenarios for alternate hierarchies. For example, you can model a reorganization of employees by copying the relationships but not the nodes in a viewpoint. Because the viewpoint is node type bound, changes made to the nodes (for example, an employee change of address) would affect your production data, but changes made to the relationships (for example, moving employees to a different manager) would not.
- If you copy both relationships and nodes, the new viewpoint is completely unbound. This is useful to create "sandbox" environments that let you freely model all changes without affecting your production data. You can use also use this option for versioning purposes.

Copying a Viewpoint

To copy a viewpoint:

1. From **Views**, open an active view.
2. Select a viewpoint.
3. Place your cursor to the right of the viewpoint name, then click the menu icon ⋮, and then select **Copy**.
4. Define the name and description for the viewpoint, select whether to copy nodes and relationships (hierarchy viewpoint only), and then click **Copy**.

   The viewpoint is created and ready for you to work with.

Versioning Best Practices

Here are a few versioning best practices when copying a viewpoint.

Use Case	Best Practice
Make a historical copy of nodes and hierarchies for a specific time period. For example, create a copy of the Balance Sheet and Income Statement as of Jan. 2018. Continue to edit the original version with changes in Feb. 2018.	Create a separate version of nodes and hierarchies by selecting Copy Nodes and Copy Relationships.
Reorganize or restructure hierarchies for different time periods or business events. For example, create a copy of a business unit hierarchy and then restructure the copied viewpoint hierarchy to visualize the effects of proposed organizational changes.	Create a separate version of hierarchies using the same nodes by selecting Copy Relationships. See Changing the Bound Viewpoint to Export an Alternate Viewpoint.

Changing the Bound Viewpoint for a Dimension

Alternate viewpoints are created outside of application registration and thus by default are not bound to an external application. If you need to import or export data for an alternate viewpoint, you must bind the alternate viewpoint to an external object by inspecting the Oracle Fusion Cloud Enterprise Data Management dimension for the binding and changing the bound viewpoint.

> **Note:**
>
> You cannot bind a time labeled viewpoint to a dimension.

The procedure for changing the bound viewpoint depends on the application type:

- Planning applications support binding to a single external dimension within a Cloud EDM dimension. If you want to bind an alternate viewpoint, you edit the existing binding and change it to point to your alternate viewpoint. See Binding a Viewpoint to a Dimension in Planning and FreeForm Applications.
- Oracle Financials Cloud General Ledger applications support binding to multiple external objects (segment/value sets and trees) within a Cloud EDM dimension. You can change an

existing bound viewpoint (for example, if you no longer want to export an old version of a tree), or you can create a new binding by copying an existing binding (for example, if you want to add a new bound version of a tree but you also want to keep the existing bound tree versions). See Copying Bindings to Create Trees or Tree Versions.

- Financial Consolidation and Close applications support binding to a single external dimension within a Cloud EDM dimension. If you want to bind an alternate viewpoint, you edit the existing binding and change it to point to your alternate viewpoint. See Binding a Viewpoint to a Dimension in Financial Consolidation and Close Applications.
- Universal applications support binding to a single external dimension within a Cloud EDM dimension. If you want to bind an alternate viewpoint, you edit the existing binding and change it to point to your alternate viewpoint. You can also customize the node types and properties in the dimension. See Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications.
- When changing the binding for normal dimensions, you cannot select a viewpoint that contains Legacy GL or Lookup class node types for any bindings other than mapping bindings. For lookup dimensions, you can select a viewpoints with Legacy GL or Lookup class node types.

22 Working with Properties

Dimensions include properties that are used by their nodes. The Properties page lists the properties that can be added to dimensions. Oracle Fusion Cloud Enterprise Data Management provides an initial set of properties. You can create custom properties from the properties page or when you register Universal applications.

The properties available to an application's dimensions depend upon the application type:

- When you register a Universal application, you can create custom properties or select existing properties.
- When you register an Oracle Fusion Cloud EPM application, properties are automatically added, based on the dimensions' Dimension Types.

You can customize how properties are displayed in viewpoints. For example, you can display an alternate name for a property, see Understanding How to Configure Properties for a Viewpoint.

An application's type determines some characteristics of properties, as described in the following table:

Table 22-1 Properties and Application Types

Application Type	Property Considerations
Cloud EPM applications	You cannot add or remove properties. The only properties available to nodes are the properties automatically added when the application was registered. For application-specific properties, see: • Predefined Properties for Planning and FreeForm Applications • Predefined Properties for Financial Consolidation and Close Applications • Predefined Properties for Enterprise Profitability and Cost Management Applications
Oracle Financials Cloud General Ledger Applications	You cannot add or remove properties. The only properties available to nodes are the properties automatically added when the application was registered. For application-specific properties, see Predefined Properties for Oracle Financials Cloud General Ledger Applications
Oracle E-Business Suite General Ledger Applications	You cannot add or remove properties. The only properties available to nodes are the properties automatically added when the application was registered. For application-specific properties, see Predefined Properties for E-Business Suite General Ledger Applications
Universal applications	• You can add properties for non-Universal application types. For example, you can add a Planning property to a Universal application. • You can add or remove properties from a dimension's node type. • You can delete custom properties as long as they are not referenced by any other data chain object or by an application registration.
Users applications	You cannot add or remove properties. The only properties available to nodes are the properties that were automatically added when the application was created. See Predefined Properties for Users Applications.

Note:

- Cloud EPM applications may have unique considerations. If so, those considerations are described in the topic for the application type.
- For all application types, you can add properties that are not registered with the application to a node type, thus making the property available in viewpoints. Unregistered properties added to node types are not included when you import or export an application's data.
  Why would you add a property to a node type if it cannot be imported or exported? You can use such properties to customize sorting or to include descriptions and comments for Cloud EDM users.

Property Data Types

Properties are defined using the following data types:

Property Data Type	Definition	Examples
Boolean	A value that can be true or false	`True` `Yes`
Date	A date value. Does not include a time value.	`03/05/2023`
Float	A number that has a decimal place	`587.5`
Integer	A whole number	`1000`

List	A sequence of strings	`Accounts Receivable`
		`Cash`
		`Depreciation`

Memo	Large strings that allow formatting and whitespace characters, such as line feeds and Tab characters.	A Java regular expression, for example `return node.properties.PLN.Formula.replace("\R", ". ")`
Node **Note**: Subtypes are available for node data types. See Node and Node List Data Type Subtypes.	Basic subtype: A pointer to another node in either the same or a different node set in the current application.	A parent node, for example, `NA - Canada`
	User subtype: A pointer to a User or User Rollup node in any Users application	A cost center manager, for example `Barry Mills`
Node List **Note**: Subtypes are available for node list data types. See Node and Node List Data Type Subtypes.	Basic subtype: A pointer to a list of other nodes in either the same or a different node set in the current application.	A list of alternate accounts for this account, for example: `11101,11103,11105`.
	User subtype: A pointer to a list of User or User Rollup nodes in any Users application	A list of cost center managers, for example: `Barry Mills,Casey Brown,Carol Judd`
Numeric String	A string that allows numeric characters (0-9) only.	`12475381469113081838`

Property Data Type	Definition	Examples

Sequence	An automatically-generated incrementing number.	`0001005`
		`0001010`
		`0001015`

String **Note**: Subtypes are available for string data types. See String Subtypes.	Basic subtype: Text and numeric characters.	`Owner's Equity`
	Email subtype: Email address	`user@companyname.com`
	Hyperlink subtype: Hyperlink information	`https://companyname.com`
Timestamp	A date and time value	`03/05/2023 11:30 AM`

Property Level

The property level determines where property values are defined:

- **Node**: Defined property values apply to that node across all locations and viewpoints.
- **Relationship**: Defined property values are unique to specific parent-child relationships within a hierarchy set.

> **Caution:**
>
> Property level applies to *defined* property values only. Inherited and derived values (if the default expression uses positional logic) may not conform to the node and relationship level constraints. See Inheriting Properties and Derived Properties.

Property Origin

The property origin identifies how a value originated for a node. Property values can have these origins:

- Default: value can be configured when an application is registered
- Defined: value is defined by the user
- Inherited: value is inherited from an ancestor node
- Derived: value is calculated. For example, number of children.
- Derived and Stored: value was derived and then stored on the node.
- Calculated (Name and Parent properties only): During a request, the value of the node name or parent was calculated. For node names, after the request has been completed the origin displays Defined.
- Unknown: the origin of the value cannot be determined

For more information, see:

- Understanding Data Chains
- Configuring How a Viewpoint Displays Properties
- Understanding Property Names and Namespaces
- Adding, Removing, and Configuring a Node Type's Properties

- Removing a Property from a Node Type in a User Defined Dimension
- Inspecting Properties
- Inheriting Properties
- Derived Properties

> **Note:**
>
> For help with building expressions for deriving property values, see Using Expressions to Define Custom Business Logic.

Understanding Property Names and Namespaces

A property is identified by its name and namespace. For example, the `Core.Node Type` property has a namespace of `Core` and a name of `Node Type`.

Namespaces are useful because different external applications might use the same property name. They ensure that each property is uniquely identified while also preserving the property name. Namespaces are automatically assigned to properties when you register an application.

The following table describes the available namespaces.

Table 22-2 Namespaces

Namespace	Description
`Core`	Oracle Fusion Cloud Enterprise Data Management system properties. For example, the `Core.Name` property is used for node names. See System and Statistical Properties.
`CoreStats`	Cloud EDM statistical properties. For example, the `CoreStats.# Children` property calculates the number of child nodes for a given node. See System and Statistical Properties.
`Custom`	Properties for Universal applications.
`EBSGL`	Properties for Oracle E-Business Suite General Ledger applications. See Predefined Properties for E-Business Suite General Ledger Applications.
`FCCS`	Properties for Financial Consolidation and Close applications. See Predefined Properties for Financial Consolidation and Close Applications.
`FCGL`	Properties for Oracle Financials Cloud General Ledger applications. See Predefined Properties for Oracle Financials Cloud General Ledger Applications.
`Lookup`	Properties for Lookup class node types. See Predefined Properties for Lookup Node Types.
`PLN`	Properties for Planning applications. See Predefined Properties for Planning and FreeForm Applications.
`User`	Properties for User class node types. See Predefined Properties for Users Applications.

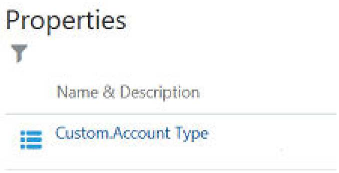
Properties with the `Core` and `CoreStats` namespaces are displayed on the Properties page regardless of whether you have registered an application. For properties with other namespaces, the Properties page will list only the properties used by registered applications. For example, if you have registered a Planning application but not a Universal application, the

Properties page will list properties with the `PLN` namespace but will not list any properties with the `Custom` namespace.

For more information, see Working with Properties.

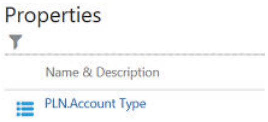
Example 22-1   Uniquely Identifying Two Properties with the Same Name

Cloud EDM includes a `PLN.Account Type` property. Suppose that you register a Universal application that has a property named `Account Type`. The `Custom` and `PLN` namespaces distinguish the two properties, as shown in the following examples:



Properties

Name & Description
Custom.Account Type



Properties

Name & Description
PLN.Account Type

System and Statistical Properties

The following properties are predefined in the system and display on the Properties page regardless of whether you have a registered application. System properties use the Core namespace. Statistical properties use the CoreStats namespace.

For more information, see:

- Working with Properties
- Understanding Property Names and Namespaces

System Properties

Property	Data Type	Level	Description
Alternate Name	String	Node	Alternate node identifier (used to identify common nodes across different node types)
Change Sign	Boolean	Relationship	Change sign indicator for mapping exports
Description	String	Node	Node description
Name	String	Node	Node name

Statistical Properties

Property	Data Type	Level	Description
# Children	Integer	Relationship	Number of direct children
# Descendants	Integer	Relationship	Number of descendants
Bottom Node	Boolean	Relationship	Determines whether node has children
Created By	String	Node	User who created the node
Created Date	Timestamp	Node	Date and time that the node was created
EDM User	Boolean	Node	Returns True if the `User.Email Address` property matches a valid user.
EDM Username	String	Node	Returns the Oracle Fusion Cloud Enterprise Data Management username if the `User.Email Address` property matches a valid user.
Last Modified By	String	Node	The user ID of the last person who updated the node or any request actions taken on the node. This includes updating the node name, description, or any of the node properties (node and relationship) that are defined for the node as well as any structural changes made to a node including moving or inserting a node across all hierarchy sets.

Property	Data Type	Level	Description
Last Modified Date	Date	Node	The date and time when the node was last updated. This includes updating the node name, description, or any of the node properties (node and relationship) that are defined for the node as well as any structural changes made to a node including moving or inserting a node across all hierarchy sets.
Level	Integer	Relationship	Level of the node in a hierarchy node set
Node ID	String	Node	Universal unique identifier (UUID) for a node Example: `31B42B2B-00FF-436F-9C95-5E7247069C81`
Node ID (32 bit)	Numeric String	Node	32 bit hash value of the Node ID Example: `3940068023` **Caution:** The hash value of a Node ID may not be unique.
Node ID (64 bit)	Numeric String	Node	64 bit hash value of the Node ID Example: `12475381469113081838` **Caution:** The hash value of a Node ID may not be unique.
Node Type	Node Type	Node	Node type of the node
Parent	Node	Relationship	Parent node name of the node in the current location
Parent Node Type	Node Type	Relationship	Parent node type
Relationship ID	String	Relationship	Universal unique identifier (UUID) for a relationship Example: `8CC68310-A7AB-44C9-89BA-026A2BDAE2BA`

Property	Data Type	Level	Description
Sibling Sort Order	Integer	Relationship	Numeric value for sorting child nodes under a parent node. This is used when **Use Custom Order** is enabled on the hierarchy set. Siblings with the same sort order number will be sorted alphanumerically.

About Property Creation

Properties are created in three ways in Oracle Fusion Cloud Enterprise Data Management:

- Predefined properties are created automatically when you register a packaged application using an application-specific adapter. The properties that get created are affected by the settings that you specify during registration (for example, how many cubes and aliases you add when registering a Planning application). These predefined properties are bound to your external applications, and they are included in imports and exports. See Understanding Application Types.
- When you register a Universal application, you can create custom properties during the registration process. Properties that you create during registration are added to a bound node type, and are included in imports and exports. See Creating a Custom Property.
- You can create properties manually from the Properties page. These properties are not bound to applications initially, and they do not get imported or exported. You can bind them to a Universal application by modifying the application registration and adding the property to a bound node type. See Creating Properties Manually.

Creating Properties Manually

In addition to the properties that get created by the system when you register packaged or Universal applications, you can manually create properties that are not bound to an external application from the properties page.

You must be a Service Administrator or have the *Owner* or *Metadata Manager* permission on an application in order to create a property. After you have manually created a property, you must add it to a node type in order to make it available in a viewpoint.

> **Note:**
>
> Properties that you manually create from the properties page are not bound to an application initially and thus do not get imported or exported. If you want to bind the property to a Universal application, you must modify the application registration and add the property to a bound node type.

To manually create a property:

1. From Properties, click **Create**.

2. Select the **Property Template** that matches the type of data the property will contain. For example, if the property values must be whole numbers, select `Custom.Integer Template`.
3. Enter the **Name**, and optionally a **Description** for the property.
4. For **Property Level**, if the property template is in the `Custom` namespace (for example, `Custom.String Template`), specify whether the property applies to nodes or to relationships between nodes.

   > **Note:**
   >
   > If the property template is not in the `Custom` namespace (for example, `PLN.Alias`), the **Property Level** field is read-only.

5. For properties with the node, node list, or string data types, optionally select the property subtype:
   - Node or Node List (see Node and Node List Data Type Subtypes):
     - Basic
     - User
   - String (see String Subtypes):
     - Basic
     - Email
     - Hyperlink
6. Specify the Default, Common, and Data Type parameters. See Understanding Property Data Type Parameters.
7. If the property uses allowed values:
   - To use a fixed allowed values list, in **Allowed Values List** select `Table` and enter the values.
   - To determine the allowed values dynamically using a node set, leave the **Allowed Values List** set to `None`, and then edit the property parameters to specify the node set and filter options for the allowed values.

   See Defining Allowed or Invalid Values for a Property.
8. Perform an action to make the property available in a viewpoint:
   - If you want to bind the property to a Universal application so that it can be imported and exported, modify the application registration and add the property to a bound node type. See Modifying a Universal Application.
   - If you want to view the property in a viewpoint but you don't want to bind the property to an application, edit a node type for the viewpoint that you want to make the property available for and add the property. See Adding, Removing, and Configuring a Node Type's Properties.

> **Note:**
>
> Manually created properties can be bound to Universal applications only. You cannot bind a manually created property to a packaged application, such as Planning.

Understanding Property Data Type Parameters

When you create or edit a property, you can configure the property parameters. The settings depend on the property and its data type. For example, string properties let you specify the minimum and maximum lengths of property values.

Many data types provide parameters for default values, minimum and maximum values, decimal precision, whether upper- or lower-case values are allowed, and so on. In addition, some data types let you configure a property to allow only values that you specify.

When creating a property manually or when registering a Universal application, select the template that corresponds to the data type of the property that you want to create (for example, `Custom.Integer`).

Parameters for All Data Types

The parameters below can be set for all property data types.

Setting	Description
Default Type **Note**: This field is not available when you create a property during Universal application registration. Enter the specified value in Default Value or leave it blank if no default value is used.	Specify whether or not the property has a default value. - **None**: No default value is used for the property. - **Specified**: A default value is used for the property. Specify the value in Default Value. - **Derived**: The property's default value is calculated from an expression. Click **Define Expression** fx to open the expression builder and create an expression to determine the value of the property. See Using Expressions to Define Custom Business Logic. **Note:** This option is available only when editing property parameters. - **Derived and Stored**: The property's default value is derived from an expression during a request, import, or viewpoint load and then stored on the node. Click **Define Expression** fx to open the expression builder and create an expression to determine the value of the property. See Using Expressions to Define Custom Business Logic. **Note:** This option is available only when editing property parameters. - **Calculate and Store**: Available on `Core.Name` and `CoreStats.Parent` properties only. When request items are created, the node name or parent value is calculated from an expression and then stored on the node (for Name) or on the request item (for Parent). Click **Define Expression** fx to open the expression builder and create an expression to determine the value of the property. See Using Expressions to Define Custom Business Logic. See Calculated and Stored Properties.
Default Value	If the property uses an Allowed Values list, select a default value from that list. Otherwise, enter a default value for the property.
Rename on Update (`Core.Name` property when the Default Type is Calculate and Store)	For Calculated and Stored name values, specifies that the name of the node should be recalculated and stored when the node is directly changed by a request action. See Calculating and Storing the Name of a Node.

Setting	Description
Rederive on Update (All properties that support a Default Type of Derived and Stored)	For Derived and Stored property values, specifies that the property value should be rederived and stored when the node is directly changed by a request action. See Derived and Stored Properties. **Note:** You cannot select Rederive on Update if Lock On Commit is enabled.
Editable **Note**: This field is not available when you create a property during Universal application registration. After the property is created by the registration, you can access this field in the property parameters.	Specify whether or not the property is able to be edited by users.
Lock on Commit	Specify that this property can only be updated on a newly added node. If this is enabled, after the request to add a node has been committed the value for the property is locked and can no longer be updated. **Note:** This option is available only when editing property parameters. Lock on Commit is available for node-level properties only. You cannot select Lock on Commit if Rederive on Update is enabled. **Caution:** Exercise caution when using Lock on Commit so that you don't unintentionally prevent users from being able to update a property that should be updated. For example, if you enable Lock on Commit on `Core.Name`, users will be unable to rename the node after it has been created. Or, if you enable Lock on Commit on a property but you do not specify a value for the property, users will be unable to update the property with a value after it has been committed.

Setting	Description
Inheritance	The type of inheritance the property uses. See Inheriting Properties. • None: Property values are not inherited from another node. • Positional: Property values are inherited from an ancestor node. **Note:** You can override inheritance parameters at the application level only.

Note:

For node data type properties only, the Default Value that you enter in conjunction with the Node Set, Allowed Node Types, and Selection criteria property parameters (see Node and Node List Data Type Parameters) will be converted to a node value as follows:

- If only one node is found then it is used as the property value.
- If more than one node is found, the first node in the order of the Allowed Node Types is used as the property value.
- If a node is found but it is not one of the allowed node types or does not meet the selection criteria, the string itself is used as the property value. You will be unable to validate the viewpoint with a string value in a node data type Default Values field.
- If no nodes matching the string and other property parameters are found, the string itself is used as the property value. You will be unable to validate the viewpoint with a string value in a node data type Default Values field.

For more information on parameters for specific data types, see:

- Boolean Data Type Parameters
- Float and Integer Data Type Parameters
- List and String Data Type Parameters
- Node and Node List Data Type Parameters
- Numeric String Data Type Parameters
- Sequence Data Type Parameters

Boolean Data Type Parameters

Setting	Description
Boolean Display Type	How the property displays true and false values.

Setting	Description
Boolean True Value	The value that will represent true in imports and exports. Typically this should be the value that the external application uses for true.
Boolean False Value	The value that will represent false in imports and exports. Typically this should be the value that the external application uses for false.

Float and Integer Data Type Parameters

Setting	Description
Minimum Value	A minimum value for the property value
Maximum Value	A maximum value for the property value
Number of Decimal Places - Float Only	Number of decimal places to use - Float only

List and String Data Type Parameters

Setting	Description
String Case	Which case to use for text: mixed case, upper, or lower
Minimum Length	Minimum length for the text
Maximum Length	Maximum length for the text **Note:** For Planning applications, the Maximum Length value is 1000 because the property configurations cannot be changed.
Invalid Characters	Characters that are invalid if used in the text
Invalid First Characters	Characters that string data type properties cannot begin with. (String data type properties only)
Invalid Values	Values that string data type properties cannot be equal to. (String data type properties only) Click + to enter the values manually, or click **Action** ⋮ to import, export, or clear values from the list. See Defining Allowed or Invalid Values for a Property. Click **X** to remove a value from the list.
Allowed Values List	Determines the allowed values for the property: • **None**: The property does not use allowed values. • **Table**: The property uses a fixed set of allowed values. Enter the list of allowed values in the table. See Defining Allowed or Invalid Values for a Property. • **Dynamic**: The allowed values for the property are dynamically determined using a specified node set and filter options. See Defining Allowed or Invalid Values for a Property.

String Subtypes

For string data types, you can specify a subtype that provides additional capabilities and optionally enables additional syntax validations to be performed. For the syntax validations, select **Error** to enforce the validation (see details in the table below) or select **Skip** to ignore that validation.

Setting	Description	Optional Syntax Validations
Basic	Basic string information without any special capabilities or validations.	None
Email	Email address	**Email Syntax Required**: Select `Error` to validate that the string has valid email syntax.
Hyperlink	Hyperlink information that provides these additional capabilities: • Users can click the link and navigate to the link destination. **Note**: When a user is in the context of a request, the hypertext link is displayed in plain text so that it can be edited. Click **Actions** ⋮ and select **Open Hyperlink** to navigate to the link target. • Users can right-click the link and perform standard link operations, such as Open in New Tab, Save Link As, etc.)	• **Secure Protocol (HTTPS) only**: Select `Error` to validate that the link begins with `https://` **Note**: By default, `https://` is used for links when the protocol is not provided. So if this validation is set to Error, `oracle.com` will not fail (since it defaults to `https://oracle.com`) but `http://oracle.com` would. • **Prevent User Information**: Select `Error` to validate that the link does not contain any user information (such as a user name or password). • **Link Syntax Required**: Select `Error` to validate that the string can only contain valid hyperlink syntax. **Tip**: If you set this validation to `Skip`, users are able to enter non-hyperlink text, such as "None" or "Not available".

Node and Node List Data Type Parameters

Setting	Description
Assigned Node Set	The node set that contains the nodes that you are referencing for this property **Note:** For basic subtypes, the node set that you are referencing must be in the same application as the node set with the node data type that you are referencing from. For user subtypes, the node set can be any active node set that uses the User or User Rollup node type class. See Node and Node List Data Type Subtypes.
Allowed Node Types	The node types within the node set that contains the nodes that you are referencing for this property **Note:** For user subtypes, the node type must have the User class.
Bottom Level Only	Determines whether to include just bottom level nodes in conjunction with the selection criteria fields listed below.
Selection Criteria Property	List of all properties that are not node data type properties in all of the allowed node types. Used with Selection Criteria Operator and Selection Criteria Value fields to select specific nodes. **Caution:** For the Selection Criteria property, because node data type properties store a *reference* to a node and not a specific *location* for the node, if you are referencing a property that contains positional information (such as inheritance or derived default values) and the node that you are referencing from is shared, the system uses the first location that it finds the referenced node in, which may not be the node that you were expecting.

Setting	Description
Selection Criteria Operator	Includes `Equal` and `Not Equal`. Used with Selection Criteria Property and Selection Criteria Value fields to select specific nodes.
Selection Criteria Value	Enter a text value to use with Selection Property and Operator fields to select specific nodes. Blank values are allowed.

After you select an **Assigned Node Set** and **Allowed Node Types**, you can filter the node selection down further by using the **Bottom Level Only**, **Selection Criteria Property**, **Selection Criteria Operator**, and **Selection Criteria Value** fields. For example, you can specify that an application-level override applies only to bottom level nodes where the Intercompany Flag is set to True.

Node and Node List Data Type Subtypes

For node and node list data type properties, you can specify a subtype that provides additional capabilities. The subtype can be changed at the application level only.

- **Basic**: Used to reference nodes from any node type in the current application only. The **Assigned Node Set** and **Allowed Node Types** must be from the current application.
- **User**: Used to reference nodes with the User or User Rollup node type in any Users application. The **Assigned Node Set** can be from any Users application, and the **Allowed Node Types** must be User or User Rollup. See Working with Users Applications.

Numeric String Data Type Parameters

Setting	Description
Minimum Length	Minimum length for the text (must be either zero or a positive number).
Maximum Length	Maximum length for the text (must be either zero or a positive number).
Padding	Total length (0-40) of the string to be zero padded up to, with 0 meaning no padding is added. For example, if you set the Padding to 10 and your numeric string value is 7 digits long, three zeroes are prepended to the string to get to 10 places.

Sequence Data Type Parameters

Setting	Description
Starting Value	The initial starting value for the sequence. **Note:** This is not a minimum value. Although all subsequent derived sequence values will be greater than this initial starting value, you can manually define a sequence value less than the Starting Value.
Step Value	The step by which to increment the sequence. For example, if your Starting Value is 1000 and your Step Value is 5, the next sequence value will be 1005.
Padding	Total length (0-40) of the sequence value to be zero padded up to, with 0 meaning no padding is added. For example, if you set the Padding to 10 and your sequence value is 7 digits long, three zeroes are prepended to the value to get to 10 places.

Setting	Description
Last Sequence Value	Displays the most recent value for the sequence, or `N/A` if no sequence values have been generated for this property. You can manually enter the last sequence value, and you can reset it back to the initial starting value for the property. See Viewing, Editing, and Resetting Sequence Values.

Sequence data type properties are formatted as numeric strings. That is, they support integer values up to a maximum length of 40 characters.

Sequence data type properties do not support positional inheritance or a Default Type other than None.

Inheriting Properties

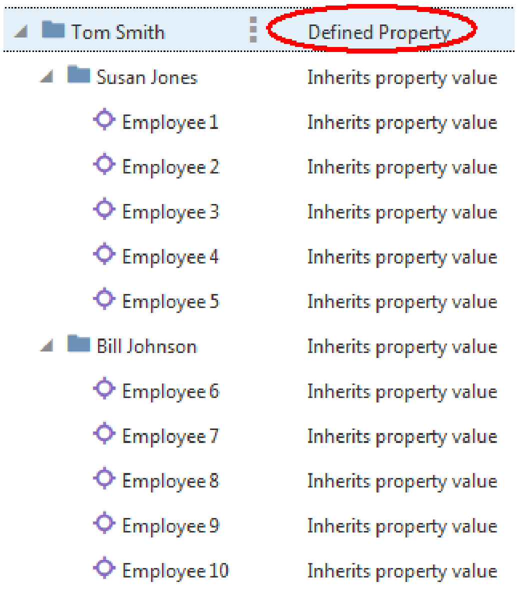
Property inheritance enables you to define a property value for a node which is automatically inherited by all of the node's descendants.

Configuring properties to inherit their values reduces the manual effort of maintaining properties and ensures consistency of values for entire branches of nodes. Inheritance of a property is based on a nodes position in a hierarchy. If a node has an inheriting property, the system looks up the tree of ancestors until it comes to the first ancestor that has a defined value for the property. That defined value is the value that is inherited by the descendant nodes.

Videos

Your Goal	Watch This Video
Learn about inheriting properties.	Overview: Inherited Properties in Enterprise Data Management Cloud

To determine the value to inherit, the system looks up the hierarchy until it finds the first ancestor node that has a defined value for the property. For example, Tom Smith is the director of the Human Resources dept. He has two managers who report to him and each manager has 5 employees. All employees in this department need to be assigned to the HR cost center. With the cost center property set as an inheriting property, you can define the cost center value as "HR" for Tom Smith and all of the employees under him will inherit the "HR" value.



Tom Smith	Defined Property
Susan Jones	Inherits property value
Employee 1	Inherits property value
Employee 2	Inherits property value
Employee 3	Inherits property value
Employee 4	Inherits property value
Employee 5	Inherits property value
Bill Johnson	Inherits property value
Employee 6	Inherits property value
Employee 7	Inherits property value
Employee 8	Inherits property value
Employee 9	Inherits property value
Employee 10	Inherits property value

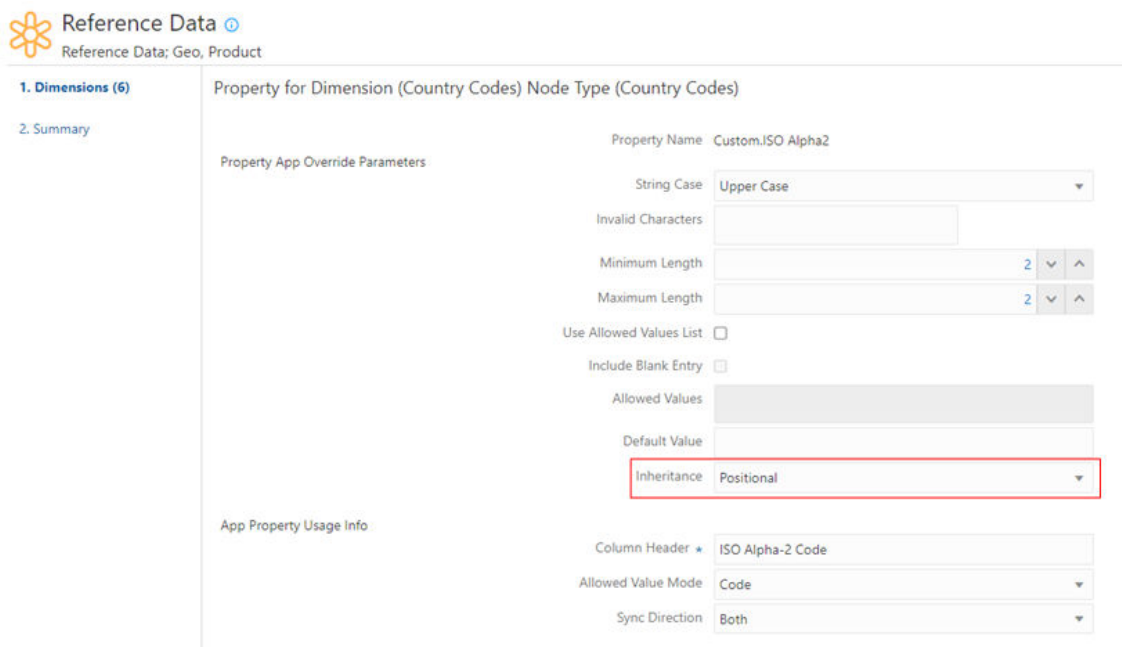
Considerations

- Property inheritance can be used only in hierarchy viewpoints.
- Inheritance stops at the top of a node set. This can cause different inherited property values in different node sets (for example, in a maintenance viewpoint that starts at a lower level than the full viewpoint).
- Inherited property values can be overridden at a lower level to handle exceptions.
- For both node and relationship-level properties, in a hierarchy with shared nodes, a node can be in multiple positions and can inherit different values for each position . For relationship level properties, a node can have different values even if its parent is the same in multiple locations because it could be under a shared ancestor node. For more information, see Property Inheritance and Shared Nodes.
- For the Planning and Planning Modules application types, these properties are automatically enabled by default with inheritance:
  - PLN.AccountType
  - PLN.Aggregation
  - PLN.BaseCurrency
  - PLN.DataType
  - PLN.ExchangeRateType
  - PLN.PlanType
  - PLN.SkipValue
  - PLN.SourcePlanType
  - PLN.TimeBalance
- Properties used in Universal applications can be configured with inheritance when you register or modify an application. For more information, see Registering a Universal Application and Modifying a Universal Application.
- Inheritance is set in the property inspector. See Editing Property Parameters.

> **Caution:**
>
> Property inheritance is not enabled for and should not be used in Oracle Financials Cloud General Ledger applications, since they export data for the segment values from the list viewpoint.

Configuring Property Inheritance for Universal Applications

You can configure inheritance for a property when you register or modify a Universal application.



Reference Data
Reference Data; Geo, Product

Property for Dimension (Country Codes) Node Type (Country Codes)

Property Name: Custom.ISO Alpha2

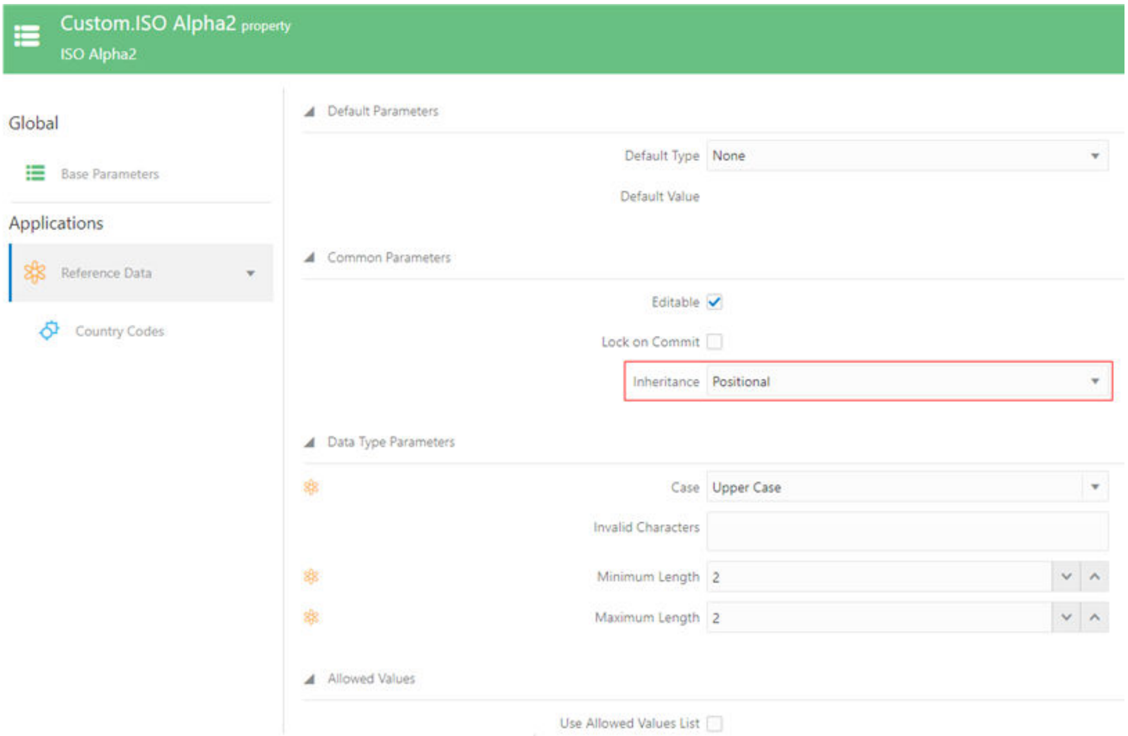
Property App Override Parameters

String Case	Upper Case
Invalid Characters	
Minimum Length	2
Maximum Length	2
Use Allowed Values List	☐
Include Blank Entry	☐
Allowed Values	
Default Value	
Inheritance	Positional

App Property Usage Info

Column Header *	ISO Alpha-2 Code
Allowed Value Mode	Code
Sync Direction	Both

You can also configure inheritance for a property from the property inspector:



Default Parameters

Default Type: None

Default Value:

Common Parameters

Editable ☑

Lock on Commit ☐

Inheritance: Positional

Data Type Parameters

Case: Upper Case

Invalid Characters:

Minimum Length: 2

Maximum Length: 2

Allowed Values

Use Allowed Values List ☐

For more information, see:

- Adding or Modifying a Node Type for a User Defined Dimension
- Working with Node Properties

Property Inheritance and Shared Nodes

Inheritance provides a way to define a default value based on the position of a node within a hierarchy structure.

Shared nodes are nodes that are in a hierarchy structure in more than one position. This can lead to shared nodes having different inherited values for node-level and relationship-level properties.

> **Note:**
>
> Defined values cannot be different by position since they are defined either at the node for node-level properties or at the node and parent for relationship-level properties.

Import

During an import, all values are imported as defined values. The import process then clears values to allow inheritance and defaults to be used. This is done because the import may not be in a top down hierarchy order and the processing cannot be done in one pass.

For non-shared nodes (excluding the top node) the import process determines if the value is equal to the inherited value or the default value and if equal, the value is removed.

Because shared nodes can have different inherited values based on their position the import process does not clear values for shared nodes even if all the shared positions would inherit the same value.

Export

During an export, the property value is repeated for shared nodes unless this logic is overridden by a system-specific requirement (for example Planning).

Derived Properties

Derived properties are properties whose default value is calculated from an expression that you define.

Derived properties can be used to reduce maintenance of property values for nodes and help ensure data integrity of those values. You'll use the Expression Builder to define the expression. For information on how to define expressions, see Using Expressions to Define Custom Business Logic.

Considerations

- You can derive the value for properties with the following data types:
  - Boolean
  - Date
  - Integer
  - List
  - Node
  - Numeric String
  - String
- You cannot derive the default value for properties in the Core or CoreStats namespaces.
- When building an expression, you can use positional information such as `bottom`, `parent`, `ancestors`, and `siblings` regardless of the property level (node or relationship) or by referencing a relationship property or a property that uses inheritance.
- When using positional information in your expression, the values may not be consistent at the node or relationship level. Note the following:
  - The derived default value can be different by the location of the node within a viewpoint for either node or relationship-level properties.
  - The derived default value for the node can be different between different hierarchy viewpoints even if they are based on the same hierarchy set.
  - The derived default for a node in a list viewpoint will be different from the derived value for that node in a hierarchy viewpoint.

Validations

Derived property values are validated on all nodes that are updated before a request is submitted or committed. Derived property validations run for requests when you:

- Validate all request items
- Validate a selected request item

- Submit a request by any of these methods: interactive, subscription, or import
- Approve a request

> **Note:**
>
> Derived property validations run only for nodes with a request item in the request.

Validation issues for derived properties can be resolved in several ways including:

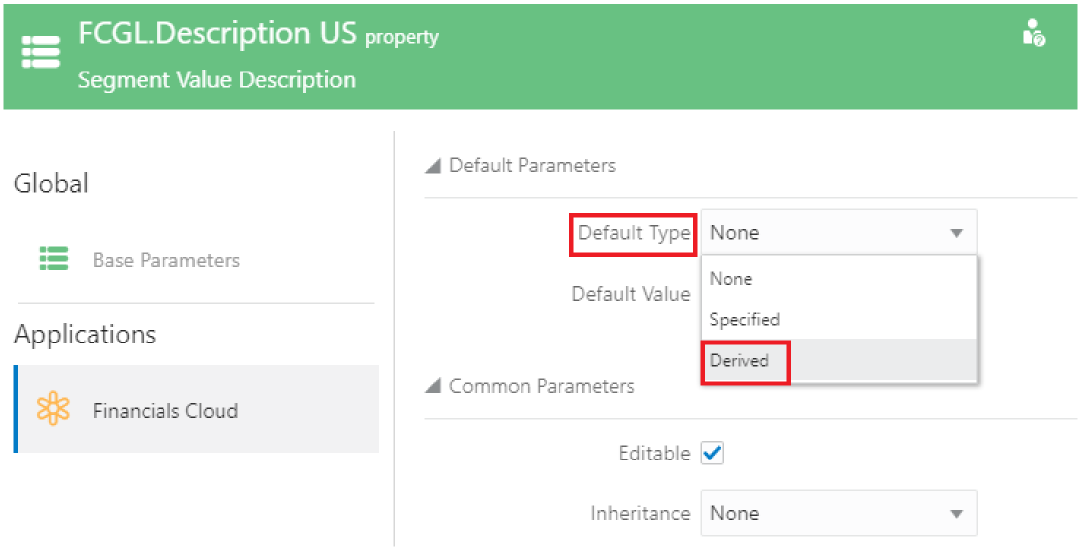
- Override the derived property with a defined value if editable
- Modify a different property that the derived property depends on
- Move or remove the node in the viewpoint
- Delete the request item or item instance

For more information see Validating Requests.

Defining a Derived Property

To define a derived property expression:

1. Inspect a property definition that uses a Boolean, Date, Integer, Float, List, Node, or String data type. See Inspecting Properties.
2. Select an application for which you have *Owner* or *Metadata Manager* permission.
3. Click **Edit**.
4. From **Default Type**, select **Derived**.



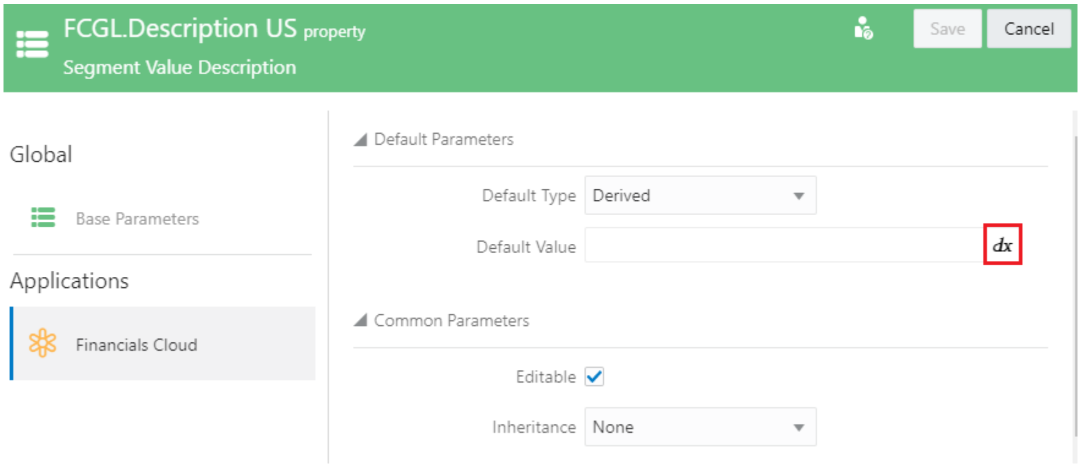
FCGL.Description US property
Segment Value Description

Global
Base Parameters
Applications
Financials Cloud

Default Parameters
Default Type: None
Default Value
None
Specified
Derived

Common Parameters
Editable ☑
Inheritance: None

5. Click the Expression icon in the **Default Value** text box.



FCGL.Description US property
Segment Value Description

Default Parameters

Default Type: Derived

Default Value:

Common Parameters

Editable: ☑

Inheritance: None

6. Define an expression to calculate the property and click **Apply**. For information on how to define expressions, see Using Expressions to Define Custom Business Logic.
7. Click **Save**.

> **Note:**
>
> To remove a derived expression for a property, repeat steps 1-4 and change **Default Type** from **Derived** to **None** or **Specified**.

Derived and Stored Properties

Derived and Stored properties are properties whose default value is calculated during a request from an expression that you define and then stored on the node when a request is completed.

Derived and Stored default property values are indexed so that they can be referenced when querying viewpoints. See Querying a Viewpoint. You can also use them to derive a value for a property in a hierarchy viewpoint and then access that stored value for the property in a list viewpoint.

> **Note:**
>
> Properties with the Derived and Stored default type are able to be queried only after a value has been either defined or derived and then stored for that property on a node. This is true even if you join the Derived and Stored query filter with an indexed property query filter using an AND statement. See Querying a Viewpoint.

Considerations

- You can derive and store default values for node level properties only.
- You can derive and store values for the Core.Description property, but you cannot derive and store values for any other properties in the Core or CoreStats namespaces.
- You cannot derive and store values for properties with the Sequence data type.

- You can set the Derived and Stored default type at the application and node type levels in the property inspector. See Editing Property Parameters.
- Derived and Stored property values are processed during a request, an import, or when loading a viewpoint.

> **Note:**
>
> Changing a property's default type to Derived and Stored does not automatically populate existing nodes with the stored value. Follow these steps to populate existing nodes with derived and stored values:
>
> 1. Change the property default type to Derived and Stored.
> 2. Download a viewpoint that contains all of the information that is needed to derive the value for that property (for example, if the derived value expression contains positional information, download a hierarchy viewpoint). See Downloading a Viewpoint.
> 3. **Recommended**: Remove the columns in the downloaded file that you are not updating.
> 4. Perform a viewpoint load in Replace mode using the downloaded file (see Working with Viewpoint Loads).

- You can clear the current value of a derived and stored default value in order to derive and store the value again:
  - For interactive requests, in the property that you want to clear and derive again click **Actions** ⋮, and then select **Clear**.
  - In a request file upload, use the `<clear>` keyword. See Request Load File Format.

Defining a Derived and Stored Property

1. Inspect a property definition that uses a Boolean, Date, Float, Integer, List, Node, Numeric String, or String data type. See Inspecting Properties.
2. Select an application or a node type, and then click **Edit**.
3. From **Default Type**, select **Derived and Stored**.
4. Click the Expression icon in the **Default Value** text box.
5. Define an expression to calculate the property and click **Apply**. For information on how to define expressions, see Using Expressions to Define Custom Business Logic.

> **Note:**
>
> If you change the Default Type from Derived to Derived and Stored (or the reverse), you do not have to re-enter the expression to calculate the property value.

6. **(Optional)** Select **Rederive on Update** to specify that the value for property should be recalculated and stored whenever a request action is performed directly on the node in a request. The recalculated value is stored when the request has been completed and closed.

> **Note:**
>
> The value for the property is recalculated only when a request action is performed on the node itself. If the calculation logic includes elements that are external to the node (such as the node's ancestors) and those are changed, the node value is not automatically recalculated. You must perform a clear action on the node in order to recalculate and store the new value.
> The value for the property is also not recalculated and stored if **Lock on Commit** is enabled for the property. See Understanding Property Data Type Parameters.

7. Click **Save**.

Derived and Stored Property Processing

This topic discusses how properties with the Derived and Stored default type are processed when nodes are added or updated in a viewpoint.

> **Tip:**
>
> In the processing sections below, a *stored* value refers to a value that is either defined or one that has already been derived and then stored on a node.

Interactive Requests

For properties with the Derived and Stored default type, values are derived during a request and then stored after the request is committed and before it is completed. While the request is in process, for any Derived and Stored properties that do not already have a stored value, the value for that property is dynamically calculated based on the expression that you defined in **Default Value**. The property origin during request processing is **Derived**.

After the request is committed and before it is completed, all properties with the Derived and Stored default type are checked. The properties that do not have a stored value or a property update action with a user defined value are updated with the value from the expression that you defined, and the value is stored as a **Derived and Stored** value.

> **Note:**
>
> These property update actions are added to the request and can be viewed in the request inspector. See Inspecting Request Items.

The process of calculating and populating the derived value in a property is an internal system property update action and therefore is not subject to security and validations. In other words, when a user adds or updates a node the system populates all properties with the Derived and Stored default type for that node that do not already have defined values even if:

- The user does not have write access on that property
- The property is hidden in the viewpoint
- The property is not defined as editable
- The property has Lock on Commit enabled

However, if a user clears or manually enters a defined value the usual security and validations are applied.

If you clear the value on a Derived and Stored property, the system displays the currently calculated value based on the expression that you defined and the property origin is **Derived**. Clearing the value provides a way to recalculate the value of Derived and Stored properties if the information used in the expression has changed. When the request is committed, the calculated value is stored as a **Derived and Stored** value.

> **Note:**
>
> For copy and model after operations the Derived and Stored values are not copied from the source nodes. The values for the new nodes are calculated and stored from the expression that is defined in **Default Value**.

Request File Load

For requests from a file load (that is, uploaded request files in an interactive request, compare requests, copy or model after requests, subscriptions, consolidation requests, and merge imports), nodes that do not already have stored values for a Derived and Stored property are updated as follows:

- If the request file contains a defined value for the Derived and Stored property, the value is updated as normal. This is a user action and security and validations apply.
  - The `<blank>` keyword updates the property with a defined blank value.
  - The `<clear>` keyword clears the existing value (including blank values) and calculates the derived value.

    > **Tip:**
    >
    > If you add a property with the Derived and Stored default type to an existing node type, an easy way to populate values for that property on existing nodes without making any other changes is to update that property with the `<clear>` keyword using a request file or viewpoint load.

- If the request file does not contain a defined value for the Derived and Stored property, then the property is updated with a derived value based on the expression that you defined in **Default Value** and stored as a Derived and Stored value. This is an internal system action and security and validations do not apply.

  > **Note:**
  >
  > Derived and Stored property values for nodes in a request file are updated even if that property isn't included in the file. For example, if your file contains updates for the Description property only, any nodes in that file that do not have a stored value for a property with a default type of Derived and Stored will also have that property updated with the calculated value, even if the user doesn't have write access to that property or the property is hidden in the viewpoint.

Imports and Viewpoint Loads

Merge Mode

Imports and viewpoint loads in merge mode are processed the same as Request File Load, above. All properties with a default type of Derived and Stored (including unbound properties) that do not already have a stored value are updated with the calculated value during the import or viewpoint load. That calculated value is then stored for that property.

You can use the `<clear>` keyword to calculate Derived and Stored values in imports and viewpoint loads in Merge mode.

Reset and Replace Modes

For imports in both Reset and Replace modes, unbound properties with the Derived and Stored default type are not modified. Only bound properties in Universal applications can be updated with Derived and Stored values. The values are updated following the same processing as Request File Load, above.

For viewpoint loads in Replace mode, only the Derived and Stored properties that are included in the viewpoint load file are updated with a derived and stored value. If your viewpoint load file does not contain a column for a property with a Derived and Stored default type, that property is not updated during the viewpoint load.

Using the <clear> Keyword in Reset and Replace Modes

For viewpoint loads in Replace mode, you can use the `<clear>` keyword to calculate Derived and Stored values for properties.

For imports in both Reset or Replace modes, you can use a clear keyword to calculate Derived and Stored values for properties if you have set one up in your import options. See Managing Import Options.

Calculated and Stored Properties

For certain properties, you can define an expression to calculate the value and then store that value for each request action.

The Calculate and Store option is available for these properties only:

- `Core.Name`. See Calculating and Storing the Name of a Node.
- `CoreStats.Parent`. See Calculating and Storing the Parent of a Node.

You configure the Calculate and Store option as a parameter override to the Default Type at the application or node type level. See Editing Property Parameters.

> **Tip:**
>
> Calculated and Stored properties differ from Derived and Stored properties in that they are used to calculate the value for the Name or Parent properties in the context of a request. For the Parent property, the result is stored on each request action, not on the node.

Calculating and Storing the Name of a Node

You can define an expression to calculate a node name during a request and then store that value when the request is completed and closed.

The calculated name can be based on the properties of the node, or the node's position in a hierarchy.

Example Use Cases

- You can calculate a node name based on a sequence number. For example, you can concatenate a prefix like "CC_" with a sequence so that when you add nodes they are automatically named CC_0034, CC_0035, etc.
- You can calculate a node name based on properties. For example, you can concatenate custom node data type properties together. Then, as you define the properties, the node name is automatically calculated. So, for custom property 1 = 120, custom property 2 = 200, and custom property 3 = 340, a node name of 120-200-340 is automatically calculated.

> **Tip:**
>
> You can also set up expressions on the node data type property to derive their values so that if you add a node name of 120-200-340 instead, the value for each property is derived as 120, 200, and 340 respectively. See Derived and Stored Properties.

- You can calculate a node name based on hierarchy structural information such as parents, ancestors, and siblings. For example, you can concatenate a parent name plus the node description.
- You can also use calculated node names to customize the default name when you add a new node. For example, when a user adds a node in the Entity node type, instead of the name defaulting to "New Entity Node", you could return a calculated name of "CC_####" and have your users manually fill in the appropriate values for ####.

Considerations

- You can calculate a node name during the following operations:
  - Adding nodes interactively
  - Copying or modeling after a node
  - Loading request files
  - Synchronizing nodes via a subscription
- During an Add action in a request, the node name is recalculated every time the node is changed.

> **Note:**
>
> When a node name is calculated based on a sibling list (for example, by finding the last sibling and adding 1 to it), be aware that the node name may change if other siblings are added after it and then the original node is updated.
>
> Also, depending on the request context, using the `node.previousSibling` object in an expression to calculate node names can have unexpected results. For best results, use `node.siblings.max` instead.

- When displaying a calculated node name in a request:
  - If the node name expression returns a value, that value is displayed with an origin of Derived and Stored.
  - If the expression returns a null value, then the calculated name keyword `<cn ####>` (where `####` is a hash of the request item's created timestamp) is displayed. See Understanding the Calculated Name Keywords.
- After a request has been completed and closed, the calculated node name is stored and the origin displays Derived and Stored.
- You can overwrite the node name by manually entering a name. The origin displays Defined.
- For nodes being added in a request, use the **Clear** action to clear a manually-entered name and go back to the calculated name, and use the **Reset** action to set the current location as the primary location (see Understanding Primary Locations) and to recalculate the node name based on the expression. See Clearing and Resetting a Calculated Value.
- Validations that are run in the context of a request that contains calculated node names will use the currently calculated name. That means, for example, that if the expression currently returns a null, the node name for the validation is "<cn ####>".
- When searching for nodes in a request, the current name in the Add action is used for new nodes that have calculated names. That means, for example, that if the expression currently returns a null, you can search on "<cn ####>" in the node name.
- When loading request files, use the `<cn>` or `<cn ####>` keyword in the name column of a request load file (where an expression is configured for the `Core.Name` property) or leave the name column blank to calculate the value for that node name. See Request File Processing with Calculated Names.

Defining a Calculated and Stored Property

1. Inspect a `Core.Name` property definition.
2. Select the application or node type that you want to calculate the node name for.
3. From Default Type, select **Calculate and Store**.
4. In Default Value, click the **Expression** button *fx*.
5. Define an expression to calculate the node name and click **Apply**. See Using Expressions to Define Custom Business Logic.
6. **(Optional)** Select **Rename on Update** to specify that the name value for the node should be recalculated and stored whenever a request action is performed directly on the node in a request. The recalculated name value is stored when the request has been completed and closed.

> **Note:**
>
> The value for the name is recalculated only when a request action is performed on the node itself. If the calculation logic includes elements that are external to the node (such as the node's ancestors) and those are changed, the name is not automatically recalculated. You must perform a clear action on the name in order to recalculate and store the new name value.
> The name value for the property is also not recalculated and stored if **Lock on Commit** is enabled for the property. See Understanding Property Data Type Parameters.

Understanding Primary Locations

If you are adding a node and then inserting the same node in multiple other locations, then internally the system designates the first location in a hierarchy where the node is added as the primary location and uses that to calculate the node name. All subsequent places where the node is inserted will use the calculated name from the first location.

> **Note:**
>
> If the first location where the node is added is in a list, the primary location is set the first time it is inserted into a hierarchy.

For example, suppose you had an expression that calculated the node name by concatenating the node's parent's name with a property on the node (for this example, that property has a value of ABC). When you add the node under parent 111, the name is calculated as 111-ABC. If you then insert the node under parent 222, the name is still 111-ABC, because 111 is the primary location.

There are multiple ways to change a node's primary location:

- If you move a node in the request from its primary location, the new location becomes the primary location and the name is recalculated.
- If you remove the node from the primary location during the request, the next location where it was inserted becomes the primary location.
- If you have a node in multiple places and you want to manually change which location is the primary location, use the **Reset** action. See Clearing and Resetting a Calculated Value

Clearing and Resetting a Calculated Value

Use **Clear** to clear a manually defined name and go back to the calculated name. The Clear action is available only for nodes that were added in this request with a default type of Calculate and Store that have a name that was manually entered.

Use **Reset** to set the primary location for the calculated name to the current location (see Understanding Primary Locations) and to recalculate the node name based on the current location. The Reset action is available only for nodes that were added in this request with a default type of Calculate and Store that do not have a name that was manually entered.

You can also use Reset to recalculate a node name if you have taken an action outside of the node that would affect the node name (for example, if you change a Parent property and you want to recalculate the node name based on the new property).

Validations and Calculated Node Names

When a request is validated, the current name in the Add action is used for new nodes that have calculated names, with one exception: If the current name is one of the keywords (`<cn>` or `<cn ####>`), the minimum and maximum length validations are skipped. This enables a submitter to submit a request with one of the keywords in the name so that an approver or enricher can add the required information to calculate the name.

The minimum and maximum length validations are run when requests are approved or committed.

Calculated Names and Subscriptions

When you set up a subscription using a calculated name, the name is calculated in the source viewpoint and the resulting name is passed in the subscription requests to the target viewpoints. If you want to have the node name calculated in the target viewpoint, you must set up a property transformation in the node type converter for the subscription (see Working with Node Type Converters) to transform the name to a blank value or one of the calculated name keywords (`<cn>` for Adds, or `<cn ####>`, where #### is a unique identifier string, for Adds with Inserts). See Understanding the Calculated Name Keywords.

> **Note:**
>
> If you want to use an Alternate Name when available, transform the name to a blank value. If you always want to use the calculated name in the target viewpoint, transform the name to one of the keywords.

Request File Processing with Calculated Names

Note the following considerations when uploading a request file that uses calculated node names.

Request files with calculated names use the same format as other request files (see Request Load File Format) with the exception of the Name column. If a property has been set up for Calculate and Store, you can leave the name column blank, or you can use the `<cn>` or `<cn ####>` keyword. See Understanding the Calculated Name Keywords.

Understanding the Calculated Name Keywords

The following keywords `<cn>` or `<cn ####>` (where #### is a string that is used as a unique identifier) can be used when adding a node with a calculated name in a request file.

- `<cn>`: Used to calculate node names for Add actions only.
- `<cn ####>`: Provides a unique identifier for the calculated name for Adds and Inserts. When a request contains both Add and Insert action, use the `<cn ####>` keyword with a unique identifier (for example, `<cn 1234>`) on the Add action, and then use that same keyword with the identifier on the Insert actions. This enables the system to calculate the name for the node during the Add operation (see Understanding Primary Locations), and then use the same name for the insert actions.

Note:

The Add action must come first in the load file.

Name Column Processing Overview

The following list provides a brief overview of how the Name column is processed with blank values or the `<cn>` or `<cn ####>` keywords when Calculate and Store is set up for a property:

Blank values:

- If an Alternate Name is provided in the file, the alternate name is used for the node.
- If an Alternate Name is not provided in the file:
  - For Adds the name is calculated from the expression.
  - For Updates the name is calculated from the expression if the node doesn't already exist.
  - For Inserts the row is skipped.

<cn>:

- For Adds the name is calculated from the expression, even if an alternate name is provided.
- For Updates the name is calculated from the expression if the node doesn't already exist.
- For Inserts the row is skipped.

<cn ####> (where #### is a string that is used as a unique identifier):

- For Adds the name is calculated from the expression, even if an alternate name is provided.
- For Updates the name is calculated from the expression if the node doesn't already exist.
- For Inserts, if the `<cn ####>` identifier string matches the identifier string of a previous Add action then the calculated name is used for the insert. If the `<cn ####>` identifier string does not match a previous Add action, the row is skipped.

Note:

If your request file does not contain a column for Action Code, then the action code for all of the request items in the file is set to Update.

Calculating and Storing the Parent of a Node

You can define an expression to calculate a parent and then store that value for each request action where the parent is unknown.

For hierarchy viewpoints only, you can calculate and store the value of the `CoreStats.Parent` property when adding or inserting a node during these operations:

- Loading a request file when the parent is undefined
- Creating request items from compare differences when the source parent is undefined (when the source viewpoint is a list and the target is a hierarchy)

- Generating subscription request items when the source parent is undefined (when the source viewpoint is a list and the target is a hierarchy)

Considerations

- When creating the expression to calculate the parent, you can select from these fields for the `node` object in the expression builder:
  - `dimension`
  - `hierarchySet`
  - `name`
  - `nodeSet`
  - `nodeType`
  - `properties` (Node properties only. Relationship properties are not available when calculating the parent.)

  See Using Expressions to Define Custom Business Logic.
- The expression for the calculated parent must refer to an existing node in the target viewpoint. You cannot define an expression to create a parent node.
- When creating request load files, use the `<Unknown>` keyword in the parent column of a request load file (where an expression is configured for the `CoreStats.Parent` property) to calculate the value for that parent. Leaving a blank in the parent column will cause the node to become a top node.
- For requests generated from compare differences and subscriptions, when request actions are created for a node in a target viewpoint which has the `CoreStats.Parent` property and the parent is unknown, the system uses the `<Unknown>` keyword in the parent column of the generated request file attachment to indicate that the value for the parent should be calculated.
- The node type of the calculated parent is calculated as follows:
  - If only one node type in the hierarchy set allows children, that node type is used for the parent.
  - If more than one node type in the hierarchy set allows children, those node types are searched for a node with the same name as the calculated parent:
    - If a node with the same name as the calculated parent is found in only one of those node types, that node type is used.
    - Otherwise, the node type of the child node is used for the calculated parent.

Working with Sequence Properties

Sequence properties use the sequence data type, which automatically generates incrementing sequence numbers for node property values.

Nodes using a sequence property will get a value when they are added to a viewpoint. If a node is already in a viewpoint, it will get a value the next time it is updated.

For example, you can use a sequence property to generate unique record IDs for a data warehouse dimension, or you could use a sequence property with a custom validation to ensure that a new node name uses that sequence (such as `CC_001005`).

Considerations

- Sequence properties can be defined as node level properties only.
- Sequence properties must have a Default Type of None, and they do not support inheritance.
- You can manually override sequence properties, and you can clear the current value (including blank values) to get the next number in the sequence.
- You can view the last sequence value in the property inspector, and you can edit the data type parameters to manually enter a sequence value or to reset the sequence value back to the initial starting value for the property. See Viewing, Editing, and Resetting Sequence Values.
- You can edit the data type parameters for sequence properties at the application level only. You cannot edit the data type parameters at the node type level.
- Sequence values are maintained at the application level for each property. That means that if you use the same sequence property in two different dimensions, the sequence is shared across those dimensions.
- Sequence values are derived when nodes are added or updated in a viewpoint (see Sequence Property Processing) and then stored and indexed. You can reference sequence values in both expressions and viewpoint queries.
- Although the sequence numbers that are generated are unique, by default there are no uniqueness constraints on the property values themselves. This means, for example, that you can manually populate two different nodes with the same sequence value in an application. If you want to enforce unique values, you can create an application-level constraint. See Working with Constraints.
- The next sequence number is maintained internally by adding the step value to the current sequence value. The system does not scan for existing values and then increment the current highest. For example, suppose you add nodes with derived values 1000, 1001, and 1002, and then manually add a node with a value of 1005. The next node that you add will get a value of 1003, not 1006. Furthermore, if you add two more nodes they will get values of 1004 and 1005, even though a node with 1005 already exists (unless you have created an application level constraint to ensure uniqueness).
- You can create sequence properties manually (see Creating Properties Manually) or during registration of a Universal application (see Creating a Custom Property).

Viewing, Editing, and Resetting Sequence Values

You can view the last sequence value by inspecting the property and viewing the sequence data type parameters. From the inspector, you can also manually enter a new sequence value, and you can clear the last sequence value to have the next value in the sequence begin with the original sequence start value.

Considerations

- Changes to sequence values apply to new requests, imports, and loads. No changes are made to In Flight requests.
- Manually entering a new sequence value will result in new sequence values starting at the number that you entered plus the sequence step value. See Editing the Last Sequence Value.

> **Note:**
>
> You can view and edit the step value by editing the property parameters. See Sequence Data Type Parameters.

- Resetting the sequence value will result in new sequence values using the initial starting value of the sequence property plus the step value. See Resetting the Last Sequence Value.

> **Note:**
>
> You can view and edit the initial sequence starting value and step value by editing the property parameters. See Sequence Data Type Parameters.

- Use caution when manually entering a last sequence value that is less than the last value or resetting the sequence value back to the initial starting value. This could result in sequence numbers that are duplicates of existing sequence numbers that have already been assigned. Consider setting up an application constraint to ensure that the sequence property has unique values. See Working with Constraints.

Viewing the Last Sequence Value

1. Inspect the sequence property that you want to view the last value for. See Inspecting Properties.
2. Select the application to view the last value for.
   The last value of the sequence that was used is displayed in the **Last Value** field. The next number in the sequence will be the last value plus the step value.

Editing the Last Sequence Value

1. From the property inspector, select the application that you want to edit the last sequence value for.
2. Click **Edit**.
3. In **Last Value**, click **Edit** ✎ .

   > **Note:**
   >
   > This icon is visible only when you are in Edit mode in the inspector.

4. In **Last Sequence Value**, enter a value. The value does not have to be greater than the current Last Value.
5. **Optional**: Review the displayed **Next Sequence Value** (the specified value plus the step value) to confirm that it meets your expectations.
6. Click **Save** to close the Edit Last Sequence Value window.

> **Caution:**
>
> After you click **Save**, the new value is stored, even if you click Cancel on the following screen.

7. Click **Save**, and then click **Yes** on the confirmation dialog box.

Resetting the Last Sequence Value

1. From the property inspector, select the application that you want to reset the last sequence value for.
2. Click **Edit**.
3. In **Last Value**, click **Edit** ✎.

> **Note:**
>
> This icon is visible only when you are in Edit mode in the inspector.

4. Next to **Last Sequence Value**, click **Clear**.
5. **Optional**: Review the displayed **Next Sequence Value** (the initial sequence starting value plus the step value) to confirm that it meets your expectations.
6. Click **Save** to close the Edit Last Sequence Value window.

> **Caution:**
>
> After you click **Save**, the new value is stored, even if you click **Cancel** on the following screen.

7. Click **Save**, and then click **Yes** on the confirmation dialog box.

Sequence Property Processing

This topic discusses how sequence properties are processed when nodes are added or updated in a viewpoint.

Interactive Requests

When you add a sequence property to a node type, any time a node with that node type is added, inserted, moved, edited, or reordered, if that property does not already have a defined value then it is populated with the next sequence value. Sequence properties are not populated if a node is removed or deleted.

The process of retrieving and populating the next sequence value in a property is an internal system property update action and therefore is not subject to security and validations. In other words, when a user adds or updates a node the system populates all sequence data types for that node that do not already have defined values even if:

- The user does not have write access on that property
- The property is hidden in the viewpoint

- The property is not defined as editable
- The property has Lock on Commit enabled

However, if a user clears or edits the sequence property manually the usual security and validations are applied.

> **Note:**
>
> For copy and model after operations the sequence values are not copied from the source nodes. The values for the new nodes are generated from the next value in the sequence.

Request File Load

For requests from a file load (that is, uploaded request files in an interactive request, compare requests, copy or model after requests, subscriptions, consolidation requests, and merge imports), nodes that do not already have defined values for a sequence property are updated as follows:

- If the request file contains a value for the sequence property, the value is updated as normal. This is a user action and security and validations apply.
  - The `<blank>` keyword updates the property with a defined blank value.
  - The `<clear>` keyword clears the existing value (including blank values) and gets the next sequence value.

  > **Tip:**
  >
  > If you add a sequence property to an existing node type, an easy way to generate sequence numbers for that property on existing nodes without making any other changes is to update that property with the `<clear>` keyword using a request file or viewpoint load.

- If the request file does not contain a value for the sequence property, then the property is updated with the next value in the sequence. This is an internal system action and security and validations do not apply.

  > **Note:**
  >
  > Sequence properties for nodes in a request file are updated even if that property isn't included in the file. For example, if your file contains updates for the Description property only, any nodes in that file that do not have values defined already for a sequence property will also have that property updated with the next sequence value, even if the user doesn't have write access to that property or the property is hidden in the viewpoint.

Imports and Viewpoint Loads

Merge Mode

Imports and viewpoint loads in merge mode are processed the same as Request File Load, above. All sequence properties (including unbound properties) that do not already have a defined value are updated with the next value in their sequence during the import or viewpoint load.

You can use the `<clear>` keyword to generate sequence values in imports and viewpoint loads in Merge mode.

Reset and Replace Modes

For imports in Reset or Replace mode, unbound sequence properties are not modified. Only bound sequence properties in Universal applications can be updated with sequence values. The values are updated following the same processing as Request File Load, above.

For viewpoint loads in Replace mode, only the sequence properties that are included in the viewpoint load file are updated with a value. If your viewpoint load file does not contain a column for a sequence property, that property is not updated during the viewpoint load.

Using the <clear> Keyword in Reset and Replace Modes

For viewpoint loads in Replace mode, you can use the `<clear>` keyword to generate sequence values for sequence properties.

For imports in Reset or Replace modes, you can use a clear keyword to generate sequence values if you have set one up in your import options. See Managing Import Options.

Inspecting Properties

You can use the **Property Inspector** dialog to examine property definitions, edit the name and description base parameters of certain properties, and make changes to application-specific default and allowed values.

When you inspect properties, you can view the base parameters set up during application registration and the application-specific properties. You can export the allowed values list for a property without being in edit mode for the inspector.

To inspect a property, from **Properties** click the link for the property that you want to inspect.

> **Note:**
>
> An editable property can be configured as read-only in a given viewpoint, see Configuring How a Viewpoint Displays Properties.

For more information, see Working with Properties.

Editing Property Parameters

Base parameters are set up during initial application registration. You can override these base parameters at the application level or node type level, and then revert them back to their application or base values as needed.

Base Parameters

Base parameters are the initial configuration of the property from the application registration process. Base parameters consist of these sections:

- Summary: Includes the property namespace, name, description, level, and data type.
- Default Parameters: Includes default type and default value.

- Common Parameters: Includes common parameters such as default value, whether the property is configured for inheritance, and whether the property is editable.
- Data Type Parameters: Includes parameters that are specific to the property data type. For example, a string property may have a minimum and maximum length for the string and a list of invalid characters.
- Allowed Values: Includes the allowed values for the property.

Base parameters for a property are established when you initially configure an application that uses that property. If you modify the registration for an application, the base parameters do not get updated.

From the property inspector, you can override all parameters except the Summary parameters for a property at the application or node type level.

Editing the Name and Description Base Parameters

Service administrators can edit the Name and Description base parameters of properties that are not in the `Core` or `CoreStats` namespace. Changes to the name or description base parameters will affect all applications that use the property.

To change the Name or Description base parameter:

1. Click **Edit** and then enter a new name or description.
2. Click **Save**.
   A confirmation message displays the applications that will be affected by the name change. Click **Yes** to continue or **No** to cancel.

> **Tip:**
>
> If you want to have a different label or description for a property in a specific viewpoint only, you can override the label or description for a property just for that viewpoint. See Configuring How a Viewpoint Displays Properties.

Adding a Property to a Node Type

You can add a property to a node type from the property inspector. After you add the property to the node type, you can define application and node type overrides for the property.

> **Note:**
>
> Adding a property to a node type in the property inspector is similar to adding it in the node type inspector (see Adding, Removing, and Configuring a Node Type's Properties) in that because the property was not added to a dimension through the application registration process, the property will not be included in imports and exports. See Working with Properties.

To add a property to a node type:

1. From **Properties** click the link for the property you want to inspect.
2. In Applications and Node Types, click **Add**.

> **Note:**
>
> You cannot be in Edit mode when adding a property to a node type.

3. From the drop down list of applications and node types, select the node type to add the property to, and then click **Save**.

> **Note:**
>
> The drop down list displays only the node types that do not already use the property, and for which you have *Owner* or *Metadata Manager* permission on the dimension that contains the node type.

Overriding Parameters at the Application or Node Type Level

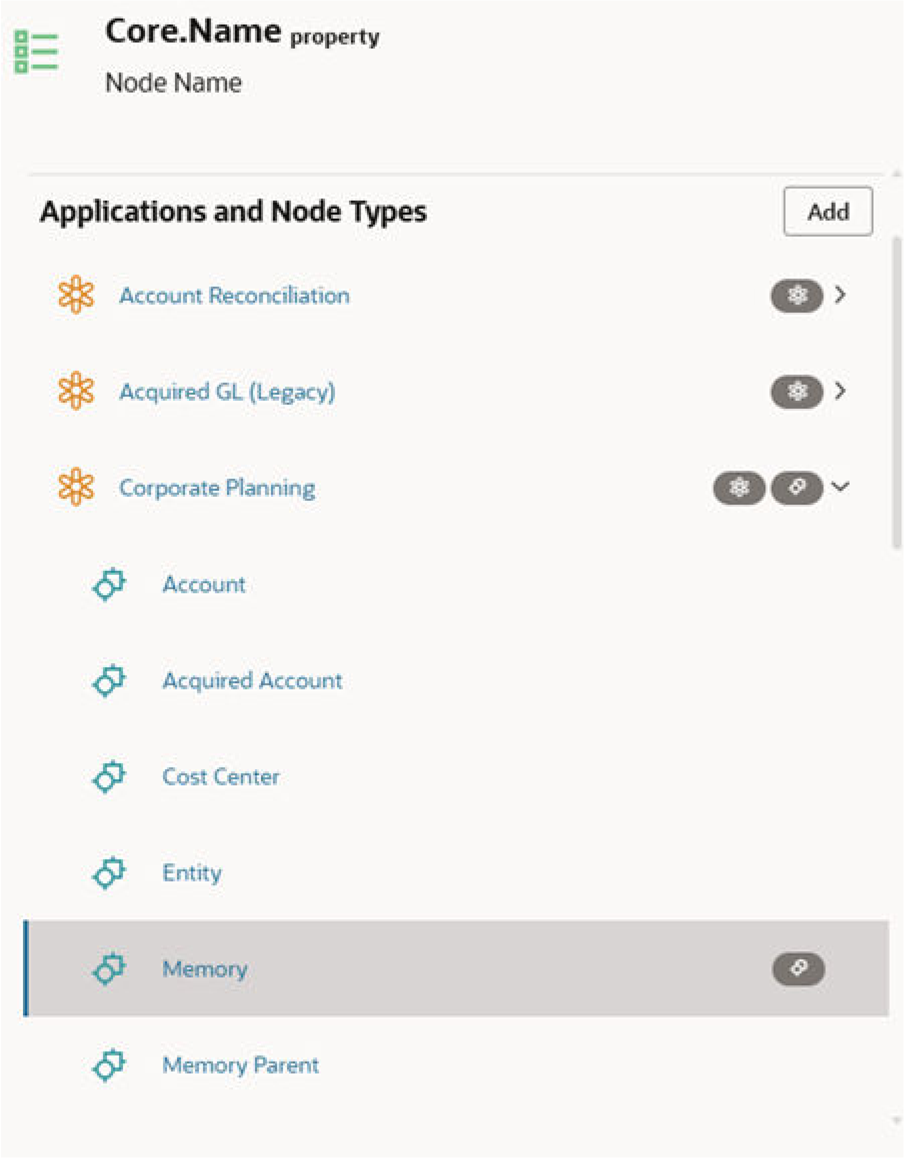
You can change the value of a parameter at the application or node type level by adding an override to that parameter. Parameters overridden at the node type level will supersede parameters at the application level. For example, if the **Editable** parameter for a property is selected at the application level but not selected at the node type level, users will not be able to edit that property for that node type.

Considerations

- You must have *Owner* or *Metadata Manager* permission on the application to override a parameter.
- When updates are saved in the property inspector, the updated values for application overrides are available when modifying an application's registration; however, the updated values for node type overrides are not.
- For system-defined properties, you can override data type parameters (case, invalid characters, minimum and maximum length) for the Name and Description properties of a node only.

To override a parameter for an application or node type:

1. From **Properties** click the link for the property you want to inspect.
2. Select an application (indicated by a ✿ icon) or a node type (indicated by a a ⚙ icon) from the list. Click › next to an application name to display the node types for that application.



3. Click **Edit**.
4. Edit the Default, Common, and Data Type parameters as needed. See Understanding Property Data Type Parameters.
5. **Optional**: For string parameters, edit the allowed values as needed. See Defining Allowed or Invalid Values for a Property.
6. Click **Save**, and then click **Yes** on the message confirming that the application values will be changed. .

When you override a parameter at the application or node type level:

- The Applications and Node Types list displays an application override ✻ icon to indicate that a parameter has been overridden at the application level and the node type override ⚙ icon to indicate that a parameter has been overridden at the node type level.

- In the Parameters pane, all of the parameters in the same group are updated with a ✾ icon at the application level or a ⚙ icon at the node type level to indicate that their values are different from the base parameter values.

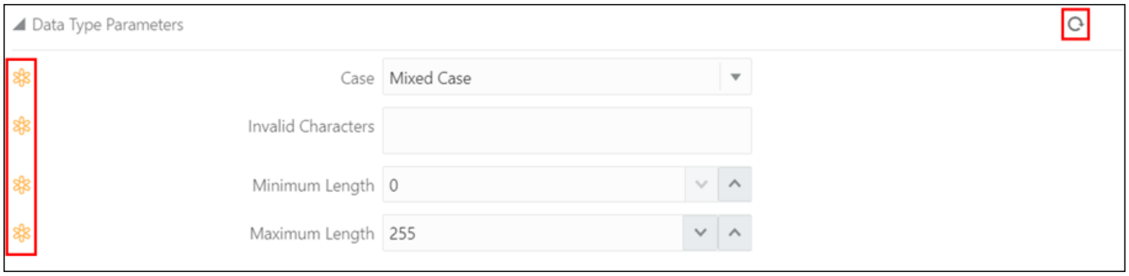
For example, if you update the **Minimum Length** parameter at the application level, the ✾ icon is displayed for **Case**, **Invalid Characters**, **Minimum Length**, and **Maximum Length**.

Disabling a Parameter at the Application or Node Type Level

You can specify that a parameter does not apply at the application or node type level by clearing the value of the entered text in that field and then clicking **Save**. The value for that parameter defaults to the value for the data type of that parameter. For example, if you clear the entered value for the maximum length of an property parameter with the integer data type, users can enter any number within the bounds of an integer type database column, regardless of any maximum length parameters at the application or base levels.

Reverting Overridden Parameters Back to their Application or Base Values

After you override a property parameter at the application or node type levels, the properties that you have overridden are updated with ✾ or ⚙ indicators to show that the current parameters are different from the base parameters. You can revert the properties that you have overridden at the application level back to their base values, and properties that you have overridden at the node type level back to the application or base values. When you revert properties back to their application or base values, you revert at the parameter type level. For example, when you revert the Data Type Parameters, the property settings for **Case**, **Invalid Characters**, **Minimum Length**, and **Maximum Length** are all reverted back to their application or base values.



Data Type Parameters

Case: Mixed Case

Invalid Characters:

Minimum Length: 0

Maximum Length: 255

To revert overridden parameters back to their application or base values:

1. In the property inspector, perform an action:
   - Select an application to revert parameters that were overridden at the application level back to their base values.
   - Select a node type to revert parameters that were overridden at the node type level back to their application or base values.
2. Click **Edit**.
3. On the parameter type that you want to revert, click ↻ to revert all of the properties for that type back to their application or base values.
4. Click **Save**.

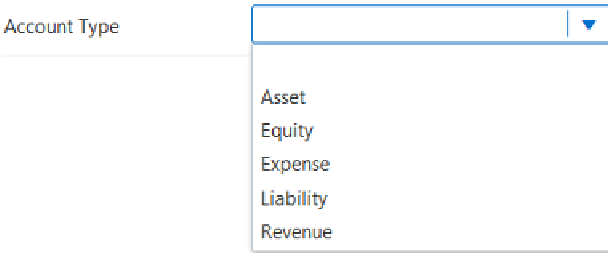
5. Click **Yes** to confirm that the overridden value will be reverted back to their application or base values.
6. When you have finished reverting all of the parameter types that you want to reset back to their application or base values, click **Save**.
7. Click **Yes** on the message confirming that the application values will be changed.

> **Note:**
>
> You cannot revert application overrides made to data type parameters for node data type properties (for example, Assigned Node Set or Bottom Level Only). These parameters are application-specific, and there are no values defined at the base level to revert back to.
>
> For the full list of data type parameters for node data type properties, see *Settings Specific to the Node Data Type* in Understanding Property Data Type Parameters.

Defining Allowed or Invalid Values for a Property

For properties with the string or list data type, you can specify that the property value is restricted to a set of values (Allowed Values), or cannot be equal to one of a set of values (Invalid Values).

For example, you can configure an Account Type property to allow only values of Asset, Equity, Expense, Liability, or Revenue, as in the following example:



You define the allowed or invalid values for a property from the property parameters page in the property inspector. See Editing Property Parameters.

Defining Allowed Values for a Property

You can define the allowed values for a property with the string or list data type in one of two ways:

- You can define a fixed set of allowed values in a table in the property parameters page.
- You can specify a node set and filter options to dynamically determine the allowed values for a property when selecting property values for nodes in a request or validating property values.

To define allowed values in a table:

1. Inspect the property that you want to define the allowed values for
2. Select the application or node type that you want to define allowed values for the property for, and then click **Edit**.

3. In **Allowed Values List**, click **Table**.
4. **(Optional)** To include blank values in the list of allowed values, select **Include Blank Entry**. See Understanding Null and Empty Values in Properties.
5. In the table header, click **Add new row** + and then enter the value and display text for the allowed value in the new row.
6. Continue adding new rows with the values and display text for all of the required allowed values for the property.
7. **Optional**: Click **Sort by** ⇅ to perform a one time sort of the Allowed Values list alphanumerically by either **Value** or **Label**.
8. To reorder an item in the list, in the **Action** column click **Action** ⋮ and then use the Up and Down commands to select the position for the item.
9. To remove an entry from the list, in the **Action** column click **Action** ⋮ in a table row and then select **Remove**.
10. To clear all items from the list, click **Action** ⋮ in the table header and then click **Clear**.
11. To import or export a list of allowed or invalid values, click **Action** ⋮ in the table header and then click the appropriate option. You can import and export values to an Excel file.

To dynamically determine allowed values using a node set:

1. Inspect the property that you want to define the allowed values for
2. Select the application or node type that you want to define allowed values for the property for, and then click **Edit**.
3. In **Allowed Values List**, click **Dynamic**
4. **(Optional)** To include blank values in the set of allowed values, select **Include Blank Entry**. See Understanding Null and Empty Values in Properties.
5. Next, define the dynamic allowed values by performing the following actions:
   a. In **Assigned Node Set**, specify the node set in the current application or a Users application with the allowed values for this property.
   b. In **Allowed Node Types**, specify one or more node types with the allowed values for this property. Click the **X** next to a node type to remove it from the list.
   c. In **Value Property**, select the property to use for **Value** of the allowed value. Only properties that are in all of the node types that you specified are displayed. The property that you select must contain only unique values (it cannot contain duplicate values).
   d. In **Label Property**, select the property to use for **Label** of the allowed value. Only properties that are in all of the node types that you specified are displayed. The property that you select must contain only unique values (it cannot contain duplicate values).

> **Note:**
>
> The **Value Property** and **Label Property** fields can reference the same property.

6. **(Optional)** Select **Bottom Level Only** to specify that the allowed values should be determined from bottom nodes only.
7. **(Optional)** In **Allowed Values Filter**, click **Edit Expression** *fx* and enter an expression to filter the nodes to be used for the allowed values. The expression is applied to the node set that you specified. See Using Expressions to Define Custom Business Logic.
8. Click **Save**.

> **Tip:**
>
> After you save the dynamic allowed values definition, click **Preview Allowed Values List** to see the allowed values that are generated from your definition. From the preview window, you can download the values to a file so that you can review them offline.
> The preview will also display any issues with the allowed values list, such as duplicate or blank entries in the labels or values.

Defining Invalid Values for a Property

To specify invalid values for a string or list property:

1. Inspect the property that you want to define the invalid values for
2. Select the application or node type that you want to define invalid values for the property for, and then click **Edit**.
3. In the **Invalid Values** table header, click **Add new row** + and then enter the value invalid value in the new row. Click **X** next to an invalid value to remove it from the list.
4. Continue adding new rows with the values for all of the required invalid values for the property.
5. To clear all items from the list, click **Action** ⋮ in the table header and then click **Clear**.
6. To import or export a list of allowed or invalid values, click **Action** ⋮ in the table header and then click the appropriate option. You can import and export values to an Excel file.

Downloading Property Definitions

You can download property definitions for a single property or for all properties from the property work list or the property inspector. The downloaded information includes base parameters as well as any application and node type overrides for the properties.

This enables you to, for example, filter and visually compare property parameters at the base, application, and node type level in an offline file.

Downloading Definitions for a Single Property

You can download the definitions for a single property from the property work list or the property inspector.

To download definitions for a single property, perform an action:

- From the property work list, in the Actions column of the property that you want to download the definitions for, click **Actions** ⋮, and then select **Download**.
- From the property inspector, in the left column click **Download** ⤓.

> **Note:**
>
> You cannot download property definitions if you are in Edit mode in the property inspector. Click **Save** or **Cancel** to exit Edit mode in order to enable the download button.

The property definitions are downloaded to an Excel file.

Downloading Definitions for All Properties

From the property worklist, click **Download** ⤓ to download property definitions for all properties to an Excel file.

> **Caution:**
>
> This operation could take several minutes if you have a lot of properties in your environment.

Property Definition Download File Format

Whether you download the definitions for a single property or all properties, the downloaded Excel file has the following tabs:

- **Properties**: Contains the property definitions at the base, application, and node type levels (excluding the detail values for Allowed Values, Invalid Values and Default Expressions).
- **Allowed Values**: Contains the detailed allowed values at the base, application, and node type levels.
- **Invalid Values**: Contains the detailed invalid values at the base, application, and node type levels.
- **Default Expressions**: Contains the detailed default expressions at the base, application, and node type levels.

> **Note:**
>
> Expressions are downloaded in `json` format.

Understanding Null and Empty Values in Properties

For properties, there is a difference between null values and defined empty values. This distinction has important ramifications for importing and exporting data to a dimension.

Null and Defined Empty Value Definitions

- **Null value**: A property that meets the following conditions:
  - Has no defined value
  - Has no default value. That is, either the property has no default value specified, or the specified default value is an empty string, or it is a derived property and the derived value is an empty string. See Derived Properties.
  - For inheriting properties, has no inheriting value. See Inheriting Properties.
- **Defined empty value**: A property that has an empty string as its stored defined value.

While any property can have a null value, only these property data types support a defined empty value:

- List
- Memo
- String (without **Use Allowed Values List** enabled)
- String (with **Use Allowed Values List** enabled, if **Include Blank Entry** is also enabled)

To illustrate the difference, suppose you have a Cost Center property that inherits its value from a parent node. If the parent node does not contain a value for that property, the Cost Center property value is **null**. When you enter a value in the parent node, the Cost Center property value gets inherited from the parent node. Then, if you clear the value from the Cost Center property, its value does not revert back to null, but rather it is now a **defined empty value**.

For information on importing and exporting null and empty values, see Managing Import Options and Managing Export Options.

Alternate Name Property

The alternate name property is used to identify common nodes across different node types when locating and comparing nodes. It is also used to match nodes during file uploads when the node primary name is not provided.

To use the alternate name property, add the `Core.Alternate Name` property to the node types that will use alternate names. See Adding, Removing, and Configuring a Node Type's Properties.

After you add the alternate name property to the node types, you can use it in the following ways:

- When locating nodes, the alternate name is searched for if the primary node name is not found. See Locating Nodes.
- When comparing nodes, both the primary node names and the alternate names are evaluated for the comparison, with the primary names taking precedence. See Running Viewpoint Comparisons.

- When loading request items, if the node name is not present in the file, the alternate name is used to locate and update the node. See Request Load File Format.

Understanding How to Configure Properties for a Viewpoint

The properties available for a viewpoint's nodes are those available on the data chain's node types. You can configure viewpoints to display properties in various ways. For example, you can specify a given property is editable in one viewpoint and read-only in another viewpoint.

The following table describes how data objects can configure properties.

Table 22-3 Data Objects and Properties

Data Object	Property Configurations
Node type	Defines the following for all viewpoints that include the node type in its data chain: • The available properties. • Whether the properties are required. A required property means that all nodes must have a value for the property. This is distinct from a mandatory property, which is a property that the node type must include in order to conform to the application's binding rules.
Viewpoint	Defines the following for a specific viewpoint: • The properties displayed in the viewpoint. • The name displayed for a property. In other words, you can specify a label for a property. • Whether a property value can be edited or is read-only.

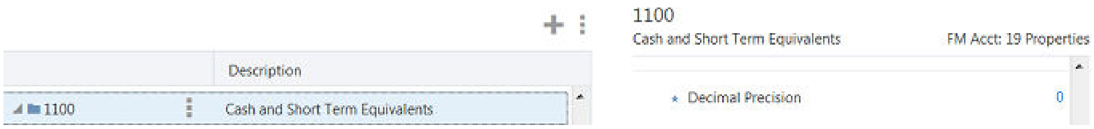
For more information, see Working with Properties.

Example 22-2 Configuring a Property

Suppose that you have a node type for a Universal application's Account dimension and that you created an optional property named `Custom.Number of Decimal Places`. In addition, suppose that you need to configure a viewpoint that handles the property in the following ways:

- The property is available and required for all account nodes.
- The property name displayed by the viewpoint is `Decimal Precision`.

The following example shows how the property should look in a viewpoint:

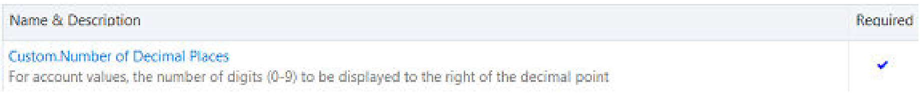


Description
1100 Cash and Short Term Equivalents

1100
Cash and Short Term Equivalents FM Acct: 19 Properties
* Decimal Precision 0

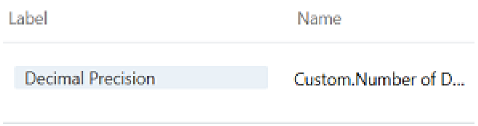
To accomplish this, you would configure the node type and viewpoint as described below:

1. In the node type's Properties tab, specify that the `Custom.Number of Decimal Places` property is required:



Name & Description	Required
Custom.Number of Decimal Places For account values, the number of digits (0-9) to be displayed to the right of the decimal point	✓

2. In the viewpoint's Properties tab, change the **Label** to `Decimal Precision`.

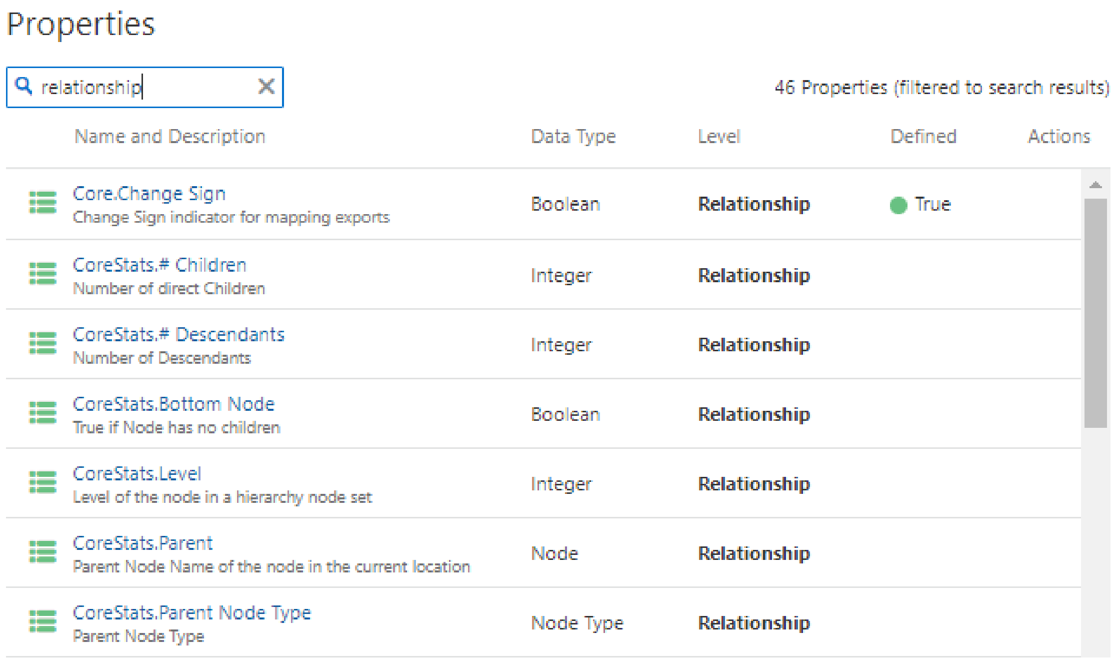


Label	Name
Decimal Precision	Custom.Number of D...

Searching for Properties

When you perform a search in the Properties list, the system searches for your text in the Name and Description, Data Type, and Level columns. Items that match your search text are displayed in the results window in bold-faced font.

For example, this search returned only Relationship level properties.



Properties

🔍 relationship ✕

46 Properties (filtered to search results)

Name and Description	Data Type	Level	Defined	Actions
Core.Change Sign Change Sign indicator for mapping exports	Boolean	**Relationship**	● True	
CoreStats.# Children Number of direct Children	Integer	**Relationship**		
CoreStats.# Descendants Number of Descendants	Integer	**Relationship**		
CoreStats.Bottom Node True if Node has no children	Boolean	**Relationship**		
CoreStats.Level Level of the node in a hierarchy node set	Integer	**Relationship**		
CoreStats.Parent Parent Node Name of the node in the current location	Node	**Relationship**		
CoreStats.Parent Node Type Parent Node Type	Node Type	**Relationship**		

To search in the Properties list:

1. From **Properties**, click 🔍.
2. Enter a text string to search for. The search is performed automatically after you stop typing.

   To close the search and return to the full list of properties, click ✕ .

Deleting Properties

Only custom properties may be deleted. A custom property may be deleted if it is not referenced by any other data chain object or by an application registration. Only a Service Administrator can delete a custom property.

> **Note:**
>
> Properties in reserved namespaces (Core, CoreStats, PLN, and so on) may not be deleted.

To delete a custom property:

1. From **Properties** find the custom property that you want to delete.
2. In the **Actions** column for the property, click ⋮, and then select **Delete**.
3. Click **Yes** to confirm the deletion.

Frequently Asked Questions about Properties

Why are my property values different for the same node between a list and a hierarchy viewpoint?

List and hierarchy viewpoints differ in their data chains:

- **List viewpoints** reference a list node set, which references one or more node types.
- **Hierarchy viewpoints** reference a hierarchy node set (which includes the top nodes for the viewpoints), which references a hierarchy set, which then references one or more node types (see Understanding Viewpoints).

For hierarchy viewpoints, the hierarchy set contains the Parent and Child relationships as well as the relationship level properties. Because list viewpoints do not reference a hierarchy set, it does not provide positional information or relationship level property information about a node.

Therefore, a list viewpoint and a hierarchy viewpoint will display different property values for the same node in these instances:

- The node level property is inherited.
- The node level property is derived based on positional information, such as Parents, Ancestors, or Siblings.

A list viewpoint and a hierarchy viewpoint will display the same property values for the same node in these instances:

- The node level property is defined.
- The node level property is not inherited.
- The node level property is not derived based on positional information, such as Parents, Ancestors, or Siblings.

> **Tip:**
>
> Using properties with Derived and Stored default values enable you to use an expression to calculate values that include positional logic and then store those values so that they can be referenced from a list viewpoint as defined values. See Derived and Stored Properties.

Consideration for Oracle Financials Cloud General Ledger and Oracle E-Business Suite General Ledger Exports

Because Oracle Financials Cloud General Ledger and E-Business Suite General Ledger applications export segment values from the list viewpoint and segment tree values from hierarchy viewpoints, it is important that you do not use any properties with a Derived default type with positional logic. Use Derived and Stored default type properties instead.

> **Note:**
>
> If you switch a property from Derived to Derived and Stored, you must use requests or loads to populate the values so that they become defined.

23

Understanding Validations and Constraints

Validations and constraints preserve data integrity and enforce business logic in Oracle Fusion Cloud Enterprise Data Management.

Videos

Your Goal	Watch This Video
Learn about validation best practices.	▶ Validation Best Practices for Trustworthy Enterprise Data

There are three types of validations that run in Cloud EDM:

- **System Validations**: General system checks. For example, a validation checks to see if you're trying to create a node type that already exists. For a list of system validations, see System Validations. System validations are enforced on all data, and you cannot disable them.
- **Predefined Validations**: Validations that are automatically created depending on the type of application that you create. Predefined validations are enforced on all bound data in a dimension, but you can disable them or set the severity level at the dimension level. See Predefined Validations.
  - Planning and FreeForm Validations
  - Oracle Financials Cloud General Ledger Validations
  - E-Business Suite General Ledger Validations
  - Financial Consolidation and Close Validations

> **Note:**
>
> To preserve data integrity, predefined validations run for data objects with these binding statuses. See Binding Status.
>
> - Bound viewpoints run all predefined validations.
> - Hierarchy viewpoints that are node set bound or hierarchy set bound run all predefined validations.
> - Hierarchy viewpoints that are node type bound run these validations:
>   - All node type-level (such as property validations) and application-level validations (such as name uniqueness) are run.
>   - Structure-based validations (such as parent-child validations) are run on the hierarchy set in the viewpoint. These validation results may be different from the bound hierarchy set.
> - List-type viewpoints that are node type bound run all predefined validations except structure-based validations (such as parent-child validations).

- **Custom Validations**: User-created validations to enforce business logic that is unique to a specific implementation. See Custom Validations.

Constraints check for specific conditions (such as name uniqueness) that can go across node types and dimensions in an application. For example, a constraint could enforce node name uniqueness at the application level, as opposed to a custom validation that runs on nodes for specific node types or hierarchy sets only. They are similar to predefined validations, but you can configure the application or dimension that they apply to and you can add them to Universal applications. See Working with Constraints.

Validations and constraints run when you:

- Import data from an external application
- Export data to an external application, if **Validate before Export** is set to Alert or Discontinue on Error. See Managing Export Options.
- Manually enter changes in a viewpoint
- Load a file with request changes to a viewpoint
- Validate a request item
- Validate a request
- Validate a viewpoint

System Validations

System Validations

System validations are always run for all application types. They cannot be disabled, and you cannot change their severity level.

The following tables list the system validations.

Table 23-1 Node Level Validations

Name	Description
Node Type Exists	Node type does not exist
Node Exists in Node Type	Node does not exist in node type
Node Exists in Request Items	Node does not exist in request items
Node Exists in Node Set	Node does not exist in node set
Node Name Length	Name exceeds the max length

Table 23-2 Hierarchy Level Validations

Name	Description
Parent Node Type Exists	Parent node type does not exist
Parent Exists in Node Type	Parent is not an existing node in node type
Parent Exists in Request Items	Parent does not exist in request items
Parent Exists in Node Set	Parent does not exist in node set
Child Node Type in Hierarchy Set	Node type is not a child node type for parent in hierarchy set

Table 23-2 (Cont.) Hierarchy Level Validations

Name	Description
Relationship Exists in Hierarchy Set	Parent/child relationship does not exist in hierarchy set
Shared Nodes in Hierarchy Set	Shared nodes are not allowed for the hierarchy set
Descendant of Self	Node is the same as the parent (or ancestor)
Sibling of Self	Node exists as a child of the parent

Table 23-3 Viewpoint Level Validations

Name	Description
Allowed Action for Node Type	Request action is not allowed for node type in the viewpoint
Viewpoint Status	Viewpoint does not allow edits
Displayed Property in Viewpoint	Property is not displayed for node type in viewpoint
Property Editable in Viewpoint	Property is not editable for node type in viewpoint

Table 23-4 Property Level Validations

Name	Description
Data Type for Property	Value does not match the data type for the property
Format for Property	Value does not match the format expected for the property
Max Length for Property	Value exceeds the maximum length for the property
Invalid Characters for Property	Value includes an invalid character for the property
Invalid First Characters for Property	Value includes an invalid first character for the property
Min and Max Values for Property	Value is not within the minimum and maximum range for the property
Allowed Value List for Property	Value is not in allowed list of values for the property
Required Property for Node Type	Value is not entered for property in node type
Node Exists in Property Node Set	Value is not a node in the node set configured for property
Leading or Trailing Spaces	Value includes leading or trailing spaces

> **Note:**
>
> Many property level validations such as minimum and maximum length, invalid characters and first characters, and invalid values can be modified by editing the property parameters. See Editing Property Parameters.

Predefined Validations

Predefined validations are created automatically when you register an application, and they enforce the requirements for the external system based on the application type or Oracle Fusion Cloud Enterprise Data Management requirements.

For example, when you register a Planning application, a predefined validation ensures that member names are unique across all nodes used in the application, and when you register a Financials Cloud General Ledger application, a predefined validation ensures that node end date is after its start date.

There are two types of predefined validations:

- **Application-specific predefined validations** apply only to specific types of applications. Most application-specific predefined validations are enforced on bound data in a dimension. They are enabled by default, but you can disable them or set the severity level at the dimension level (see Configuring Validation Enforcement and Severity).
- **Generic predefined validations** apply to all types of applications except Users applications. You can enable and disable them on the Validations tab of the dimension inspector. See Configuring Validation Enforcement and Severity.

Source Application Types for Validations

All predefined validations have a source application type which identifies the type of application the validation originated from. Applications can have predefined validations from multiple source types.

Planning validations represent the basic EPM Cloud Platform validations. Other EPM Cloud application types will therefore have validations with the Planning source type, as well as their own validations. For example, a Financial Consolidation and Close application has validations with Planning source types (such as Alias Uniqueness) as well as validations with Financial Consolidation and Close type (such as Default Movement Level).

Application-specific Predefined Validations

For details about the application-specific predefined validations that are created for each application type, see:

- Planning and FreeForm Validations
- Financial Consolidation and Close Validations
- Enterprise Profitability and Cost Management Validations
- Tax Reporting Validations
- Oracle Financials Cloud General Ledger Validations
- E-Business Suite General Ledger Validations

Generic Predefined Validations

When you register any application type other than Users applications, these validations are automatically created for that application:

Table 23-5 Generic Validations

Name	Description	Scope
Equivalent Nodes with Filter	All nodes in this viewpoint that match a specified filter expression (or all nodes, if there is no filter expression) must exist in all related viewpoints in at least one location. See Configuring Related Viewpoints. This validation is disabled by default.	Related Viewpoints
Ownership Approval Permission	An ownership approval property for a node refers to a user who does not have at least *Participant (Read)* access to the viewpoint. See Understanding Ownership Approvals. This validation is disabled by default.	Node and user node
Previously Merged Node	Nodes that are added in the viewpoint cannot have the same name as nodes that were previously merged as part of the matching process. See Working with Matching and Deduplication. This validation is enabled by default.	Node

Custom Validations

Custom validations enable you to add custom business logic to enforce data rules on your applications, dimensions, node types, and hierarchy sets.

Videos

Your Goal	Watch This Video
Learn about working with and creating custom validations.	▶ Working with and Creating Custom Validations

Considerations

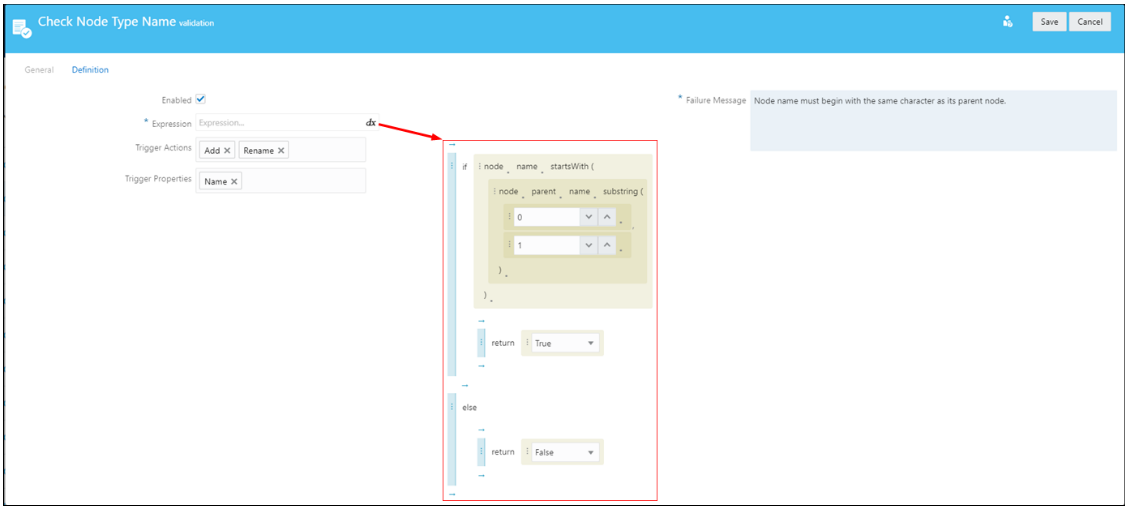
- You must have the *Owner* or *Metadata Manager* permission on at least the dimension to create, edit, or delete a custom validation.
- Node type validations are executed for all nodes in the viewpoint by node type.

- Hierarchy set validations are executed for all nodes in the viewpoint, as long as the viewpoint is a hierarchy.
- Validations are run on the state of nodes after a request is run (that is, what the node will look like after the request is committed). You cannot run a validation on the state of a node before a request is run.
- Validations are enforced when a request is submitted, approved, committed, or a validation is performed. The following table describes which custom validations are run when validations are performed in different contexts:

Validation Context	Validations Run
Validate a request	Custom validations are run for request items that meet either the trigger actions or trigger properties
Validate a viewpoint with an open request	Custom validations are run for all nodes in the viewpoint, including the request items
Validate viewpoint outside of a request	Custom validations are run for request items that meet the trigger properties, but not the trigger actions

Custom Validation Example

In this example, we add a validation to the Account node type in the Corporate Planning application that says that all node names must begin with the same character as that node's parent. The following screenshot shows the custom validation that we created.



Check Node Type Name validation

General   Definition

Enabled ☑

* Expression Expression...

```
if node . name . startsWith (
    node . parent . name . substring (
        0 ,
        1 .
    ) .
) .
    return True
else
    return False
```

Trigger Actions: Add ✕ Rename ✕

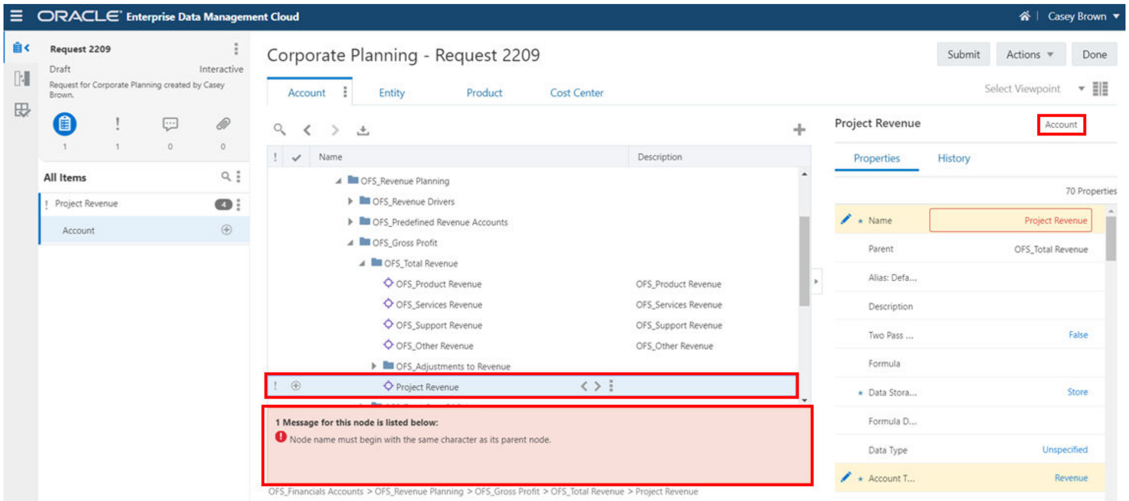
Trigger Properties: Name ✕

* Failure Message: Node name must begin with the same character as its parent node.

The validation has this definition:

- The Enabled flag is set to True.
- The expression returns a value of True if the node name starts with the same character as the first character in the node parent's name, and a value of False if it doesn't.
- The trigger actions for the validation are Add and Rename, and the trigger property is Name.
- A failure message indicates that the validation fails if the account node doesn't start with the same character as its parent node.

To test the validation, let's try to submit a request that adds node for the Account node type whose name starts with a character that is different from the first character of its parent. In this case, under the parent node `OFS_Total Revenue`, we will attempt to add a node named `Project Revenue`.



1 Message for this node is listed below:

Node name must begin with the same character as its parent node.

OFS_Financials Accounts > OFS_Revenue Planning > OFS_Gross Profit > OFS_Total Revenue > Project Revenue

Name	Project Revenue
Parent	OFS_Total Revenue
Alias: Defa...	
Description	
Two Pass ...	False
Formula	
Data Stora...	Store
Formula D...	
Data Type	Unspecified
Account T...	Revenue

When we submit the request, we get a validation error that the node name must begin with the same character as its parent node.

Creating Custom Validations

1. Inspect the data chain object that you want to add a validation for. See:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Node Type
   - Inspecting a Hierarchy Set
2. Perform an action:
   - For applications, node types, hierarchy sets, and dimensions in Universal applications: On the Validations tab, click **Create**.
   - For dimensions in applications other than Universal: On the Validations tab, click the Custom sub tab, and then click **Create**
3. Enter a name and, optionally, a description for the validation, and then click **Create**. The validation is displayed in the validation inspector.

Editing Custom Validations

1. Inspect the data chain object that you want to edit a custom validation for.
2. Perform an action:
   - For applications, node types, hierarchy sets, and dimensions in Universal applications: On the Validations tab, click **Edit**.
   - For dimensions in applications other than Universal: On the Validations tab, click the Custom sub tab, and then click **Edit**

3. On the **General** tab of the validation inspector, click **Edit** to edit the validation name or description.
4. On the **Definition** tab, define the validation by entering the following information:
    - **Enabled flag**: Specify whether or not a custom validation is enforced. Validations are disabled by default, and you can only enable them when these conditions are met:
        - A valid expression is defined
        - At least one trigger action or property is configured
        - A failure message is defined

        > **Note:**
        >
        > You can also enable or disable validations from the Validation tab of the node type or hierarchy set inspector. See Enabling, Disabling, or Deleting a Custom Validation.

    - **Expression**: Click **Define Expression** *fx* to open expression builder and then define the business logic for the validation. See Using Expressions to Define Custom Business Logic.
      The expression for a validation must return a Boolean value. If the expression returns a value of True, the validation is passed. If the expression returns a value of False, the validation fails and the failure message is displayed.
    - **Trigger Action**: Select one or more request actions that will cause the validation to run, or select the **All** check box to select all actions.
    - **Trigger Properties**: Select one or more properties that will cause the validation to run if they are updated, or select the **All** check box to select all properties.
        - For node type validations, all properties that are assigned to that node type are available to be selected.
        - For hierarchy set validations, all properties that are assigned to any node type that is referenced by the hierarchy set are available to be selected.

      Derived properties are validated whenever a node is updated, regardless of whether or not they are selected as Trigger Properties. See Derived Properties.

      > **Note:**
      >
      > Trigger Actions and Trigger Properties are evaluated using an OR condition. That is, the validation is run if either a trigger action is performed or a trigger property is updated.

> **Tip:**
>
> Validations are run for every trigger action and trigger property in a request. That means that if you add several trigger properties and all of them fail, you will receive several failure messages. You should add the minimum number of trigger properties or actions that still enforce your business logic on a request.

- **Request Validation Scope**: Select the context in which the validation is run when validating a request.
  - **Node** (default): The validation is evaluated in the context of the node in the request action. The validation is triggered when changes are made to the node itself.
  - **Parent**: The validation is evaluated in the context of the parent of the node in the request action. When changes are made to a node in a hierarchy, the parent of the node in the request action is evaluated for any validations with a scope of **Parent**.
  - **Previous Parent**: The validation is evaluated in the context of the previous parent node (the parent that the node was moved from) when the parent is being changed in a request.

    > **Note:**
    >
    > When you set the scope to Previous Parent, the Trigger Action is set to **Move** and the Trigger Properties are set to **None**. These settings cannot be changed.

  - **Both Parents**: The validation is evaluated in the context of the both the previous parent node (the parent that the mode was moved from) and the new parent (the parent that the node was moved to) when the parent is being changed in a request.

    > **Note:**
    >
    > When you set the scope to Both Parents, the **Move** action is added to the Trigger Actions. You can specify additional trigger actions and trigger properties. However, the previous parent will be evaluated only for move actions.

  Note the following about request validation scope:
  - The scope is used when validating request items only. When validating viewpoints or exports:
    - If a validation has a scope of **Parent** or **Both Parents**, the scope setting is ignored and the validation is run using the **Node** scope (that is, the validation is run on the node where the validation is defined).
    - If the validation has a scope of **Previous Parent**, it is not run during the validation operation.
  - Validations with a scope of Parent, Previous Parent, or Both Parents are not run when validating requests for a list viewpoint.

  - When validating requests, trigger actions and properties are evaluated based on the child action performed in the request. For example, a validation with a scope of **Parent** and a trigger property of `Core.Description` is evaluated any time the description property is updated on a child node of that parent.
- **Request State**: Select the state in which custom validations are run in the context of a request:
  - **Committed** (Default): The validation is evaluated against data *after* the request items and actions have been applied to the viewpoint. Nodes that are removed or deleted in the request are excluded when evaluating data conditions in the Committed state.
  - **Visualized**: The validation is evaluated against data *before* the request items and actions have been applied to the viewpoint. Nodes that are removed or deleted in the request are included when evaluating data conditions in the Visualized state.

> **Note:**
>
> Validations using the Visualized state are only run when validating request items and do not run when validating a viewpoint.

- **Severity**: Select the severity for the validation (**Error**, **Warning**, **Ignore**) at the Request Submit, Approve, and Commit stages, as well when validating a viewpoint and exporting a dimension. See Configuring Validation Enforcement and Severity.
- **Failure Message**: Enter the message to display to users if the validation fails.

> **Tip:**
>
> When configuring validation failure messages, provide the context of the validation (node or parent) in order to help you identify where the issue was found.

5. Click **Save**.

Enabling, Disabling, or Deleting a Custom Validation

1. Inspect the data chain object that you want to enable, disable, or delete a custom validation for.
2. Perform an action:
   - For applications, node types, hierarchy sets, and dimensions in Universal applications: On the Validations tab, click **Edit**.
   - For dimensions in applications other than Universal: On the Validations tab, click the Custom sub tab, and then click **Edit**
3. In the **Actions** column next to the validation that you want to take action on, click ⋮, and then select an option:
   - **Enable** (available only if the validation is disabled)
   - **Disable** (available only if the validation is enabled)
   - **Delete**

4. Click **Yes** on the confirmation message, and then click **Save**.

Configuring Validation Enforcement and Severity

For custom and predefined validations, by default a validation failure displays an error and stops the current operation. You can modify the severity to display a warning or ignore the validation instead. You can also disable them.

> **Note:**
>
> You can configure the enforcement and severity for custom and predefined validations only. System validations are generally enforced and their severity is usually Error, although this may vary by property type.

You can set the severity levels for each of the following operations:

- **Request Submit**: The validation is evaluated when the request is in the Submit stage (regardless of the status of Draft, Pushed Back, or Recalled)
- **Request Approve**: The validation is evaluated when the request is in the Approve stage
- **Request Commit**: The validation is evaluated when the request is in the Commit stage

  > **Tip:**
  >
  > Configuring different severity levels for each request stage enables you to, for example, configure a validation so that a failure will display a warning during the Submit stage (which still allows the request to be submitted), but to display an error during the Approve stage so that the request cannot be approved until the failure is resolved.

- **Viewpoint Validate and Export**: The validation is evaluated when you validate a viewpoint or export a dimension.

  > **Note:**
  >
  > When exporting a dimension, only validations with a severity of Error are run, and the outcome depends on the dimension's **Validate Before Export** setting. See Validating Viewpoints before Exporting Dimensions.

The outcome of each scenario is described in the table below.

Table 23-6 Validation Severity Per Operation

Severity	All Request Stages	Validating a Viewpoint	Exporting a Dimension
Error	Displays an error and prevents the request from transitioning to the next stage.	The validation is performed and the Error severity is indicated in the viewpoint validate results.	The validation is run before exporting a dimension. The **Validate Before Export** setting determines the outcome of validation failures. See Validating Viewpoints before Exporting Dimensions.
Warning	Displays an error, but permits the request to transition to the next stage (after user confirmation).	The validation is performed and the Warning severity is indicated in the viewpoint validate results.	The validation is not run before exporting a dimension.
Ignore	The validation is not processed.	The validation is not performed during viewpoint validation.	The validation is not run before exporting a dimension.

Managing Enforcement and Severity of Predefined Validations

1. Inspect the dimension that you want to manage a validation for. See Inspecting Dimensions.
2. On the **Validations** tab, select the **Predefined** sub tab to view the predefined validations that are run on that dimension.
3. Click **Edit**.
4. To enable or disable a validation, select or clear the **Enabled** check box.
5. To modify the validation severity, click **Edit Validation Settings** ✎ and then select the severity for the validation (**Error**, **Warning**, **Ignore**) at the Request Submit, Approve, Commit, and Viewpoint Validate and Export stages.
6. Click **Save**.

> **Note:**
>
> For Oracle Financials Cloud General Ledger and Oracle E-Business Suite General Ledger applications, if the validation for upper case node names should not be enforced, then in addition to disabling the `Uppercase Node Name` validation at the application level you must also update the Case setting for `Core.Name` property at the override level. See Editing Property Parameters.

Managing Enforcement and Severity of Custom Validations

- To enable or disable a custom validation, see Enabling, Disabling, or Deleting a Custom Validation.
- To manage the severity of a custom validation, see Editing Custom Validations.

Working with Constraints

Constraints enable you to enforce specific data rules (such as name uniqueness) across node types and dimensions in an application.

Constraints differ from custom validations in that they check for conditions across a set of nodes, whereas custom validations check for conditions at the node level. For example, a custom validation can check for a condition on a node or its parent, but constraints enable you to check that condition across a set of nodes at the dimension or application level.

Considerations

- You must have the *Owner* or *Metadata Manager* permission on an application in order to create any type of constraint in that application.
- Constraints can be added to enforce property value uniqueness for an application, a dimension, or across specific node types. When enabled, a constraint will check that a property that you specify has unique values across either all of the bound node types in a specified application or dimension, or across any node types (bound or unbound) that you specify.

  > **Note:**
  >
  > - Planning, Planning Modules, and Financial Consolidation and Close applications in Oracle Fusion Cloud Enterprise Data Management already have predefined validations that check for node name uniqueness across node types at the application level. You do not need to create a constraint to enforce this data rule for those application types.
  > - Oracle Financials Cloud General Ledger applications in Cloud EDM already have predefined validations that check for node name uniqueness across node types at the dimension level. You can add a constraint at the application level if you want to apply the node name uniqueness rule across segment value sets.

- You can create constraints for a property that meets these criteria:
  - The property must be a node level property. You cannot create a constraint on a relationship level property.
  - The property data type must be a string, node, numeric string, or a sequence data type.
  - The property value cannot be inherited, and it cannot have a Derived or Derived and Stored default value.

    > **Note:**
    >
    > If you have added a constraint for a property, you cannot change that property to have an inherited value or a Derived or Derived and Stored default value.

  - For the `Core.Name` property only, the default type can be None or Calculate and Store.

- You can add constraints for up to four unique properties per application. However, you can add more than one constraint on a single property. For example, you can have an application constraint on the `Alias:Default` property, and then create a separate node type constraint at the node for `Alias:Default` that checks against your unbound node types. This still counts as one property towards the application limit of four.

  > **Note:**
  >
  > If you have created an application constraint for a specific property, you do not need to create a separate dimension constraint for that same property in that application. Application constraints already check for property value uniqueness across all bound dimensions in that application.

Creating a Constraint

1. Inspect the application that you want to create a constraint for.
2. On the Constraints tab, click **Create**, and then perform an action:
   - To check uniqueness of a property for all nodes within all bound node types in an application:
     a. Select **Application Bound Uniqueness**.
     b. Enter a name and description.
     c. From **Properties**, select a property for the constraint. Only node level properties with the string, node, numeric string, or sequence data type whose default values are not inherited or derived are displayed. `Core.Name` can be selected regardless of its default type.
     d. Select whether to enable the constraint or not. By default, the constraint is enabled.
     e. **Optional**: Assign the severity level (**Error**, **Warning**, or **Ignore**) of the constraint for these operations:
        – Request Submit
        – Request Approve
        – Request Commit
        – Viewpoint Validate and Export

        See Configuring Validation Enforcement and Severity for details.
   - To check uniqueness of a property for all nodes within all bound node types in a dimension:
     a. Select **Dimension Bound Uniqueness**.
     b. From the dimension drop down menu, select a dimension. The drop down menu displays bound dimensions only.
     c. Enter a name and description.
     d. From **Properties**, select a property for the constraint. Only node level properties with the string, node, numeric string, or sequence data type whose default values are not inherited or derived are displayed. `Core.Name` can be selected regardless of its default type.

   e. Select whether to enable the constraint or not. By default, the constraint is enabled.

   f. **Optional**: Assign the severity level (**Error**, **Warning**, or **Ignore**) of the constraint for these operations:

      - Request Submit
      - Request Approve
      - Request Commit
      - Viewpoint Validate and Export

      See Configuring Validation Enforcement and Severity for details.

- To check uniqueness of a property for all nodes within selected node types in an application:

   a. Select **Node Type Uniqueness**.

   b. In **Node Types**, select a node type to add it to the constraint. You can select multiple node types for the constraint, and you can select bound and unbound node types. Click the **X** to remove a selected node type from the constraint.

   c. Enter a name and description.

   d. From **Properties**, select a property for the constraint. Only node level properties for the node types that you selected that have the string, node, numeric string, or sequence data type and whose default values are not inherited or derived are displayed. `Core.Name` can be selected regardless of its default type.

   e. Select whether to enable the constraint or not. By default, the constraint is enabled.

   f. **Optional**: Assign the severity level (**Error**, **Warning**, or **Ignore**) of the constraint for these operations:

      - Request Submit
      - Request Approve
      - Request Commit
      - Viewpoint Validate and Export

      See Configuring Validation Enforcement and Severity for details.

3. Click **Save**.

Editing a Constraint

1. From the Constraints tab of the application inspector, select the constraint that you want to edit and click **Edit**.

   > **Note:**
   >
   > You cannot change the type of a constraint (for example, from application to dimension type) or the dimension that a constraint applies to after it has been saved .

2. **Optional**: Edit the **Name** or **Description** of the constraint.

3. **Optional**: In Property, select the property for the constraint. Only node level properties that have the string, node, numeric string, or sequence data type and whose default values are

not inherited or derived are displayed. `Core.Name` can be selected regardless of its default type.

4. **Optional**: In **Enabled**, select to enable or disable the constraint.
5. **Optional**: Assign the severity level (**Error**, **Warning**, or **Ignore**) of the constraint for these operations:
   - Request Submit
   - Request Approve
   - Request Commit
   - Viewpoint Validate and Export

   See Configuring Validation Enforcement and Severity for details.

Deleting a Constraint

To delete a constraint, click **X** on the constraint to be deleted. You must not be in edit mode to delete a constraint.

24
Working with Lookup Sets

Lookup sets enable you to transform a set of values for properties into a different set of values. For example, in the Account Type property, the value "A" can be translated into the value "Asset". Lookup sets are used in expressions for deriving properties and converting node types.

See Using Expressions to Define Custom Business Logic.

Considerations

- You must have the *Owner* or *Metadata Manager* permission on at least one application to create or delete a lookup set. After a lookup set is created, all users are able to view and inspect it.
- You must have the *Owner* or *Metadata Manager* permission on an application to add or update a lookup for that application.
- Lookup sets contain source values, called keys, and target values. Key values must be unique, but you can associate more than one key to each target value.
- Lookups are performed from key to value only. You cannot lookup from value to key.
- When defining a lookup set, add key and value pairs only when the key and associated value are different. If the key and value are the same, you do not need to be add them to the lookup set. When you pass a key that is not in the lookup set, the key itself is returned.
- Lookup keys and values support the string data type only.

Creating Lookup Sets

To create a lookup set:

1. From the Home page, click the Lookup Sets tile.
2. Click **Create**.
3. Enter a name and, optionally, a description, and then click **Create**.
   The lookup set is created.

To add keys and values to a lookup set, click the name of the lookup set to open it in the inspector. See Inspecting Lookup Sets.

Deleting Lookup Sets

Service administrators can delete lookup sets that are not being used by any applications (that is, there are no lookup values for an application defined in the lookup set).

To delete a lookup set:

1. From the Home page, click the Lookup Sets tile.
2. In the Actions column, click the **Action Menu** ⋮, and then click **Delete**.

> **Note:**
>
> The **Delete** menu item is not available if the lookup set is being used by an application.

3. Click **Yes** to confirm that you want to delete the lookup set, or click **No** to cancel.

After you delete a lookup set, you can reuse its name to create a new lookup set or to rename an existing lookup set.

Inspecting Lookup Sets

You can inspect a lookup set to add, edit, or delete keys and values to it.

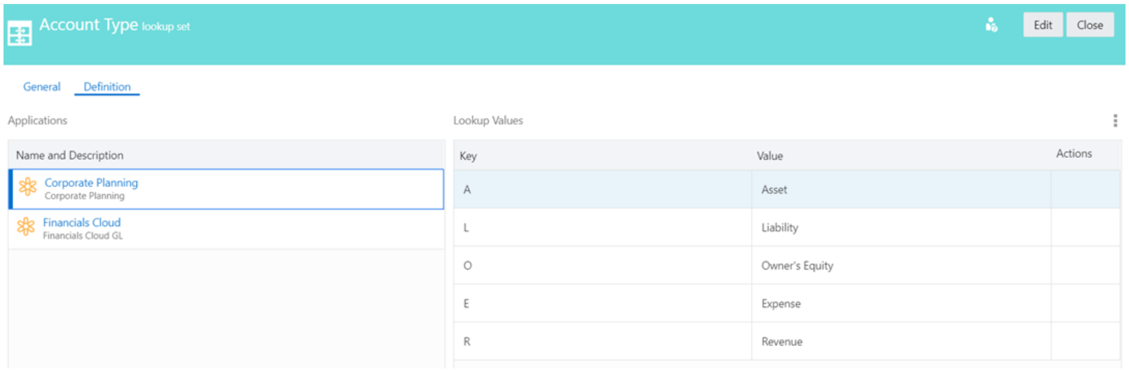
1. From **Lookup Sets**, scroll to the lookup set that you want to inspect, click ⋮, and select **Inspect**.
2. Select from the options available on these tabs:
   - **General:** Displays the name, description, and the Created By and Modified By users.
   - **Definition:** Displays the application lookups and the keys and lookup values for each application.

To add an application lookup:

1. From the Definition tab, click **Edit**.
2. In Applications, click **Add**.
3. Select the application that you want to add to the lookup. Only the applications for which you have *Owner* or *Metadata Manager* permission are displayed.
4. In Lookup Values, click ✚.
5. Enter a key and value pair for the lookup.

> **Note:**
>
> Each key must be unique, but you can add more than one value to a key.

6. Repeat steps **4** and **5** until you have added all of the key and value pairs for the application lookup.
7. To delete a key and value pair, in the Actions column, click ⋮ and select **Delete**.
8. When you have finished entering key and value pairs for the application lookup, click **Save**



Name and Description
Corporate Planning Corporate Planning
Financials Cloud Financials Cloud GL

Key	Value	Actions
A	Asset	
L	Liability	
O	Owner's Equity	
E	Expense	
R	Revenue	

Importing and Exporting Lookup Set Values

The key and value pairs for lookup sets can be imported and exported to an Excel file. The values in the first column are the keys, and the values in the second column are the values. All other columns are ignored.

When you import key and value pairs for an application lookup, the import overrides the existing values in the application lookup.

To import an application lookup:

1. On the Definition tab, click **Edit**.
2. Next to Lookup Values, click ⋮, and select **Import**.
   An import dialog box is displayed.
3. Select or drag an Excel file to the dialog box.
   The import is run automatically and the dialog box is closed.
4. Click **Save**.

To export an application lookup, on the Definition tab, click ⋮, and select **Export**. The export file is created and downloaded automatically to the client local file system. The file is named for the property and application being configured, for example, `Account+Type-Corporate+Planning.xlsx`.

25

Using Expressions to Define Custom Business Logic

Expressions enable you to define business logic for managing your data. You can use expressions to transform properties when you bring data across applications, derive the value for a property based on the value of another property, or create custom validations for node types or hierarchy sets. The expression builder enables you to define complex business logic in a structured environment that requires minimal coding or scripting knowledge. The expressions that you build are automatically validated as you build them.

This topic discusses how to use the expression builder to define business logic. For more information about the contexts in which that business logic is applied, see these topics:

- Working with Node Type Converters
- Derived Properties
- Custom Validations
- Adding Filters and Conditions to Subscriptions
- Creating and Enabling Approval Policies
- Creating and Enabling Notification Policies

Videos

Your Goal	Watch This Video
Get familiar with Custom Business Logic	Incorporating custom business logic using expressions
Derive properties using expressions	Deriving Properties with Expressions
Transform properties using expressions	Transforming Properties with Expressions

Tutorials

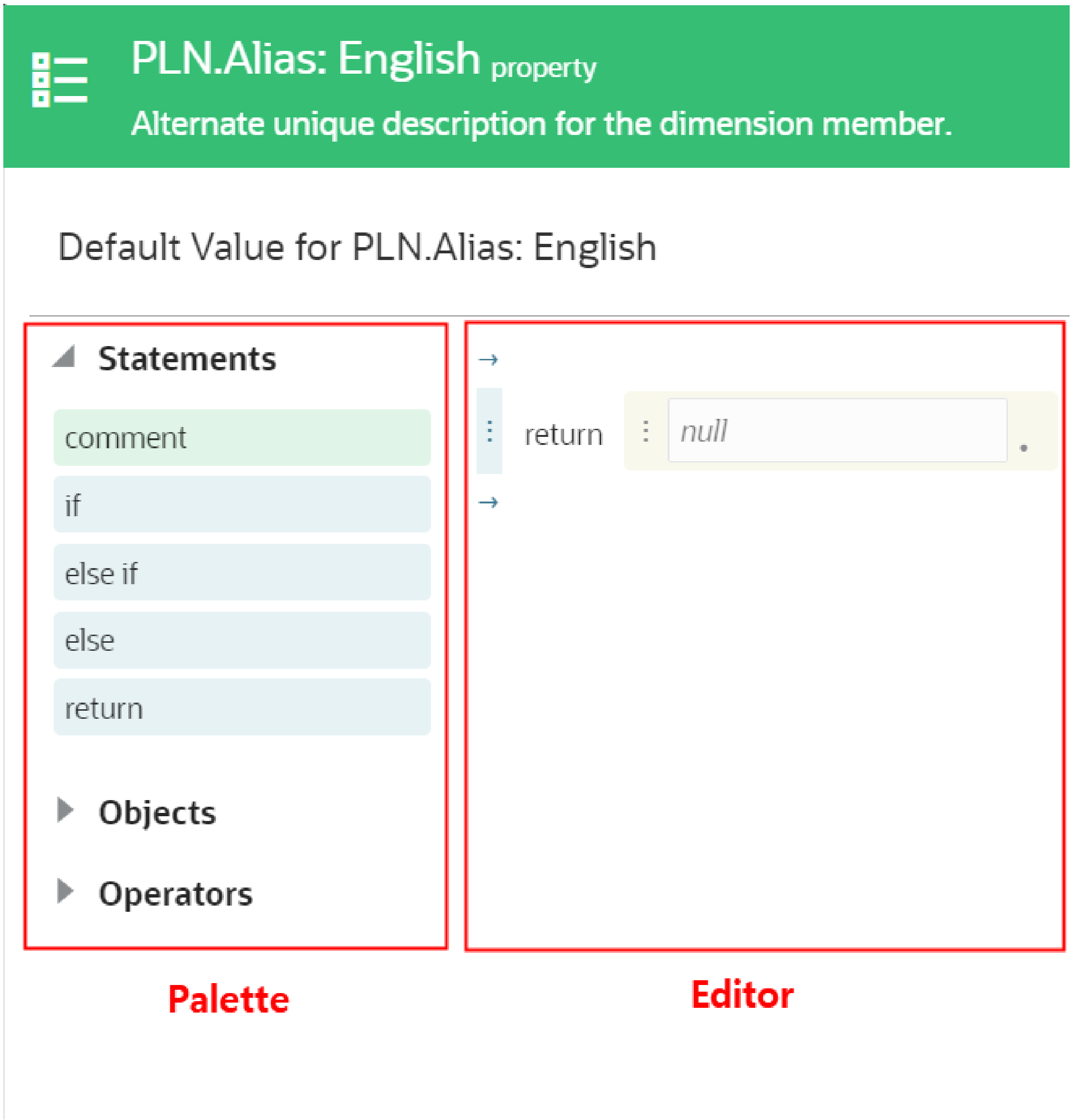
Tutorials provide instructions with sequenced videos and documentation to help you learn a topic.

Your Goal	Learn How
Build an expression in a node type converter to derive the parent of a node	Finding a parent node using a node type converter expression

Expression Builder Overview

The expression builder is made up of two areas:

- The **Palette** allows you to select fields to insert into the expression.
- The **Editor** enables you to compose the expression.



PLN.Alias: English property

Alternate unique description for the dimension member.

Default Value for PLN.Alias: English

Accessing the Expression Builder

You can access the expression builder in two modes:

- **Read Only mode**: Enables you to view the expression logic without making any changes. While in read only mode, you can copy or download the expression, capture it as text, or test it, but you cannot edit it.
- **Edit mode**: Enables you to create and edit the expression. While in edit mode, you can perform all of the functions listed below in the *Expression Builder Toolbar* section.

You access the expression builder in either mode by clicking the **Define Expression** *fx* button in a context where business logic is applied (for example, in a derived property). The mode in which the expression builder is opened depends on the mode you were in from the inspector where you accessed the expression builder. If you were in edit mode in the inspector, the expression builder is opened in Edit mode. If you were not in edit mode in the inspector, the expression builder is opened in Read Only mode.

Expression Builder Toolbar

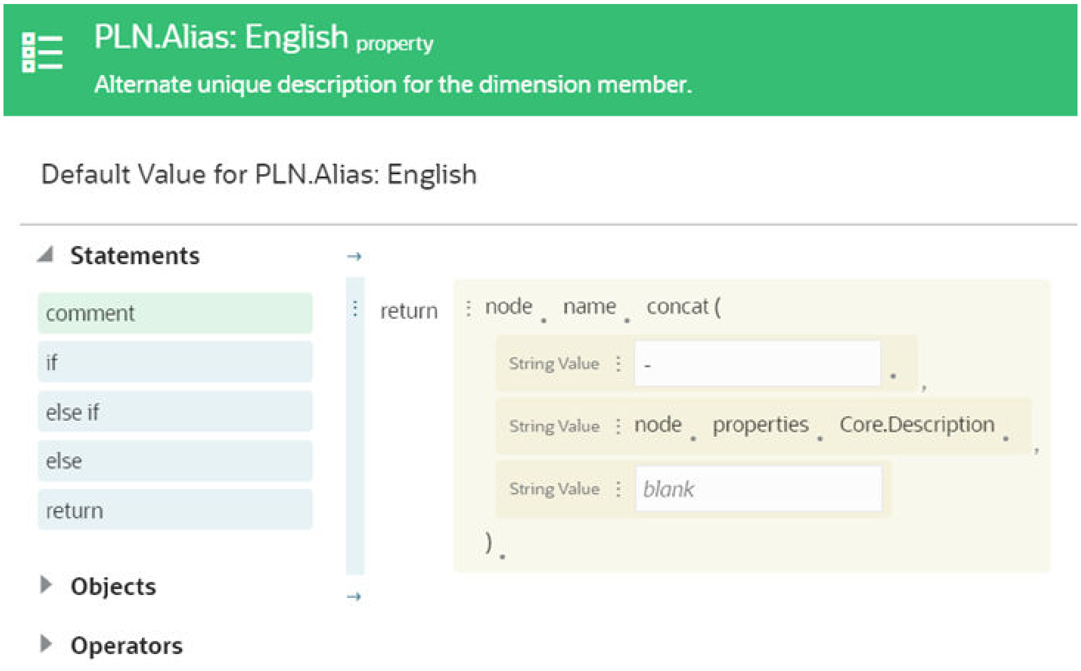
The toolbar in the expression builder enables you to perform functions like copying and pasting expression terms, loading expressions from a file, and testing your expressions. The following functions are available:

- **Copy** : Copies an expression. See Copying and Pasting Entire Expressions.
- **Paste** : Pastes an expression. See Copying and Pasting Entire Expressions.
- **Capture Expression as Text** : Copies a text version of the expression to your clipboard
- **Load from File** : Loads an expression from a `JSON` file. See Loading an Expression from a File.
- **Save to File** : Saves the current expression to a `JSON` file. See Saving an Expression to a File
- **Test Expression** : Tests the current expression. See Testing Expressions.
- **Undo** : Reverses the previously taken action
- **Redo** : Restores the action that was previously undone

Terminology

The following concepts are helpful to understand when using the expression builder:

- An **expression** consists of one or more lines of logic, called **statements**.
- **Statements** are discrete pieces of instruction that control the logic of an expression. There are three types of statements:
  - *Return* statements return a value.
  - *If* statements specify a list of statements to be performed when its condition is true. You can add *Else* and *Else If* statements to If statements to expand on the condition.
  - *Comments* enable you to annotate portions of the expression (for example, to identify where a change was made). They are informational only, and they do not get evaluated when the expression is run.
- **Expression terms** are made up of objects and operators that get evaluated to a value. In this example, an expression derives the value of an Alias property by concatenating the node name, plus a hyphen, and then the node's description:



PLN.Alias: English property

Alternate unique description for the dimension member.

Default Value for PLN.Alias: English

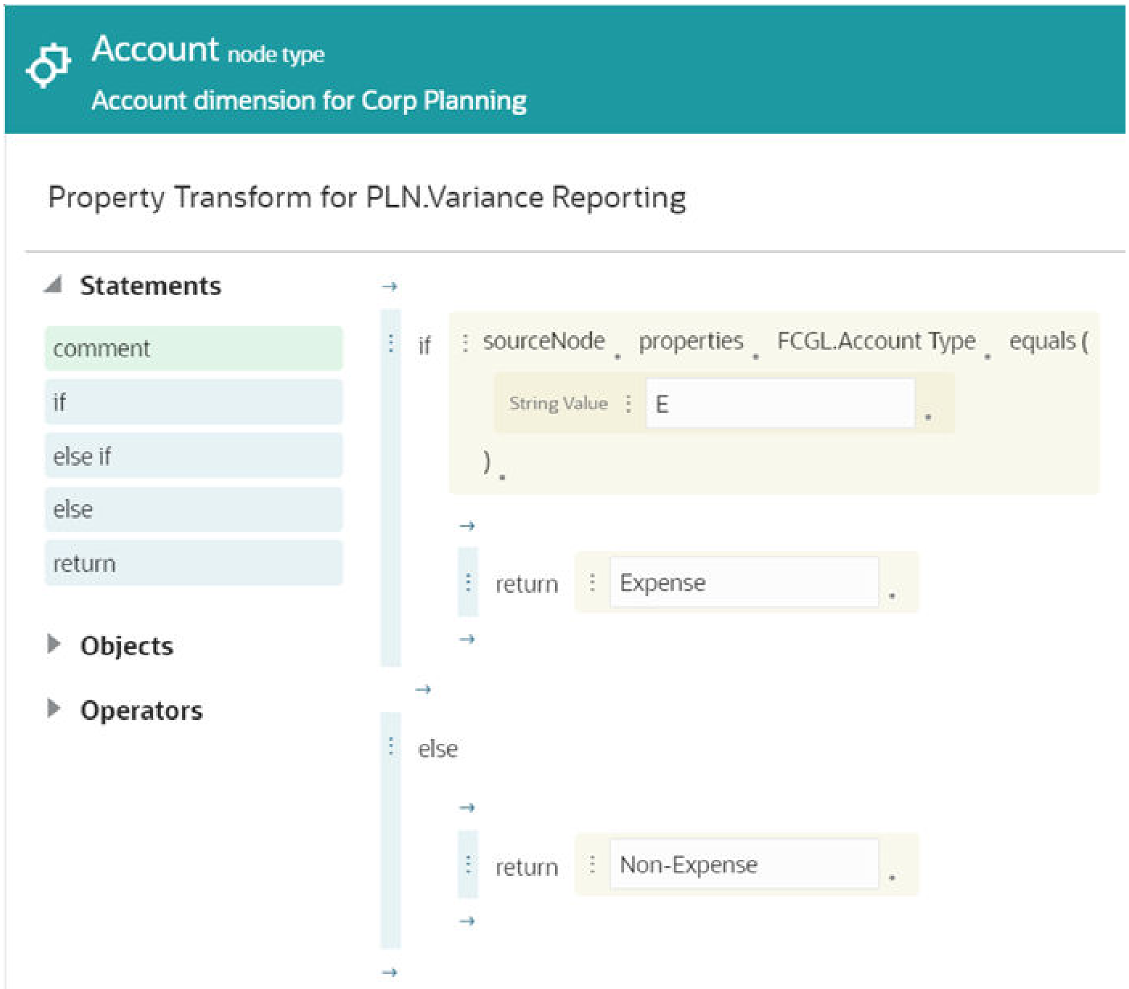
This expression has three expression terms (plus an optional placeholder to add a fourth), each indicated by a ⋮ icon:

- `Node.name.concat(`
- - (hyphen character)
- `Node.properties.Core.Description`

Expression terms are indicated with yellow backgrounds in the expression builder, and each expression term can be moved or deleted within the statement. You can also copy and paste expression terms within an expression. See Copying and Pasting in Expressions.

- **Objects** are the elements in an expression that represent values. Each object has a data type that determines what additional objects, methods, and attributes are available for that object. For example, if you select the `sourceNode` object in an expression, the expression builder will display only the fields that are applicable to node data type, such as `dimension` or `name`. See Data Types for Expression Terms.
  After you select an object in an expression, you can further refine your expression by selecting **attributes** and **methods** for that object.
  - **Attributes** describe an aspect of an object. For example, the `dimension` object has an attribute called `name`.
  - **Methods** perform actions on an object. For example, the `concat` method concatenates strings together, and the `length` method returns the length of a string. Some methods have *arguments*, which are input parameters for the method. A label describes the required input (for example, **String Value**).
- **Operators** perform logical comparisons in an expression. The expression builder supports the operators `AND` and `OR`.
- **Literals** are constant values that you manually enter in an expression. For example, in the statements `If Account.Type = E, return Expense`, both "E" and "Expense" are literals.

Let's look at an example of an expression to get familiar with the terminology:



Account node type

Account dimension for Corp Planning

Property Transform for PLN.Variance Reporting

if sourceNode . properties . FCGL.Account Type . equals (

String Value : E .

) .

return : Expense .

else

return : Non-Expense .

This is an expression on a node type converter that maps between a Corporate Planning account dimension and a Financials Cloud GL account dimension. Written out, this expression says:

For the `PLN.Variance Reporting` property, if the source node property `FCGL.Account Type` equals "E", return "Expense". If not, return "Non-Expense".

This expression in the screenshot contains these elements:

1. Three *statements*:
   - If FCGL.Account Type equals "E"
   - Return "Expense"
   - Return "Non-Expense"

> **Note:**
>
> When building the expression, you do not have to add an Else If statement. When the expression is run, if the IF statement is not true, the logic moves to the next statement. See Building Expressions.

2. Four *expression terms*:
   - FCGL.Account Type equals
   - "E"
   - "Expense"
   - "Non-Expense"
3. One *object*: sourceNode.
4. Two *attributes*: FCGL.Account Type and properties.
5. One *method*: equals.
6. Three *literals*: E, Expense, Non-Expense.

Building Expressions

Expressions are made up of statements, which contain expression terms made up of objects and operators, which can further be refined by selecting methods and attributes.

In this topic, we will cover the general steps of using the expression builder to create an expression. For a detailed example, see Expressions: Detailed Example.

You create an expression by selecting either an *If* or a *Return* statement and then building the expression terms in that statement with objects, methods, attributes, and operators.

For details on the data types in an expression, see Data Types for Expression Terms.

When editing an expression, click ↶ to undo your last action, and click ↷ to redo it.

See the following topics for more information about building expressions:

- Working with Statements
- Working with Comments
- Working with Objects and Operators
- Working with Literal Values
- Using Lookup Sets in Expressions
- Copying and Pasting in Expressions

Working with Statements

Statements control the logic of an expression.

There are two types of statements:

- *Return* statements return a value. In order for a return statement to be valid, the result must match the data type of the property that the expression is being defined for.
- *If* statements specify a list of statements to be performed when its condition is true. In order for an *If* statement to be valid, the result must be a Boolean value. You can use the *Else* and *Else If* statements to expand on the condition.

To insert a statement:

1. From the Statements section of the palette, drag a statement to the editor, or click → and select the statement from the drop down list.

You can insert a statement anywhere there is a blue arrow in the editor. When you add a statement to the editor, a vertical blue bar is added to the editor to indicate a statement block. You can nest statements within other statements, and each statement is indicated by a blue bar.

2. Select objects and operators for your expression. After you select an object, you can select that object's attributes or methods. See Working with Objects and Operators

3. To remove a statement block, click ⋮ in the statement block and select **Remove**

When you begin selecting fields in a statement, the expression term is outlined in red to indicate that it is not valid. You cannot save an expression until all of the statements in the expression are valid, as follows.

- `If` statements must evaluate to a Boolean value.
- `Return` statements must return the expected data type:
  - For property transformations and derived properties, the expected data type is the data type of the property that is being transformed or derived. See Properties.
  - For custom validations, the expected data type is a Boolean value.

Working with Comments

Use comments to add notations to expressions. The information in a comment block is informational only, and it will not be evaluated as part of the expression.

To add or remove a comment:

1. To add a comment, from the Statements section of the palette, drag a comment to the editor, or click → and select **comment** from the drop down list. The expression builder guides you to where you can insert a comment. For example, you can insert a comment before or after an If statement, but not within the statement itself.
   When you add a comment to the editor, a vertical blue bar is added to the editor to indicate a statement block.

2. To remove a comment, click ⋮ in the statement block around the comment and select **Remove**.

Working with Objects and Operators

Objects and operators get added to expression terms inside statements.

To add objects and operators:

1. In the yellow expression term for the statement, use the palette or click ⋮ to select an object or an operator. Initially, you can select these objects:
   - **node** (for derived properties) or **sourceNode** (for property transformations)
   - **request**
   - **lookupSets**
   - **currentProperty** (derived properties only)
   - **Boolean**
   - **integer**
   - **float**

- **now**
- **null**
- **string**
- **today**

You can also add the **And** or **Or** operators to a statement to link expression terms.

2. Select methods and attributes to your objects by clicking the dot next to each element and selecting from the drop down list. The fields that are available to be selected depend on the field's data type. For example, if you select a field after a node, you can select the node's ancestors, children, dimension, name, node type, parent, or properties. If you select a field after a dimension, you can select the dimension's name attribute. See Data Types for Expression Terms.
3. To remove methods and attributes from your objects, click the dot next to the element that you want to remove and select **Clear**.

Data Types for Expression Terms

When you select a field such as an object or an operator in an expression, the data type of that field determines the objects, attributes, and methods that can be selected next in that expression. The following topics provide more detail about the data types of the fields that can be added to expressions:

- Boolean
- Date and Timestamp
- Current Date and Time
- Integer and Float
- Integer List and Float List
- List and String List
- LookupSets
- Memo
- Node
- Node Data Type Property
- Node Collections and Node List Data Type Properties
- Operators
- Properties
- Request
- Strings, Numeric Strings, and Sequences

Boolean

Boolean data types indicate if a condition is true or false. You can add these methods to the Boolean data type:

Table 25-1 Boolean Methods and Attributes

Method	Description
`and`	Enables you to join multiple expression terms as an alternative to using the And operator. Returns True if both the current expression and the specified condition are true.
`equals`	Returns True if the Boolean contains a value that equals a specified field.
`format`	Converts the Boolean value to a string. Select from the following string formats: • True/False • T/F • Yes/No • Y/N • 1/0
`isNull`	Returns True if the Boolean has a null value
`not`	Returns the inverse of the Boolean value
`or`	Enables you to join multiple expression terms as an alternative to using the Or operator. Returns True if either the current expression or the specified condition are true.

You can add literal values to Boolean data types. See Working with Literal Values.

Current Date and Time

Use one of these objects to return the current date or time in an expression.

Table 25-2 Current Date and Time Objects

Object	Description	Attributes and Methods
`now`	Returns the current timestamp.	• `after` • `afterOrEqual` • `before` • `beforeOrEqual` • `equals` • `toDate` See Date and Timestamp.
`today`	Returns the current date.	You can specify any of the date attributes in Date and Timestamp except `isNull`.

CurrentProperty

Use the `currentProperty` object in a derived property expression to access property metadata such as the property name and description.

Note:

The `currentProperty` object is available for derived property expressions only.

The following attributes and methods are available for the `currentProperty` object:

Attribute or Method	Description	Parameter	Example
`Description`	Attribute that returns the Description of the current property for the expression	None	`return currentProperty.description` returns the description of the current property
`IsValueAllowed` (Node data type properties only)	Method that determines whether a specified value exists in the assigned node set and the allowed node types for a node data type property.	Node name to be checked	`if currentProperty.isValueAllowed("100"), return "100"` returns a value of 100 only if that value is allowed for the current property
`Name`	Attribute that returns the Name of the current property for the expression	None	`if currentProperty.name.startsWith("EDM")` will perform the rest of the expression if the name of the current property starts with EDM.

Date and Timestamp

The Date data type contains a date value, and the Timestamp data type contains a date and time value. Date and Timestamp methods must evaluate to a Boolean value. You can add these methods to the date or timestamp data types:

Table 25-3 Date and Timestamp Methods

Method	Description
`after`	Returns True if the date or timestamp value is after the specified point in time
`afterOrEqual`	Returns True if the date or timestamp value is after or on the specified point in time
`before`	Returns True if the date or timestamp value is before the specified point in time
`beforeOrEqual`	Returns True if the date or timestamp value is before or on the specified point in time
`equals`	Returns True if the date or timestamp value equals the specified point in time

Table 25-3 (Cont.) Date and Timestamp Methods

Method	Description
`format` (Date data type only)	Convert a date into a formatted string data type. For example, `node.properties.EBS.StartDate.format(MM/dd/yyyy)` converts the start date into a formatted string. See Date Formatting Symbols for date and time formatting.
`isNull`	Returns True if the date or timestamp has a null value
`minus`	Returns a date with the specified period subtracted. You must specify the amount and the unit (Days, Weeks, Months, Years) of time to subtract.
`plus`	Returns a date with the specified period added. You must specify the amount and the unit (Days, Weeks, Months, Years) of time to add.
`toDate` (Timestamp data type only)	Converts a timestamp into a date by truncating the time portion.
`toTimestamp` (Date data type only)	Converts a date into a timestamp by adding the time of day to the date. Specify the following: • START_OF_DAY: Adds 00:00:00 based on UTC time zone • END_OF_DAY: Adds 23:59:59 PM based on UTC time zone

Note:

You cannot enter literal values as date or timestamp values in expressions. For example, you can create a custom validation for `node.properties.startDate before node.properties.endDate`, but not `node.properties.startDate before "01-01-2019"`.

Integer and Float

An integer is a whole number and a floating-point value, or float, is a number that has a decimal place. You can add the following methods to the integer and float data types.

- `add`
- `divide`
- `equals`
- `greaterThan`
- `greaterThanOrEqual`
- `lessThan`
- `lessThanOrEqual`

- `isNull`
- `multiply`
- `modulo` (Integer only): Returns the remainder when one integer is divided by another.
- `subtract`
- `toNumericString`: Converts an integer or float value to a numeric string value. **Optional**: Specify the `Pad length`: Total length of the numeric string to be zero padded up to, with 0 meaning no padding is added. For example, if you set the Padding to 10 and your numeric string value is 7 digits long, three zeroes are prepended to the string to get to 10 places.
- `toString`: Converts an integer or float value to a string value.

You can add literal values to integer and float data types. See Working with Literal Values.

Integer List and Float List

Use the `toIntegerList` or `toFloatList` methods to return an integer or float value from each node in a node list.

You can add the following methods to an Integer or Float list:

- `Average`: Calculates an average from each numeric value in an Integer or Float list
- `Sum`: Sums a numerical value from each numeric value in an Integer or Float list.
- `Size`: Number of float values in an Integer or Float list

List and String List

The list data type contains a sequence of strings. String list objects allow you to select lists of values, either returned from a List data type property, by converting a node list into a list of strings using the `toStringList` method, or by splitting a string into a string list using the `Split` method.

After you add a list data type property or a string list object, you can add the following methods. After you add a method, you must specify a condition for that method that can be evaluated to a Boolean value.

Method	Description	Data Type of Returned Object	Parameters	Example
`Any`	Returns True if any of the strings in a string list match the specified condition	Boolean	The condition that the string list object gets evaluated against	`node.properties.PLN.UDA(Account).any.stringItem.matches("UDA"))` will return True if any of the strings in the string list match "UDA".

Method	Description	Data Type of Returned Object	Parameters	Example
`Count`	Returns a count of the strings in a list that match a specified condition	Integer	The condition that the string list object gets evaluated against	`node.properties.PLN.UDA (Account).count.stringItem.greaterThan("ABC"))` will return the number of strings in a string list whose value is greater than "ABC."
`Find`	Returns the first string in a string list that matches a specified condition	String	The condition that the string list object gets evaluated against	`node.properties.PLN.UDA (Account).find.stringItem.equals("Expense"))` returns the first string in a string list of a selected node whose name equals "Expense".
`Get`	Returns the string in a string list at the specified index	String	The index to return the string from **Note**: Positive index numbers (or zero) count from the first index to the last, with zero being the first index. Negative index numbers count from last to first.	`node.properties.PLN.UDA (Account).get(0)` returns the first string in the PLN.UDA (Account) string list.
`IndexOf`	Returns the position of the specified string in a string list. **Note**: Returns `-1` when the specified string is not found.	Integer	The specified string that you want to return the position for.	`node.ancestors[name].toStringList.indexOf(Texas)` returns the starting position of the specified string "Texas" in the string list of ancestor names for a node.
`Intersect`	Returns the common values that exist in both specified lists.	String List	A second string list that you are comparing to the first string list for common values.	`node.ancestors[name].toStringList.intersect(node.properties.UDA)` returns the common values that exist in both the list of ancestor names for a node and the node.properties.UDA string list.

Method	Description	Data Type of Returned Object	Parameters	Example
IsEmpty	Returns True if the property contains either a null value or an empty string.	Boolean	None	node.properties.PLN.UDA (Account).IsEmpty returns True if the value of the PLN.UDA (Account) string list is null or empty.
isNull	Returns True if the property contains a null value. Note: The isNull method is available for list data type properties only. You cannot use isNull for string lists from the toStringList or Split methods.	Boolean	None	node.properties.PLN.UDA (Account).isNull returns True if the value of the PLN.UDA (Account) string list is null.
Join	Concatenates string values from a list into a delimited string using the specified delimiter	String	The delimiter that separates each string	return.node.properties.PLN.UDA (Account).join("|") returns a string that consists of all of the strings in the PLN.UDA (Account) string list concatenated together and separated by "|".
Reverse	Returns the string list values in the reverse order.	String List	None	node.ancestors[name].toStringList.reverse returns a list of ancestor names for a node in reverse order.
Size	Returns the number of strings in a string list **Note**: The Size method is different from Count in that it does not require a predicate condition.	Integer	None	return.node.properties.PLN.UDA (Account).size returns the number of strings in the PLN.UDA (Account) string list as a string value.

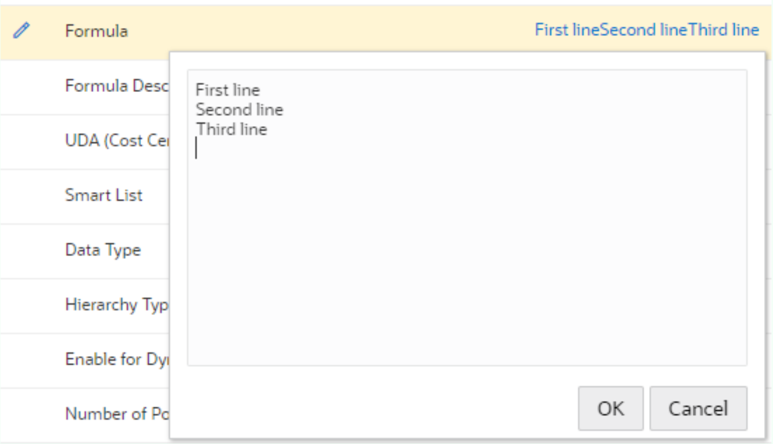
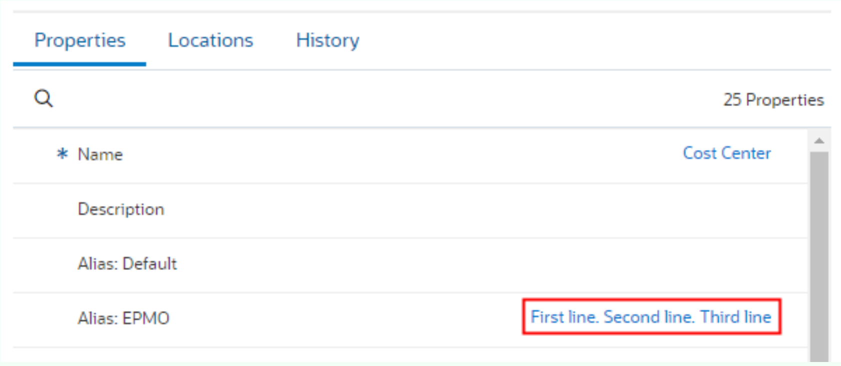
LookupSets

LookupSets enable you to associate keys with values in order to transform an input value to an output value. When you select a LookupSet object, the only attribute that you can select is the name of the lookup set. After you select the name of the lookup set, you can select the find method to find the key and value pair to look for in the expression.

See Working with Lookup Sets.

Memo

`Memo` objects are large strings that allow formatting and whitespace characters, such as line feeds and Tab characters. They use the same methods and attributes as the `String` data type. See Strings, Numeric Strings, and Sequences.

> **Tip:**
>
> For methods such as `matches` or `replace` that use Java regular expressions, you can construct regular expressions to identify formatting characters in memo fields. For example, `return node.properties.PLN.Formula.replace("\R", ". ")` replaces the line separators in the PLN.Formula property with a period and then a space, which converts this:
>
> 
>
> Formula: First lineSecond lineThird line
>
> First line
> Second line
> Third line
>
>
>
> to this:
>
> 
>
>
>
> | | |
> |---|---|
> | * Name | Cost Center |
> | Description | |
> | Alias: Default | |
> | Alias: EPMO | First line. Second line. Third line |

Node

When you select a `node` object in a derived property expression or a `sourceNode` object in a property transformation expression, you can select node attributes depending on the data type of the node object:

Table 25-4 Node Objects and Attributes

Node Object	Description	Attributes and Methods of Object
`Actions`	Returns a list of request action objects for the node in a request. (Available for subscription filters, policy filters, and custom validations only.)	- `Any` - `Count` - `Filter` - `Find` - `Get` - `IsEmpty` - `Size` Action object: see *Action Object Attributes* table, below.
`AllAncestors`	Returns all ancestor node objects combined from all node locations in the node set.	See Node Collections and Node List Data Type Properties.
`Ancestors`	Returns ancestor node objects for a node instance in the node set	See Node Collections and Node List Data Type Properties.
`BoundAncestors`	In hierarchy set bound viewpoints, returns a list of ancestor nodes for a node in the bound viewpoint. In node type bound or unbound viewpoints, returns ancestor node objects for a node instance (the same result as when using **Ancestors**).	See Node Collections and Node List Data Type Properties.
`BoundLocations`	In hierarchy set bound viewpoints, returns a list of locations for a node in the bound viewpoint. In node type bound or unbound viewpoints, returns node locations (the same result as when using **Locations**). **Optional**: Enable **Exclude Implicit Shares** to filter the returned list of bound locations to only node instances under unique parents. **Tip**: Using the bound versions of Ancestors and Locations enables you to access the full scope of the bound viewpoint when using an expression in a maintenance viewpoint.	See Node Collections and Node List Data Type Properties.
`Children`	Child node objects for a node.	See Node Collections and Node List Data Type Properties.

Table 25-4 (Cont.) Node Objects and Attributes

Node Object	Description	Attributes and Methods of Object
CompareTo	Compare two node locations with these options: • SAME_LOCATION: Node location is the same as a specified node location • SAME_PARENT_AND_NODE: Node location has the same parent as a specified node location • SUBSET_LOCATION: Node location has a set of contiguous ancestors that is the same as a specified node location **Tip**: This enables you to build an expression that returns different results based on a comparison of node locations. For example, you can derive the value of a Data Storage property as stored or shared based on whether a particular node is the first location in the hierarchy and whether it shares a parent or ancestors with other node locations.	Node for the compare.
Dimension	Node dimension	name attribute
HierarchySet	Hierarchy set of a node in a viewpoint.	name attribute

Table 25-4 (Cont.) Node Objects and Attributes

Node Object	Description	Attributes and Methods of Object
`Locations`	Node locations (can contain node instances). - **Optional**: Enable **Exclude Implicit Shares** to filter the returned list of locations to only node instances under unique parents. - **Optional**: Enable **Sort** to return the locations in the order in which they appear in a hierarchy viewpoint. This allows business logic for shared nodes based on hierarchy order to be implemented. For example, you can create a derived Data Storage property for Planning dimensions which assigns the first member instance a Stored value and other member instances in the hierarchy a Shared value.	See Node Collections and Node List Data Type Properties.
`Name`	Node name	See Strings, Numeric Strings, and Sequences
`NextSibling`	Node sibling after a node in the hierarchy. If this is the last node, then the `NextSibling` attribute returns `Null`.	- `dimension` - `name` - `nodeType` - `properties`
`NodeSet`	Node set for a node. (Available for calculated and stored Parent properties only)	`name` attribute
`NodeType`	Node type for a node.	`name` attribute
`Parent`	Node parent	Parent also has the Node data type, which means that you can select other node attributes such as the dimension or the node type of the parent. However, you cannot select the parent of a parent.
`PreviousSibling`	Node sibling before a node in the hierarchy. If this is a first node without a previous sibling, then the `PreviousSibling` attribute returns `Null`.	- `dimension` - `name` - `nodeType` - `properties`
`Properties`	Node properties.	See Properties.

Table 25-4 (Cont.) Node Objects and Attributes

Node Object	Description	Attributes and Methods of Object
`Siblings`	Sibling node objects for a node instance.	See Node Collections and Node List Data Type Properties.

Table 25-5 Action Object Attributes

Object Attribute	Description	Attributes and Methods of Returned Object
`CreatedBy`	User who created the request action	• `isMemberOf` Parameter: the name of the group to check • `username`
`ModifiedBy`	User who modified the request action	• `isMemberOf` Parameter: the name of the group to check • `username`
`PropertyName`	Name of property in the request action	See Method or Attribute in Strings, Numeric Strings, and Sequences.
`Type`	The type of request action	• `isAdd` • `isDelete` • `isInsert` • `isMove` • `isRemove` • `isRename` • `isReorder` • `isUpdate`
`ViewpointName`	Name of the viewpoint in the request action.	See Method or Attribute in Strings, Numeric Strings, and Sequences.

Node Data Type Property

When you add a node data type property to an expression, you can perform a property transformation, derive a property value, enforce a custom validation, filter subscription requests, filter approval or notification polices, or filter extracts using a property with a data type of Node. You can select the following attributes of the node that is being referenced by the property.

> **Note:**
>
> The node being referenced in the node data type property can be outside of the current data chain.

Table 25-6 Node Data Type Objects and Attributes

Node Data Type Object	Description	Attributes and Methods of Object
`Dimension`	Dimension of the node referenced in the property	`name` attribute
`IsNull`	Returns True if the property is null	See Methods or Attributes in Boolean.
`Locations`	Returns a list of node locations in the associated node set. The list contains node instances.	• Name • Dimension • Node Type • Parent • Node, relationship, and inherited properties
`Name`	Name of the node referenced in the property	See Strings, Numeric Strings, and Sequences
`NodeType`	Node type of the node referenced in the property	`name` attribute
`Properties`	Defined or derived node level properties of the node referenced in the property	See Properties.

Node Collections and Node List Data Type Properties

A node collection is a list of node values that comes from a function, such as ancestors, locations, or children. A node list data type property is a list of node values that comes from another property.

When you add a node list data type property to an expression, you can perform a property transformation, derive a property value, enforce a custom validation, filter subscription requests, filter approval or notification polices, or filter extracts using a property with a data type of Node List.

> **Note:**
>
> The nodes being referenced in the node list data type property can be outside of the current data chain.

After you add a node collections or a node list data type object, you can add the following methods to that object. After you add a method, you must specify a condition for that method (depending on the method, this will be either a value that can be evaluated to a Boolean value, or a float, integer, or string value to be returned for each node in the list for the `toFloatList`, `toIntegerList`, and `toStringList` methods).

Table 25-7 Node Collections and Node List Data Type Methods and Attributes

Method	Description	Returned Object Data Type	Parameters	Example
`Any`	Returns True if any of the nodes in a node list match the specified condition.	Boolean	The condition that the node list object gets evaluated against.	`any.childNode.properties.AccountType.equals(Expense)` will return True if any of the children of the selected node have an account type of Expense. Predicate node: See *Predicate Node Attributes and Methods* table, below.
`Count`	Returns a count of the nodes in a node list that match a specified condition. **Tip**: If you want to return the total number of nodes without specifying a predicate condition, use `Size` instead.	Integer	The condition that the node list object gets evaluated against.	`count.childNode.properties.SourcePlanType.equals(Plan1)` returns the number of children for a selected node whose source plan type is "Plan1". Predicate node: See *Predicate Node Attributes and Methods* table, below.
`Filter`	Returns a list of nodes which match the specified condition.	Node List **Note**: The type of node list that is returned is based on the type of collection that you are performing the method on.	The condition for the filter (must return a Boolean value).	`ancestors.filter(ancestorNode.properties.Level.lessThan(4))` returns a list of ancestor nodes on the first three levels. Predicate node: See *Predicate Node Attributes and Methods* table, below.
`Find`	Returns the first node in a node list that matches a specified condition.	Node **Note**: The type of node that is returned is based on the type of collection that you are performing the method on.	The condition that the node list object gets evaluated against.	`find.childNode.properties.Name.startsWith(A_)` returns the first child of a selected node whose name starts with the letter "A_" Predicate node: See *Predicate Node Attributes and Methods* table, below. Returned node: See *Returned Node Attributes and Methods* table, below.

Table 25-7 (Cont.) Node Collections and Node List Data Type Methods and Attributes

Method	Description	Returned Object Data Type	Parameters	Example
`Get`	Returns the node in a list of nodes at the specified index	Node **Note**: The type of node that is returned is based on the type of collection that you are performing the method on.	The index (position in the list) of the node to get. The position that you specify must be an integer value. Positive index numbers (or zero) count from the first index to the last, with zero being the first index. Negative index numbers count from last to first.	`node.children.get(0)` gets the first child node of a node Returned node: See *Returned Node Attributes and Methods* table, below.
`isEmpty`	Returns `True` if the node list empty.	Boolean	None	`node.properties.PLN.UDA (Account).IsEmpty` returns True if the value of the PLN.UDA (Account) string list is empty.
`isNull`	Returns `True` if the node list null.	Boolean	None	`node.properties.PLN.UDA (Account).IsNull` returns True if the value of the PLN.UDA (Account) string list is null.
`Max`	Returns a single node with the maximum value of the specified data type. If multiple nodes in the list have the maximum value, the first node is returned.	Node **Note**: The type of node that is returned is based on the type of collection that you are performing the method on.	The assessed value to determine which node in the list has the maximum value. Allowed data types for the value are String, Integer, Float, Date, Timestamp, or Boolean.	`node.children.max.(childNode.name)` returns the maximum child name based on data type. For example, if the children node names are integers, it would return the highest numeric value.
`Min`	Returns a single node with the minimum value of the specified data type. If multiple nodes in the list have the minimum value, the first node is returned.	Node **Note**: The type of node that is returned is based on the type of collection that you are performing the method on.	The assessed value to determine which node in the list has the minimum value. Allowed data types for the value are String, Integer, Float, Date, Timestamp, or Boolean.	`node.children.min.(childNode.properties.Salary)` returns the subordinate under a manager which has the lowest salary in an employee hierarchy.

Table 25-7 (Cont.) Node Collections and Node List Data Type Methods and Attributes

Method	Description	Returned Object Data Type	Parameters	Example
`Reverse`	Returns the node list values in the reverse order.	Node List **Note**: The type of node list that is returned is based on the type of collection that you are performing the method on.	None	`node.ancestors.reverse` returns a list of ancestors for a node in reverse (bottom up) order.
`Size`	Returns the number of nodes in a node list. **Tip**: The `Size` method is different from `Count` in that it does not require a predicate condition.	Integer	None	`return node.children.size.toString` returns the number of children for a node as a string value.
`toFloatList`	Returns a float value from each node in a node list For methods, see Integer List and Float List.	List of Float Values	None	`return node.children.toFloatList(childNode.properties.Ownership%).sum` returns the sum of the ownership percentage for all children of a node.
`toIntegerList`	Returns a integer value from each node in a node list For methods, see Integer List and Float List.	List of Integer Values	None	`return node.children.toIntegerList(childNode.properties.Employees).size` returns the number of employees across all child nodes.
`toStringList`	Returns a string value from each node in a node list. See List and String List for the methods that you can add to a string list.	List of String Values	None	`return node.ancestors.toStringList(ancestorNode.name).join("|")` returns a string list of ancestors for a node separated by "`|`".

The following attributes and methods are available for the predicate node when using `any`, `count`, `filter`, or `find`.

Table 25-8 Predicate Node Attributes and Methods

Attribute or Method	Description
`compareTo`	Compare two node locations with these options: • `SAME_LOCATION`: Node location is the same as a specified node location • `SAME_PARENT_AND_NODE`: Node location has the same parent as a specified node location • `SUBSET_LOCATION`: Node location has a set of contiguous ancestors that is the same as a specified node location
`dimension`	Node dimension
`hierarchySet`	Hierarchy set of a node in a viewpoint.
`name`	Node name
`nodeType`	Node type for a node
`properties`	Node properties

The following attributes and methods are available for the returned node when using `find` or `get`.

Table 25-9 Returned Node Attributes and Methods

Attribute or Method	Description	Available For These Node Collections
`boundLocations`	In hierarchy set bound viewpoints, returns a list of locations for a node in the bound viewpoint. In node type bound or unbound viewpoints, returns node locations (the same result as when using **Locations**).	• ancestors • siblings
`compareTo`	Compare two node locations with these options: • `SAME_LOCATION`: Node location is the same as a specified node location • `SAME_PARENT_AND_NODE`: Node location has the same parent as a specified node location • `SUBSET_LOCATION`: Node location has a set of contiguous ancestors that is the same as a specified node location	all
`dimension`	Node dimension	all

Table 25-9 (Cont.) Returned Node Attributes and Methods

Attribute or Method	Description	Available For These Node Collections
`hierarchySet`	Hierarchy set of a node in a viewpoint.	• ancestors • children • locations • siblings
`locations`	Node locations (can contain node instances).	• ancestors • node list properties • siblings
`name`	Node name	all
`nextSibling`	Node sibling after a node in the hierarchy. If this is the last node, then the `NextSibling` attribute returns `Null`.	• ancestors • locations
`nodeType`	Node type for a node	all
`previousSibling`	Node sibling before a node in the hierarchy. If this is a first node without a previous sibling, then the `PreviousSibling` attribute returns `Null`	• ancestors • locations
`properties`	Node properties	all
`siblings`	Sibling node objects for a node instance.	• ancestors • locations

Operators

Operators, such as `AND` and `OR`, perform logical comparisons in expressions. You can select the `not` method for operators.

Properties

The property data type enables you to select the properties of a node. For property transformations, because the node type of the source node is known, only the properties for that node type are displayed. However, when displaying properties for derived properties, because the node type is not known, all of the properties that are valid for the application are displayed.

The data type of the property that you select determines how the property can be used in an expression:

- Properties with any data type can be referenced in an expression.
- Properties with any data type except for Memo and Timestamp can have their values derived or transformed.

After you select a property, the data type of that property determines the methods and attributes that can be selected for that property. For example, the `name` property is a string, and you can select any of the methods and attributes for strings. The `Custom.Allow Posting` property is a Boolean, and therefore you can only select the `not` method.

You can create expressions that reference derived properties for `parent`, `ancestors`, `allAncestors`, `boundAncestors`, `boundLocations`, and `locations`. After selecting one of these terms in your expression, the list of properties that can be referenced for that term includes properties which have a derived value for the application or node type (the current property is included in the list for `parent` only). See Derived Properties.

You cannot reference derived properties for `children`, `previousSibling`, `nextSibling`, or `siblings`.

When calculating the value of a property, you cannot reference the value of the same property in a node collection, such as `children` or `siblings`.

> **Note:**
>
> The expression builder prevents you from creating an expression that includes circular references (for example, Property A is based on Property B, but the expression to derive Property B references Property A). If a circular reference is detected, the expression term is outlined in red and a validation message is displayed that indicates the problem.

Request

The request object data type enables you to access attributes of a request in subscription filters, approval and notification policies, and custom validations. After you add a method, you add attributes and parameters to return a Boolean value.

Request Attribute	Description	Attributes and Methods of Returned Object
`attachments`	List of attachments for the request	Attachments list: • `any` • `count` • `find` • `get` • `isEmpty` • `size` Attachment object: see *Request Attachment and Comment Object Attributes* table, below.
`comments`	List of request level comments	Comments list: • `any` • `count` • `find` • `get` • `isEmpty` • `size` Comment object: see *Request Attachment and Comment Object Attributes* table, below.

Request Attribute	Description	Attributes and Methods of Returned Object
`createdBy`	User who created the request	- `isMemberOf` Parameter: the name of the group to check - `username`
`description`	Description of the request	See Method or Attribute in Strings, Numeric Strings, and Sequences.
`isNull`	Returns True if the request context is null.	See Methods or Attributes in Boolean.
`owner`	Assigner or submitter depending on the request stage	- `isMemberOf` Parameter: the name of the group to check - `username`
`priority`	Priority of the request	- `isNone` - `isLow` - `isMedium` - `isHigh`
`stage`	The request stage	- `isApprove` - `isClosed` - `isCommit` - `isSubmit`
`status`	The request status	- `isCompleted` - `isDraft` - `isInFlight` - `isPushedBack` - `isRejected`
`title`	The title of the request	See Method or Attribute in Strings, Numeric Strings, and Sequences.
`type`	The request type	- `isConsolidation` - `isImport` - `isInteractive` - `isLoad` - `isSubscription`

For the request attachments and comment objects, the following attributes are available.

Table 25-10 Request Attachment and Comment Object Attributes

Object Attribute	Description	Attributes and Methods of Returned Object
Attachment Objects		
`name`	Name of the attachment file	See Method or Attribute in Strings, Numeric Strings, and Sequences.

Table 25-10 (Cont.) Request Attachment and Comment Object Attributes

Object Attribute	Description	Attributes and Methods of Returned Object
fileOrigin	Origin (system or user) of the attachment file	• isSystem (request load file) • isUser (request attachment)
createdBy	User who created the attachment	• isMemberOf Parameter: the name of the group to check • username
Comment Objects		
value	The contents of the comment in HTML format. For example, for a comment with the text, "Approved", the comment value object contains" <p>Approved</p>"	See Method or Attribute in Strings, Numeric Strings, and Sequences.
createdBy	User who created the comment	• isMemberOf Parameter: the name of the group to check • username

Strings, Numeric Strings, and Sequences

Strings in expressions include fields like property names and descriptions, as well as literal values. After you select a string in an expression, there are several methods and attributes that you can select for that string. For example, you can concatenate a node's name and description and use the resulting string in an Alias property, or you can return just a portion of a source string by using the substring method. Oracle Fusion Cloud Enterprise Data Management

Numeric strings and sequence data types are strings that support numeric values only (0-9). They are limited to 40 characters.

You can add literal values to string data types. See Working with Literal Values.

Many of the string methods require you to enter additional parameters. The expression builder creates a new expression term for those parameters when you select that method. For example, when you select a concat method, one additional expression term is added for you to specify the string to append. When you select a replace method, two additional expression terms are added for you to specify the text to search for and the text to replace it with.

The following table describes the methods and attributes that you can use to manipulate string values in the expression builder, as well as the parameters for those fields.

Method or Attribute	Usage	Parameters
`add` (numeric string only)	Adds an integer value to the numeric string **Note:** Zero padding on the numeric string is retained.	The integer value to add.
`Concat`	Concatenates (or appends) two or more string fields together and returns the combined string as the result. **Note:** You can append multiple strings in a single `concat` statement. When you have multiple strings to concatenate, you can click ⋮ in an expression term and select **Insert Above** or **Remove** to move that field around or remove it.	The string to append

Method or Attribute	Usage	Parameters
`endsWith`	Returns `True` if the original string ends with the specified string. **Note:** The specified string is case-sensitive.	The specified string to check if the original string ends with.
`equals`	Returns `True` if the string equals the specified string.	The specified string to check if the original string equals.
`greaterThan`	Returns `True` if the string value is greater than the specified value (both must be same data type).	The specified string to check if the original string is greater than.
`greaterThanOrEqual`	Returns `True` if the string value is greater than or equal to the specified value (both must be same data type).	The specified string to check if the original string is greater than or equal to.
`indexOf`	Returns the starting position of the specified string in the original string. **Note:** The `indexOf` method returns an integer, which cannot be used directly as a return value. You can use this integer to determine the starting place for a substring. See Integer and Float.	The string that you want to find the starting position for.
`IsEmpty`	Returns `True` if the property contains either a null value or an empty string.	No parameters required
`isNull`	Returns `True` if the string property contains a null value.	No parameters required

Method or Attribute	Usage	Parameters
`length`	Counts the number of characters in a string The `length` attribute is an integer. See Integer and Float.	No parameters required
`lessThan`	Returns `True` if the string value is less than the specified value (both must be same data type).	The specified string to check if the original string is less than.
`lessThanOrEqual`	Returns `True` if the string value is less than or equal to the specified value (both must be same data type).	The specified string to check if the original string is less than or equal to.
`matches`	Returns `True` if the string matches the specified Java regular expression. **Tip:** Regular expressions specify patterns to search for in string data using standardized syntax conventions. A regular expression, or *regex*, can specify complex patterns of character sequences. For example, the following regular expression: `a(b|c)d` searches for the pattern: `a`, followed by either `b` or `c`, then followed by `d`. This regular expression matches both `abd` and `acd`.	The specified Java regular expression to check if the original string matches. **Note:** Java regular expressions perform complete matches on strings, not partial matches. So, if you are searching for `Corporate` in an application name, an application with the name `Corporate Planning` would not be a match. Use wildcards before and after a string to search for partial matches. For example, `.*Corporate.*` will match with `Corporate Planning`.

Method or Attribute	Usage	Parameters
`orElse`	Returns a specified value if the preceding expression term has a null value.	The value to return if the preceding expression term is null. For example, `return node.properties.Core.Description.orElse('Default Descr')` will return "Default Descr" if the Core.Description property for a node is null.
`replace`	Replaces all instances of an old string with a new string	• The original string to search for **Note:** You can use a Java regular expression to identify the string pattern to search for. • The string to replace the original string with
`split`	Splits a string into a list of strings based on the specified delimiter. **Note:** See *List and String List* for the methods that you can add to a string list.	The delimiter to split the string on **Include Blanks**: Specify whether or not blank values should be included as list items in the resulting string list. • `True`: Allow blank values to be included in the resulting string list. • `False` (default): Blank values are excluded from the resulting string list. For example, for the string `A//C`, if **Include Blanks** is enabled, the string list will have three entries: `A, (blank value), C`. If it is disabled, the string list will have two entries: `A` and `C`.
`startsWith`	Returns `True` if the original string begins with the specified string. **Note:** The specified string is case-sensitive.	The specified string to check if the original string begins with.
`substring`	Returns part of a string.	• The starting position • The number of characters to return

Method or Attribute	Usage	Parameters
`subtract` (numeric string only)	Subtracts an integer value from the numeric string. **Note:** Zero padding on the numeric string is retained.	The integer value to subtract.
`toDate`	Converts a string value into a date value	The date format (for example, (MM/dd/yyyy). See Date Formatting Symbols for date and time formatting.
`toFloat`	Converts a string value into a float value **Note:** If the source string contains alphabetic characters, it is converted to zero.	No parameters required
`toInteger`	Converts a string value into an integer value. **Note:** If the source string contains alphabetic characters, it is converted to zero.	No parameters required
`toLowerCase` (applicable to alphabetic strings only)	Returns a string in all lowercase.	No parameters required

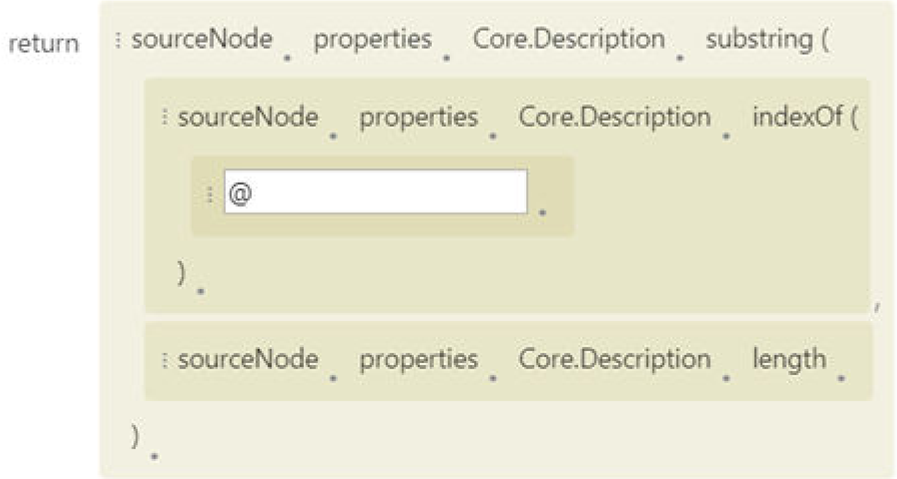
Method or Attribute	Usage	Parameters
`toNumericString` (applicable to string only)	Converts a string value into a numeric string value. **Note:** If the source string contains alphabetic characters, it is converted to zero.	`Pad Length`: Enter the total length of the numeric string to be zero padded up to, with 0 meaning no padding is added. For example, if you set the Padding to 10 and your numeric string value is 7 digits long, three zeroes are prepended to the string to get to 10 places.
`toUpperCase` (applicable to alphabetic strings only)	Returns a string in all uppercase.	No parameters required
`trim`	Returns a string with leading and trailing spaces removed. **Note:** `Trim` will also remove whitespace characters, such as Tabs and Carriage Returns.	No parameters required

For `greaterThan`, `greaterThanOrEqual`, `lessThan`, and `lessThanorEqual`, the string is sorted by the first character, then the second character, and so on. When comparing, 0-9 is less than A-Z.

Examples:

- A < AA
- AA > B
- 11 < 2
- A > 1

You can combine string methods and attributes in your expression. For example, the following string searches for the @ character in a node's description and returns everything after that character.



```
return  sourceNode . properties . Core.Description . substring (
            sourceNode . properties . Core.Description . indexOf (
                @ .
            ) .
        ,
            sourceNode . properties . Core.Description . length .
        ) .
```

In this example, the first expression term is a `substring` expression, which requires two parameters: a starting position, and the number of characters to return.

- For the starting position expression term, we selected an `indexOf` method with a parameter of "@". This will search the description field for the @ sign and return the integer value of where that sign is located as the starting position.
- For the number of characters to return expression term, we selected a `length` attribute. This ensures that however long the string is, the full value after the @ sign will be returned because the number of characters is equal to the length of the complete string.

Using Null and Blank Values in Expressions

You can use null and blank values in expressions, either to check for null and blank values in an If statement, to compare null and non-null values, or to return a null or blank value in a return statement.

Considerations

- When you transform a property, returning a null value results in the property for the target node not getting updated, while returning a blank value overrides the target node property with a blank value.
- When you derive a property, returning a null value results in a calculated value that is null, and returning a blank value results in a calculated value that is blank.
- When comparing two null values, the `equals` method returns **True**. When comparing a null and a non-null value, the `equals` method returns **False**.

You can insert null or blank values in string and integer fields in expressions. By default, when you insert a string or integer object into an expression, the value is `null`. If you want to insert a blank value, enter any text in the field and then clear the entered text. The string or integer field will display `blank` instead of `null`.

You can also insert a `null` object to enable null values to be returned for non-String data types (such as Integer or Float) as well as to allow for easy evaluation of null values in an expression.

To reset a blank value back to null, click ⋮ next to the field and select **string** or **integer** again.

Using Lookup Sets in Expressions

Lookup sets enable you to transform an input value to an output value in an expression. For example, you can create a lookup set that transforms the values "A", "L", and "O" in an account type property field to the values "Asset", "Liability", and "Owner's Equity".

See Working with Lookup Sets.

To use a lookup set in an expression, select the lookup set that you want to use, select **find**, and then specify the lookup value that you want to search for in the lookup set. By default, if the lookup value is not found, the search value is passed through and returned. You can disable the pass through to return a null value instead.

> **Tip:**
>
> Use **Disable Passthrough** with the **orElse** function to return a specified value when a lookup value is not found in a lookup set.
>
> For example, the expression `lookupSets.Account Type.find("A", true).orElse("Not Found")` will search the Account Type lookup set for a value of "A", and if it is not found it will return "Not Found".

1. From the palette, drag `lookupSet` to a return block in the editor.
2. Click the dot next to the lookup set, and select the lookup set that you want to use. Only those lookup sets that contain lookup values for the current application are displayed.
3. Click the dot next to the name, and select `find`.
4. In the **find** string, enter a literal value or specify a property for a node as the lookup value. For example, `sourceNode.properties.FCGL Account Type`.
5. **Optional**: In **Disable Passthrough**, select `True` to return a null value if the lookup value is not found.

Date Formatting Symbols

The following table describes the date format symbols that you can use when formatting dates in Oracle Fusion Cloud Enterprise Data Management, such as when defining extract options or using the `toDate` or `format` methods in expressions. Note that the fields are case-sensitive.

Symbol	Date or Time Component	Examples
G	Era designator	AD
y	Year	1996; 96
Y	Week year	2009; 09
M	Month in year	July; July; 07
w	Week in year	27
W	Week in month	2
D	Day in year	10
d	Day in month	2
E	Day name in week	Tuesday; Tue

Symbol	Date or Time Component	Examples
u	Day number of week (1 = Monday, 7 = Sunday)	1

Working with Literal Values

Literal values enable you to manually enter string, Boolean, integer, or float values in your expression. For example, you can use a literal value to evaluate if a node's account type is equal to "Asset".

To add a literal value to an expression term, drag the **string**, **Boolean**, **integer**, or **float** object from the palette to the editor, or click ⋮ and select one of those objects.

> **Note:**
>
> You can insert null or blank literal values in expressions. See Using Null and Blank Values in Expressions.

Copying and Pasting in Expressions

To facilitate building expressions, you can copy and paste individual expression terms within an expression, copy entire expressions across contexts, or capture an expression as text so that it can be shared with other parties outside of the expression builder (such as in email).

Copying and Pasting Expression Terms

You can copy and paste individual expression terms within a single expression. When you copy an expression term, that term is stored. Only one expression term can be stored at a time, but you can paste the expression term multiple times. The copied expression term is retained for the duration of the user session.

To copy an expression term, in the yellow box surrounding the expression term, click ⋮, and then select **Copy**.

To paste an expression term, in the yellow box surrounding the expression term that you want to replace, click ⋮, and then select **Paste**.

When you paste the expression term, the stored term replaces the selected expression term. You may have to insert a new If or Return statement in order to create an expression term that you can paste over.

Copying and Pasting Entire Expressions

You can copy an entire expression from one context and paste it to another. When you copy an expression, that expression is stored. Only one expression can be stored at a time, but you can paste the expression multiple times. The copied expression is retained for the duration of the user session.

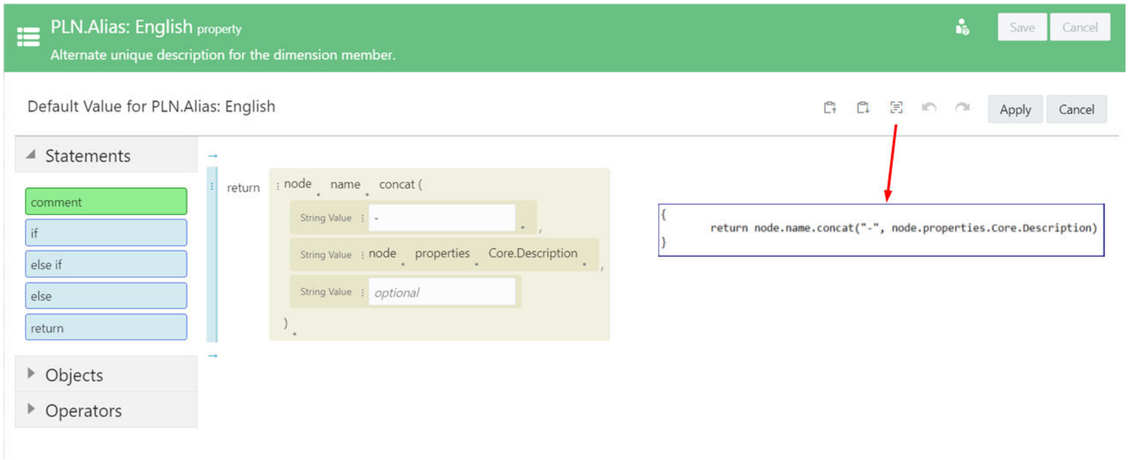
Considerations

- When you paste the expression, the copied expression replaces the entire existing expression for the current context.
- When you paste an expression in a different context, some of the objects in the copied expression may no longer be valid in the new context. The system converts invalid objects to valid objects as follows:
  - `SourceNode` from property transformations or subscription filters is converted to `Node` when pasting to derived properties, custom validations, or policy filters.
  - `Node` from derived properties, custom validations, or policy filters is converted to `SourceNode` when pasting to property transformations or subscription filters.
- You cannot copy an expression from one environment (for example, *Test*) and paste it into another (such as *Production*).
- Some properties and lookup sets in the source expression may no longer be valid in the pasted context.
- Invalid expression terms in the pasted expression are truncated to the valid portion. For example, `node.properties.Custom.SourceProp.concat(-)` is truncated to `node.properties`.

In the expression builder, click to copy an expression, and then click to paste it.

Capturing Expressions as Text

In the expression builder, click to copy a text version of the expression to your clipboard in order to share it with other parties via email, service requests, instant messages, etc.



```
{
        return node.name.concat("-", node.properties.Core.Description)
}
```

Saving and Loading Expressions from a File

You can save or load expressions from a `JSON` file in order to migrate expressions from one environment to another.

Saving an Expression to a File

Click **Save to File** to save the current expression in the expression builder to a `JSON` file.

The expression file is downloaded to the browser default download location. The file name is the type of object that you are exporting (for example, `Derived Property`, `Policy Filter`, `Subscription`), plus the name of the object being edited.

For example, if you are saving the "Approval - Corp Plan" policy filter, the file name is:

`Policy_Filter_Approval_-_Corp_Plan.json`.

After you save an expression to a file, you can review it in a `json` viewer.

> **Caution:**
>
> Manually creating or editing the expression `json` file is not supported.

Loading an Expression from a File

1. In the expression builder, click **Load from File** ⭱.
2. In the **File Open** dialog, select the `JSON` file you want to load, and then click **Open**.

   > **Note:**
   >
   > If you already have an expression open in the expression builder, the existing expression is overridden. Any unknown object references or syntax errors are omitted when the expression is loaded. Incomplete expression terms are displayed with a red outline, indicating an error.

3. **Optional**: Click **Cancel** to cancel the load and retain the current expression.

Expressions: Detailed Example

This detailed example further illustrates how to use the expression builder to build custom business logic.

See these topics for more information about the expression builder:

- Expression Builder Overview
- Building Expressions

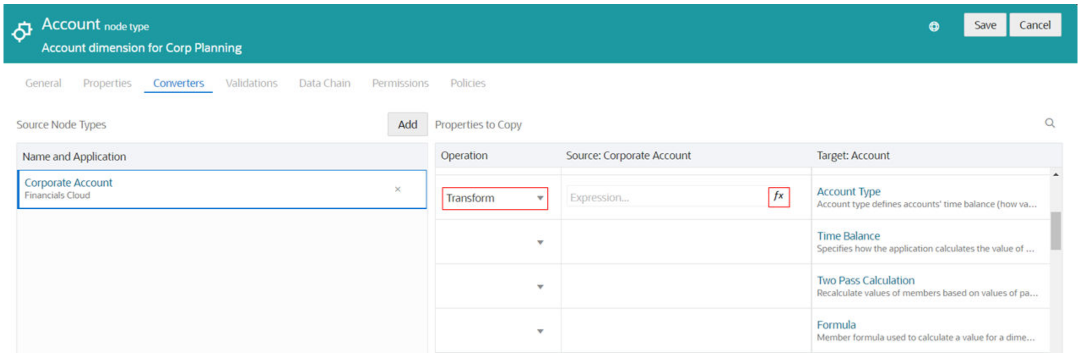
In this example, we want to be able to drag nodes from an Account dimension in an Oracle Financials Cloud General Ledger application to an Account dimension in a Corporate Planning application. In order to do that, we have set up a node type converter between the node types in each application. (For more information about converting node types, see Working with Node Type Converters). We will build the following expression to transform the Account Type property in the Planning application:

If the account type in the Oracle Financials Cloud General Ledger application is "Asset", "Liability", or "Owner's Equity", we want the account type property in the Corporate Planning application to be "Expense". If it is any other kind of account type, we want the Corporate Planning account type property to be "Non-Expense".

First, let's look at the properties in the node type converter that we set up:

1. From Node Types, click the Account node type for Corporate Planning to open the node type in the inspector.

2. Click the Converters tab.
3. In the Properties to Copy panel, locate the **Account Type** property in the Target column.
4. Click **Edit**.
5. In the Operation column, select `Transform` from the drop down menu.
6. In the Source column, click **Define Expression** *fx* .

Name and Application
Corporate Account Financials Cloud

Operation	Source: Corporate Account	Target: Account
Transform	Expression...	Account Type Account type defines accounts' time balance (how va...
		Time Balance Specifies how the application calculates the value of ...
		Two Pass Calculation Recalculate values of members based on values of pa...
		Formula Member formula used to calculate a value for a dime...

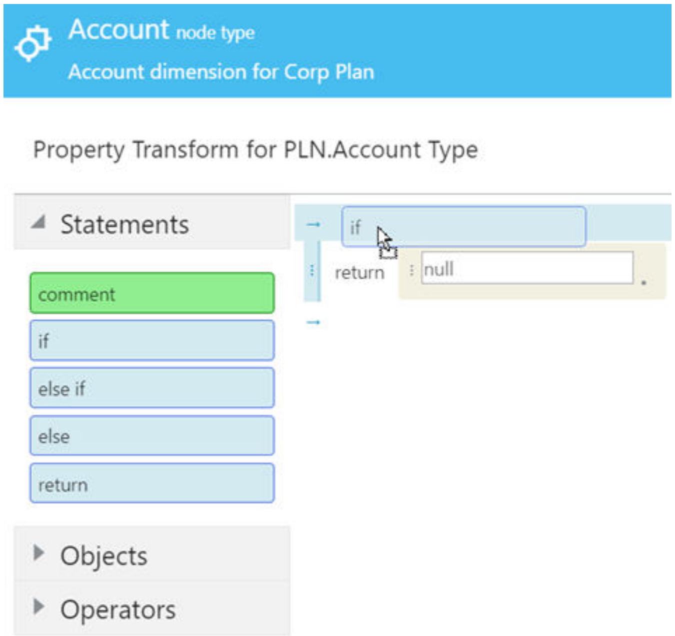
   .
   The expression builder is displayed.

Next we will build the expression to transform the account type.

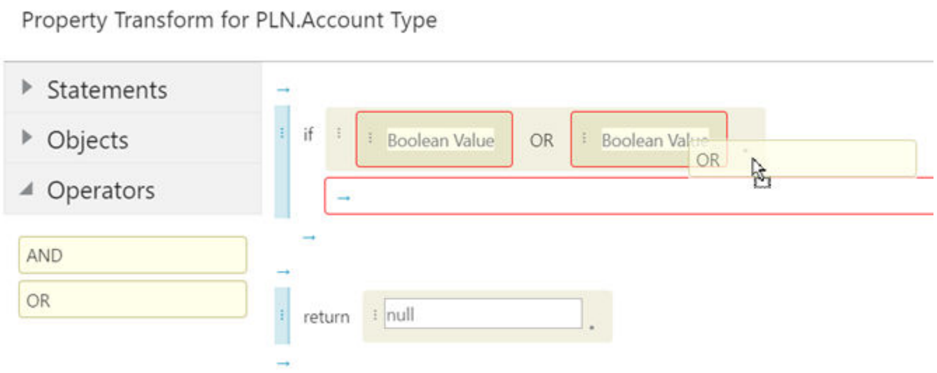
Building the Expression

To build the transformation expression:

1. From the palette, in Statements, drag the **IF** statement to the blue arrow at the top of the editor.

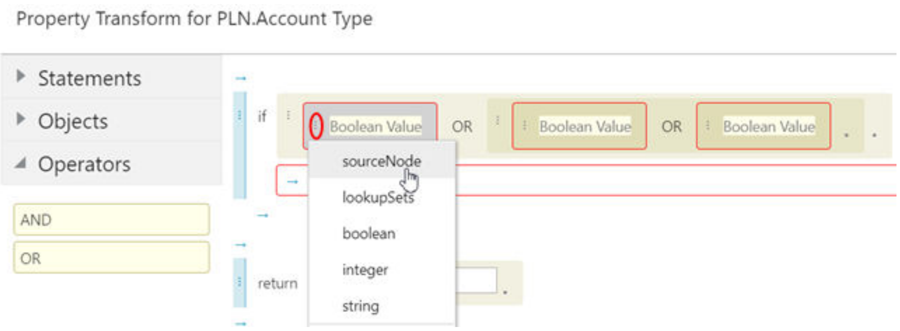
   

2. From Operators, drag two **OR** operators to the IF condition in the editor.

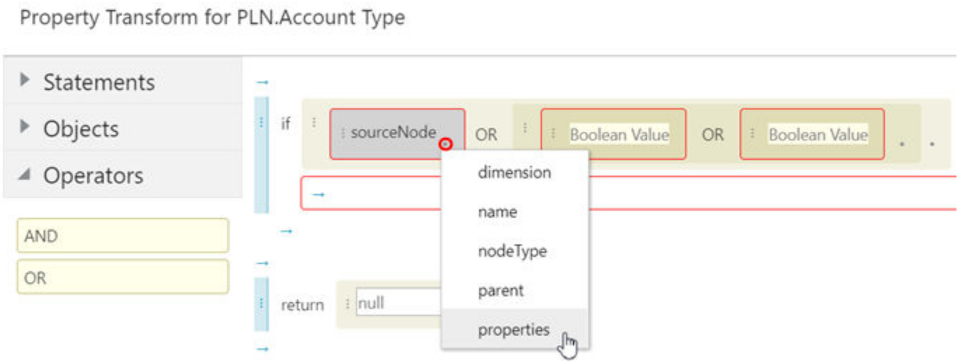


Notice that each of the IF statements and the statement block have red outlines around them. This means that they are not valid. Each IF statement has a condition that is expected to be of type Boolean, and each statement block is expected to contain an return statement. The expression builder evaluates the expression in real-time as you build it, and these validation errors will be resolved as we continue to add fields to our expression.

3. In the first IF condition, click ⋮, and then select `sourceNode` from the drop down menu.



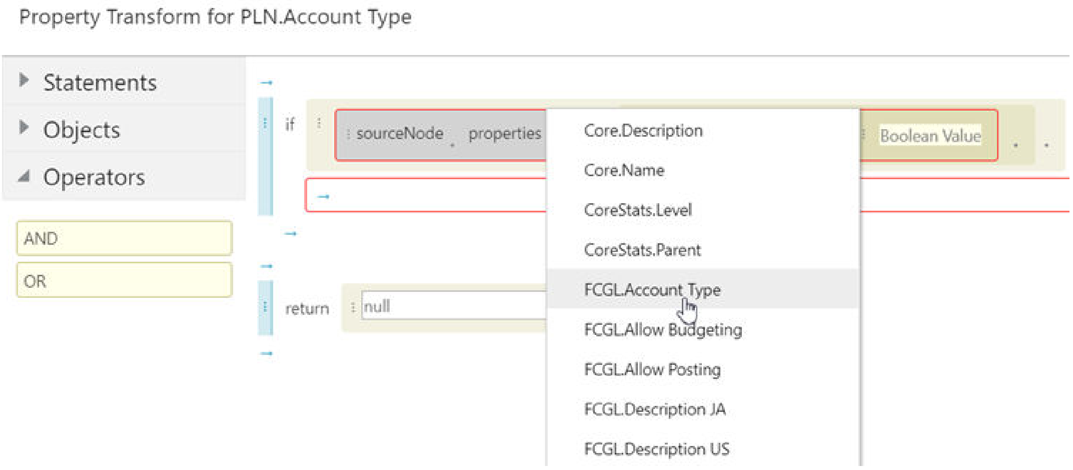
4. Click the dot next to **sourceNode**, and select `properties` from the drop down menu.



Note:

When you click the dot next to a field in an expression, the drop down menu displays only the objects, methods, and attributes that are applicable to that field.

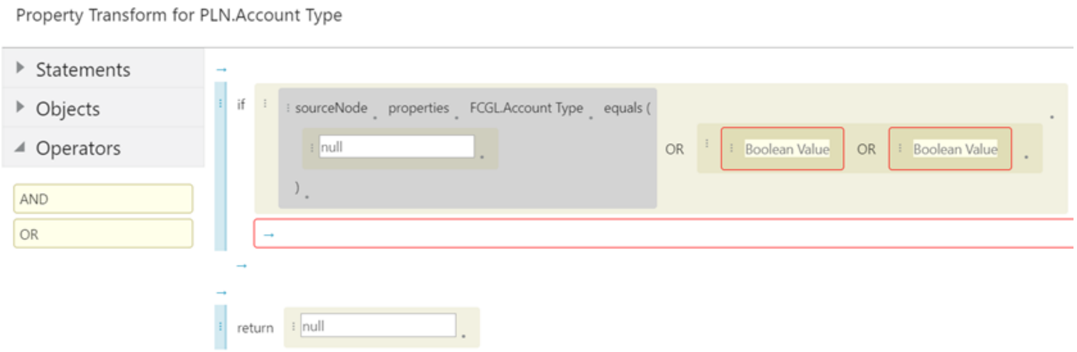
5. Click the dot next to **properties**, and select `FCGL.Account Type` from the drop down menu.



Note:

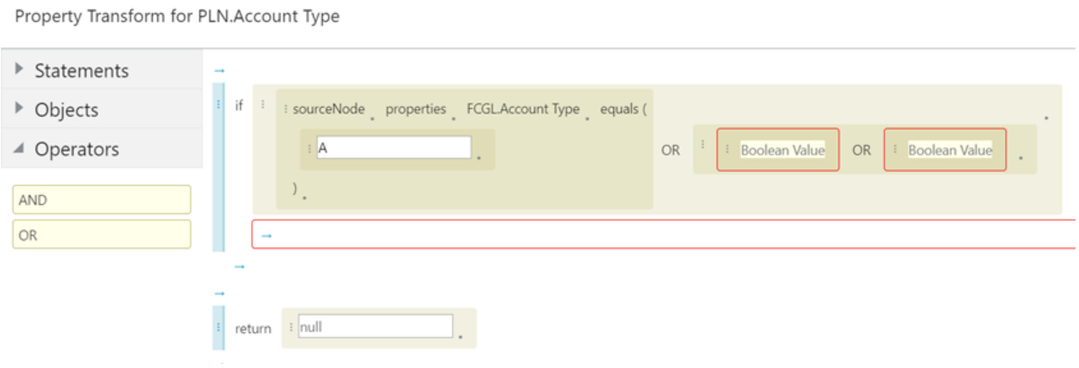
When displaying properties for property transformations, because the node type of the source node is known, only the properties for that node type are displayed. However, when displaying properties for derived properties, because the node type is not known, all of the properties that are valid for the application are displayed.

6. Select the dot next to **FCGL.Account Type**, and select `equals` from the drop down menu.

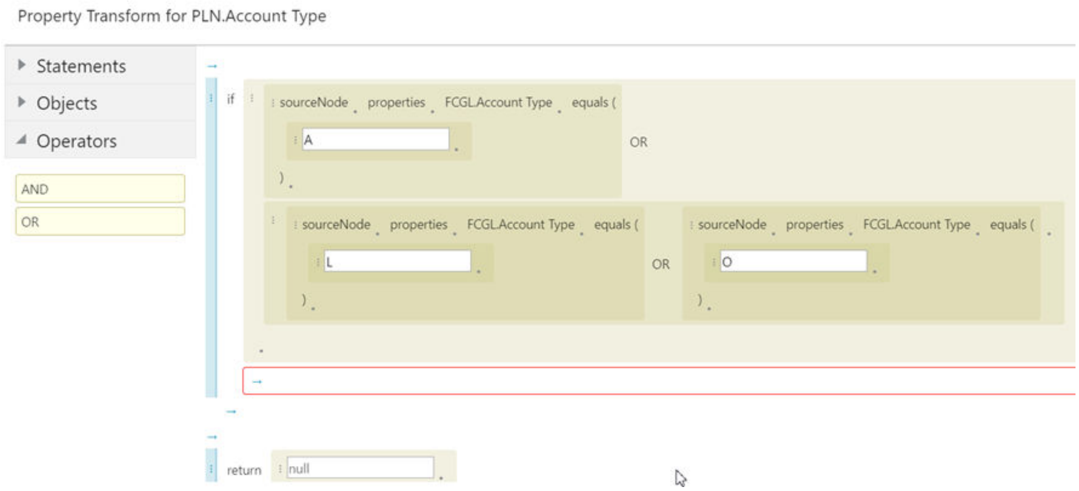


Notice that the red validation border is no longer displayed on the first IF condition. This is because by adding an equals statement, the condition now contains the required Boolean value.

7. In the literal text field, enter "A" (for Asset).

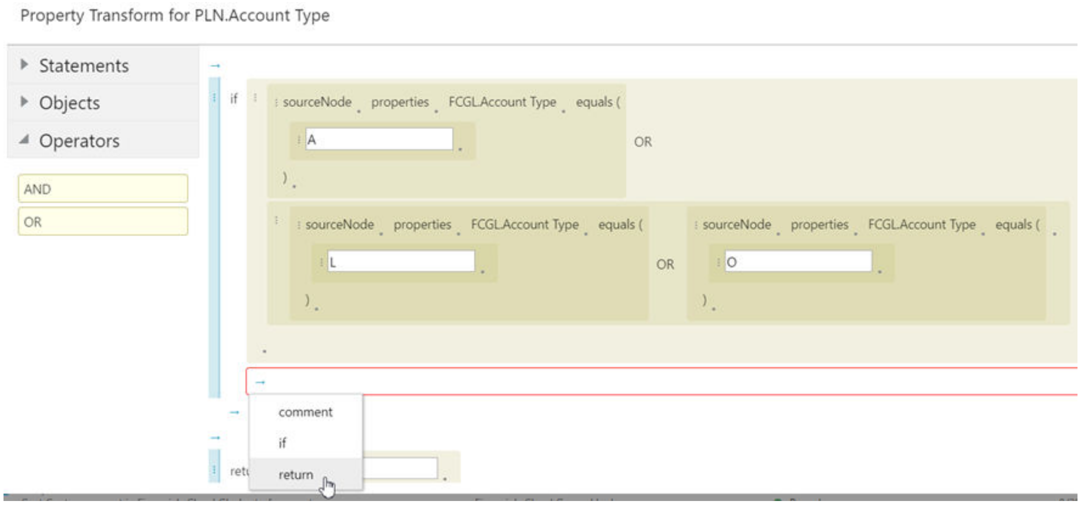


8. Repeat steps **3-7** to add IF conditions for FCGL.Account Type equals "L" (for Liability) and "O" (for Owner's Equity).



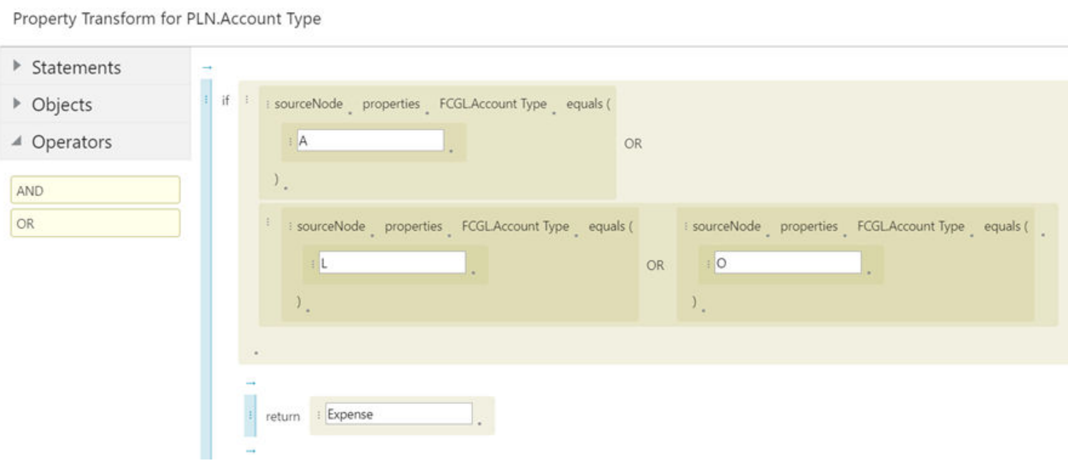
All of the IF conditions now contain Boolean values and therefore no longer display red borders indicating validation issues. However, the statement block is still missing a return statement. Thus, it still has a red border indicating a validation error.

9. In the statement block, click → and then select `return` from the drop down menu.

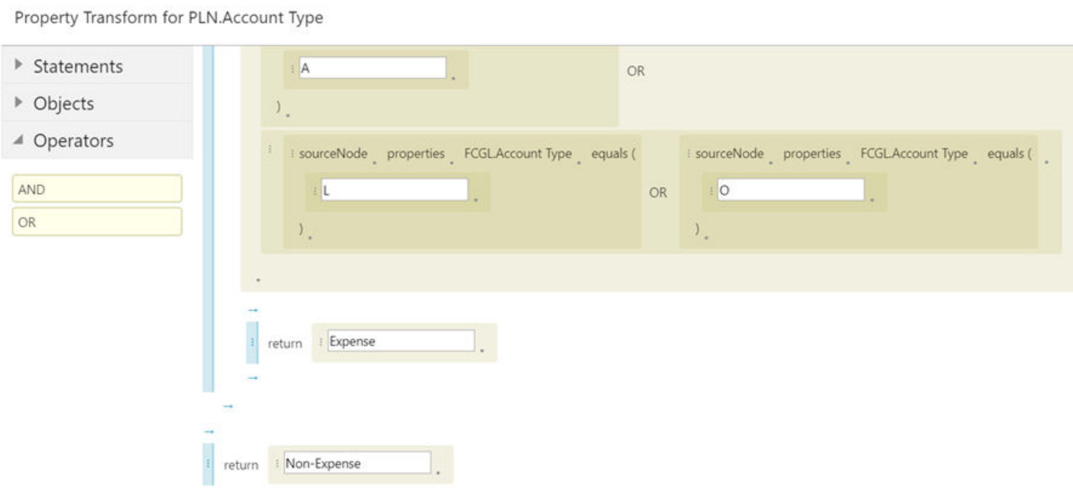


Notice that after you add the return statement, the red validation border is no longer displayed around the statement block.

10. In the return block, enter "Expense".



11. In the lower return block, enter "Non-Expense".



Note:

You do not have to manually enter an Else If statement after the first IF statement. If an IF statement is not evaluated as True, the logic continues to the next statement automatically.

12. Click **Apply**.
The expression builder is closed and the node type inspector is displayed.

Testing Expressions

From the expression builder, you can test your expression to ensure that the expression logic works as expected.

When testing an expression, you select a viewpoint and a node to evaluate the current expression. For expressions in custom validations, policy filters, or subscription filters, you can also select a request in the Submit, Approve, or Commit stage for the view to evaluate the current expression with.

After you select a node, click the **Evaluate** button to return the value of the selected node as calculated by the current expression logic.

To test an expression:

1. From the expression builder, click **Test Expression** .
   The Test Expression drawer is displayed.
2. In **Viewpoint**, select the viewpoint that contains the node to evaluate. You can select any viewpoint that you have at least *Participant (Read)* access to that is either in the selected application or that uses the same node type for the property.
3. In **Node** click the **Node Selector** and select the node to be evaluated.
   The node's properties are displayed. Use the Search bar to search for specific properties.

   > **Tip:**
   >
   > When testing expressions for transforming node parents in a node type converter, make sure you are evaluating the expression on the parent node and not the child node.

4. **Optional**: For expressions in custom validations, policy filters, or subscription filters, click the **Request** drop down menu and select a request to evaluate the expression with. You can select from requests for the current view that you have access to that are in the Submit, Approve, or Commit stage.
5. Click **Evaluate** to evaluate the current expression logic for the selected node.
   The calculated result is displayed next to the **Evaluate** button.

Frequently Asked Questions about Expressions

How can I improve the performance of my expressions when they contain a large number of sibling nodes?

Expressions with logic around siblings like `Node.Siblings.Any(SiblingNode.Properties.Boolean Prop)` can cause performance issues in hierarchies where there are a lot of siblings. This is because each node has a unique set of siblings. For example, for a top node with children A, B, C, D, and E, node A has siblings B,C,D,E, node B has siblings A,C,D,E, etc.

A better way to construct the expression is to create a property that you will reference on the parent using the children and then referencing that property on the node. For example, you can create a property called **ChildrenBooleanSet** with expression `Node.Children.Any(ChildNode.Properties.Boolean Prop)`, and then reference that property on the node: `Node.Parent.Properties.ChildrenBooleanSet`.

This type of expression results in better performance because the parent value can be cached.

Caution:

Be aware that with the example expression above, the children will include the node itself and the siblings do not. You can add extra logic to filter out the node itself if needed.

26 Understanding Data Sources

Data sources represent logical source systems for nodes.

They are used in the following ways in Oracle Fusion Cloud Enterprise Data Management:

- They establish links between nodes. When you bring a node in (through interactive requests, request file uploads, subscription requests, viewpoint loads, or merge imports), if the incoming node is updating an existing node the data source on the incoming node establishes a link between those nodes. See About Node Links.
- They identify which matching rules to use when matching request items. When you create request items that you want to match on, the data sources that you assign to them determine the matching rules that are used for the match. See Working with Matching and Deduplication.
- They identify the survivorship rule to use when merging properties and relationships for accepted matches. See Creating, Editing, and Deleting Survivorship Rules.

There are two types of data sources:

- **Registered data sources** represent other Cloud EDM applications. They are created and deleted automatically when you register or delete an application in Cloud EDM. They cannot be created or deleted manually.
- **Unregistered data sources** are manually created (see Creating, Editing, and Deleting Data Sources) to represent other external systems and files that you want to bring data in from.

Specifying Data Sources for Request Items

There are a number of ways to specify a data source for request items:

- Request items that are generated from other Cloud EDM applications (for example, from viewpoint compare requests or subscriptions) are automatically assigned the registered data source for that application.
- For request file uploads, use the Data Source column to identify the data source for each request item in the file. The data sources that you specify are assigned to the nodes when the request file is uploaded. See Request Load File Format.
- You can specify a default data source for a defined viewpoint load. During viewpoint load, each request item created in the load request is populated with the name of the data source that you specified. See Creating, Copying, and Deleting Loads

Creating, Editing, and Deleting Data Sources

Considerations

- You must have *Owner* or *Metadata Manager* permission on at least one application in order to create, modify, or delete an unregistered data source.
- You must have *Owner* or *Metadata Manager* permission on an application in order to edit the registered data source for that application. You can enable and disable the data

source, and you can add a Code to identify the data source instead of the name when a request file is processed. No other edits can be made to registered data sources.

- You can create and delete unregistered data sources only. You cannot create or delete a registered data source.

Creating a Data Source

Considerations

1. From the home page, click **Tools**, and then click **Data Sources**.
2. Click **Create**.
3. Enter the following information:

Field	Description
Name	Name (must be unique) for the data source.
Description	**Optional**: Description for the data source
Code	Unique code that identifies this data source when a request file is processed. **Note:** If the Code field is left blank, the name of the data source is used to identify it when a request file is processed. However, because the data source name can change over time it is a best practice to use the Code field to identify the data source instead of the name.

4. Click **Create**.
   The data source is displayed in the inspector.

Editing a Data Source

After you create a data source, you can edit it in the inspector to enable or disable it or to change any general details.

1. From **Data Sources**, click the name of a data source to open it in the inspector, and then click **Edit**.
2. **Optional**: Edit any of the summary fields for the data source listed in Creating a Data Source.

> **Note:**
>
> For registered data sources, the only field that you can update is the Code field.

3. In **Enabled**, click to enable or disable the data source. Data sources must be enabled in order to reference them in matching rules.

> **Note:**
>
> You cannot disable a data source if it is being referenced by a matching rule.

Deleting a Data Source

To delete a data source, in the Action column for the data source that you want to delete, click **Actions** ⋮, and then select **Delete**.

> **Note:**
>
> You cannot delete a data source if it meets any of the following conditions:
>
> - It is referenced by a matching or a survivorship rule. See Creating, Editing, and Deleting Matching Rules and Creating, Editing, and Deleting Survivorship Rules.
> - Node link history exists for the data source. See About Node Links.

27

Working with Matching and Deduplication

Matching and deduplication refers to the process of comparing nodes in various contexts, identifying nodes that are the same, and then merging in the results. This helps you prevent duplication of data in your system.

Oracle Fusion Cloud Enterprise Data Management provides two mechanisms for preventing duplicate data:

- **Matching and Merging Request Items**: Prevent duplication before it occurs by matching incoming nodes in a request to existing nodes in a viewpoint in order to identify and merge nodes that are the same. See Understanding Matching and Merging Request Items.
- **Deduplication**: Correct duplication in your system after it occurs by matching nodes that already exist in a viewpoint in order to identify and merge nodes that are the same. See Understanding Deduplication.

For both mechanisms, you create matching rules to specify how nodes are matched with other nodes based on their property values and survivorship rules to specify how the properties and relationships from the nodes are merged.

Terminology

The following terms can help you understand the matching process:

- **Data source**: An object that represents the source for the incoming data to be matched and linked in Cloud EDM. This can be either another Cloud EDM application (called a *registered* data source) or an external system whose data is not being managed in Cloud EDM (called an *unregistered* data source). See Understanding Data Sources.

  > **Note:**
  >
  > You can match request items for any data source. You can deduplicate data for registered data sources only.

- **Matching rule**: Controls how nodes are matched either from an incoming data source to nodes that already exist in a node type (for matching and merging) or in a viewpoint (for deduplication). See Creating, Editing, and Deleting Matching Rules.
- **Survivorship rule**: Specifies which properties and relationships from the source node get merged into the target nodes in a viewpoint after a match has been confirmed. See Creating, Editing, and Deleting Survivorship Rules.
- **Matching workbench**: Enables you to review match candidates based on the criteria from the matching rules and accept the ones that you want to merge into the existing nodes. See Matching and Deduplicating.
- **Clustering property** (Deduplication only): A property that you identify to group nodes into clusters so that you can run matching on them in order to identify and combine duplicate nodes.

Understanding Matching and Merging Request Items

Matching and merging request items refers to the process of comparing incoming nodes to existing nodes and identifying nodes that are the same, and then merging in the results.

When you confirm that a record from a particular data source is a match to an existing node in specific node type, the system stores information about that match so that the next time you bring data in from that data source, the incoming node is automatically matched to the existing node

Use Cases

There are two main uses cases for matching and merging request items:

Matching and merging multiple records with different identifiers from the same data source.

For example, suppose you have an existing entity node with the name **Oracle Incorporated**, and you are merging in a file with a node name **Oracle Inc.** If you were to submit the request without matching, the Oracle Inc. node would be added as a new node because there isn't an exact match on the existing node name. But by setting up matching rules and reviewing the match results, you can specify that the incoming node is a match for the existing node. That way, when the request is processed instead of adding a new node the incoming node is merged with the existing node.

Matching and merging multiple records with different identifiers across multiple data sources.

For example, let's say you have a customer record, and you have multiple data sources that each contain different information about that customer. You might get credit card information for that customer from a point of sale device, and biographical information such as a name and email address from an online ordering system. If you were to submit a request to merge the records from these two data sources together, you would get two different customer records, each with different information about that customer.

With matching, you can set up a match rule that looks at, for example, the credit card number as a way to match the two records. Then, when you merge the data sources together, you can use the matching workbench to confirm that the two records are a match, and thus create a master record for that customer with all of the information from both data sources.

Process Overview

Configuring and using matching and merging request items follows this general process:

1. A metadata manager or a dimension owner performs the initial setup of the matching process:
   - Creates the data sources if you are bringing in data from a source other than an Oracle Fusion Cloud Enterprise Data Management application.
   - Defines the matching rules that determine how nodes from other data sources are matched with existing nodes based on their property values.
   - Defines the survivorship rules that specify how the properties and relationships from the incoming nodes from unregistered data sources are merged with the existing nodes.
2. A business user creates and submits a request with nodes to merge into existing nodes. The request items can be manually created or loaded from a file.

> **Note:**
>
> The matching process compares incoming nodes to existing nodes in a node type. Therefore, matching takes place in the context of a request.

3. A metadata manager runs matching against the incoming nodes, reviews the matching candidates, and accepts or rejects the matches. See Matching and Merging Request Items.
4. The metadata manager applies the match results to the incoming request items. For those nodes that are accepted as a match, the action in the request item is changed from an Add to an Update, Insert, or Move on the existing target nodes. See Reviewing Match Results and Applying Changes.
   Accepted match results are saved for each data source so that the next time request items are brought in from that data source they are matched automatically.

Understanding Deduplication

Deduplication enables you to use the matching process to identify nodes in a viewpoint that are duplicates of each other and combine them into a single node.

Deduplication is run on nodes that already exist in a viewpoint, unlike Matching and Merging Request Items, which works on incoming nodes being added in a request. This lets you find and merge duplicate existing nodes that may have been added to the viewpoint before matching was available, or outside of the request process (such as by an import or load).

Deduplication uses many of the same elements as matching and merging request items:

- Matching rules are used to identify potential duplicate nodes.
- Survivorship rules control how properties and relationships are merged after a match is confirmed.
- Use the matching workbench to accept, reject, or skip the match candidates.

Deduplication Modes

You can deduplicate nodes in a viewpoint in two modes:

- **Cluster key**: Define a clustering property for the node types of the nodes to be matched, and then perform matching for each cluster. See Deduplicating Using a Cluster Key.
- **Time-based**: Deduplicate nodes in a viewpoint based on the date that they were created. See Time-Based Deduplication

You can run only one mode of deduplication for a specific viewpoint and node type in a single request, but you can use both modes to deduplicate nodes in a viewpoint in different contexts. For example, you could initially deduplicate nodes in a viewpoint by cluster key and then any nodes created after that could be deduplicated incrementally using time-based deduplication.

> **Note:**
>
> You can deduplicate nodes of a particular node type in only one active request at a time, regardless of the mode.

The cluster key and the node creation date for time-based deduplication work to essentially limit the scope of the deduplication operation. Unlike matching and merging, which is automatically constrained by the maximum limit of request items in a request, viewpoints could potentially contain millions of nodes. Specifying either a node creation date or a clustering property lets you target the specific nodes that you want to deduplicate in a single operation.

> **Note:**
>
> Both cluster key and time-based deduplication require that the `CoreStats.Created Date` property is included in the node type being deduplicated in order for the system to be able to track the progress of which nodes have been evaluated and which have not.

Deduplicating Using a Cluster Key

In order to deduplicate nodes using a cluster key, you must define a clustering property for the node types of the nodes to be matched. This filters the list of nodes in the viewpoint to be matched to other nodes in the same viewpoint. When you run the deduplication process, you specify the value of the clustering property that you want to deduplicate nodes for.

> **Tip:**
>
> When you define a clustering property for a node type, the property that you select must have an Allowed Values list for that node type (see Configuring a Clustering Property for a Node Type). Then, when you run deduplication using a cluster key you select the clustering property from that list of allowed values. For example, if you are deduplicating customers and the clustering property is State, you could select Texas as the clustering value to deduplicate customers in the state of Texas.

The cluster key is applied to the set of nodes that you are matching, not the nodes that are being matched against. So, in the above example where you are matching customers in the state of Texas, a match with the same name in California would be displayed.

Time-Based Deduplication

Time-based deduplication enables you to deduplicate nodes that were created on or after a specified date. It does not require you to specify a clustering property. Instead, when you create a match to deduplicate you specify a node creation start date and, optionally, a batch size.

Creating, Editing, and Deleting Matching Rules

Matching rules control how nodes that are named differently are matched when matching request items or deduplicating a viewpoint. Instead of an exact match on the node name, you can perform matches based on other properties and match types.

Matching rules are created at the node type level. You can define multiple rules for a node type, and you can specify the order in which the rules are evaluated. You can also specify thresholds to automatically accept or exclude match results based on their confidence match.

Matching Rules Overview

When matching request items for new nodes in a request or when matching existing nodes in a viewpoint, by default nodes are considered to be the same node when there is an exact match name on the Name property (including any node type qualifiers and alternate names). Matching rules enable you to specify other properties and match types to determine whether nodes are the same.

To illustrate the matching concept, let's look at an example of matching request items. Suppose you had a viewpoint with a node named "Oracle Corporation", and you upload a request file that contains a node with the name "Oracle Inc". If you were to apply the changes from this request file without matching, you would end up with two different records (because there isn't an exact match on the name).

Instead of the exact name match, you could set up a matching rule that looks for names that are similar to one another. Then, when you run matching on the request items, the system presents the incoming "Oracle Inc" node as a potential match to the existing "Oracle Corporation" node. When the match is accepted, the incoming node is merged into the existing node, and you have a single record for Oracle with the merged information from both nodes.

You could also configure the matching rule to match on properties other than Name. Suppose both records had a property called **Symbol**, with the value "ORCL". You could set up a matching rule to match on Symbol instead of Name in order to match the incoming node to the existing one.

Best Practices

While you can enable up to three match rules at a time, each rule requires additional processing time when running the matches. Therefore, it is a best practice to enable only the minimum number of matching rules that meet your business needs.

Considerations

- You must have *Owner* or *Metadata Manager* permission on the application or dimension that contains the node type in order to create, edit, or delete matching rules.
- If you enable multiple matching rules, matching is performed using OR logic. That is, if a node meets the criteria of one rule or the other rule, it is presented as a potential match.
- If you add multiple criteria to a single matching rule, matching is performed using AND logic. That is, in order for the nodes to be considered a match for a rule, all of the criteria that you specify must be true.
- Matching rules for registered data sources can be used for either matching request items or deduplication. Matching rules for unregistered data sources can be used for matching request items only.

Creating Matching Rules

1. Inspect the node type that you want to create a matching rule for. See Inspecting a Node Type.
2. Navigate to the **Rules** tab, and on the **Matching** sub-tab click **Create**.
3. Enter a name and, optionally, a description for the matching rule.
4. In the **Data Sources** drop down menu, select the data source that will use this matching rule, and then click **Create**.
   The matching rule is displayed in the inspector.

Editing Matching Rules

After creating a matching rule, you can edit it to change the name or description, enable it, change the rule order, set thresholds to automatically accept or exclude a match result, or edit the definition parameters. You cannot edit the data source for a matching rule after it has been created.

1. Inspect the matching rule that you want to edit.
2. On the **General** tab, click **Edit** and perform an action:
   - **(Optional)** Edit the name or description.
   - In **Rule Order**, use the Up and Down arrows or enter an integer to specify the order in which this matching rule should be run for this data source and node type.

     > **Tip:**
     >
     > It can be helpful to enter larger values for the rule order so that if you create new rules that you want to place before your existing rules you do not have to manually reorder all of your rules.

   - Use the **Enabled** checkbox to enable or disable the matching rule.
   - In **Usage**, specify whether the rule will be used for **Matching** (see Understanding Matching and Merging Request Items) **Deduplication** (see Understanding Deduplication), or both.

     > **Note:**
     >
     > You can specify a Usage for matching rules for registered data sources only. Unregistered data sources can only be used with Matching rules, not Deduplication rules.

   - **(Optional)** In **Auto Accept Threshold**, specify a match score that must be met or exceeded in order for a match result to automatically be accepted when matching is run. If a matching rule has a match score that meets or exceeds this threshold, it is displayed as Accepted in the match results screen and other matching candidates are displayed as Rejected. If multiple matching rules meet or exceed the threshold, the match result with the highest match score is accepted. If the match scores are the same for multiple rules, then the rule with the highest priority (that is, the lowest **Rule Order**) is accepted.

     > **Note:**
     >
     > You can specify an Auto Accept Threshold for Matching rules only. You cannot specify an Auto Accept Threshold for Deduplication rules.

   - **(Optional)** In **Auto Exclude Threshold**, specify a match score that must be equal to or below in order for a match result to automatically be excluded when matching is run. Match candidates where the matching rule has a score below this threshold are not displayed in the match results screen.
3. On the **Definition** tab, click **Edit**, and then click the **+** sign to specify rule criteria:

- Use the **Property** drop down to select the property for the match. You can select from node-level properties with indexed values only.
- Use the **Match Type** and **Match Option** drop downs to select the operator (and option, as needed) for the match. The match type values depend on the data type of the property that you select, and the match option values depend on the match type, as follows:

Property Data Type	Operators and Match Options
String, Memo, Numeric String, Sequence	– Equals – Contains – Similar To: Performs a fuzzy search on the property value. In **Match Options**, optionally specify a **Prefix Length** that specifies the number of characters at the beginning of the string that must exactly match. The prefix length can be used to reduce the query results to only those that have the same characters at the beginning of the string.
Integer, Float	– Equals – Between In **Match Options**, specify an offset to indicate the upper and lower boundaries of the range of values to match on. For example, if you are matching with an offset of 5 on a target property that has value of 10, then any incoming source property value from 5 to 15 will be considered a match. – Between % In **Match Options**, specify an offset percentage that the property value must be within of the source node's property value. For example, if you are matching with an offset of percentage of 15% on a target property that has value of 5000, then any incoming source property value from 4250 to 5750 will be considered a match. – Greater Than – Less Than

Property Data Type	Operators and Match Options
Date, Timestamp	– Equals – Between In **Match Options**, specify an offset (in days) to indicate the upper and lower boundaries of the date range to match on. For example, if you are matching with an offset in days of 3 on a target property that has a date of March 10, then any incoming source property value from March 7th to March 13th is considered a match. – Before – After
Node data type	Equals **Tip:** This identifies match candidates with a Node data type property that has the same property value as the source node.

- When you have finished entering the first matching criteria, click **+** to enter a second criteria, or click **Save** to save the definition for the matching rule.
- To remove a rule criteria, in the Action column click the **Action** menu ⋮, and then click **Remove**.

4. Click **Save** to save the criteria that you entered.

Deleting a Matching Rule

1. Inspect the node type that contains the matching rule that you want to delete.
2. From the **Rules** tab, select the **Matching** sub-tab, and then click **Edit**.
3. In the Action column click the **Action** menu ⋮ for the rule that you want to delete, and then click **Delete**.
4. Click **Yes** to confirm that you want to delete the matching rule.

> **Note:**
>
> You cannot delete a matching rule that has any matching history. If the matching rule is no longer needed, you can disable it instead.

Creating, Editing, and Deleting Survivorship Rules

Survivorship rules determine which properties and relationships get merged from an accepted match candidate into a matching node in a node type during a match and merge or deduplication operation.

Survivorship Rules Overview

While matching rules determine which nodes might match other existing nodes, survivorship rules specify which properties and relationships from the source node get merged into the target nodes in a viewpoint after the match has been confirmed.

The source and target nodes are determined by the type of match operation being run:

- When matching and merging request items, the source nodes are the incoming nodes in the request, and the target nodes are the existing nodes in that node type. See Understanding Matching and Merging Request Items.
- When deduplicating nodes in a viewpoint, the source nodes are the nodes from the registered data source that you are evaluating, and the target nodes are the nodes that you are matching against and that you will be merging into during the merge operation after you accept a match . See Reviewing Deduplication Results and Applying Changes.

When you create a survivorship rule, for each property in the node type for which you are creating the rule you can choose one of these options:

- **None** (default): The property value in the source node is ignored during a merge.
- **Copy**: The property value in the source node is copied to the target node during a merge.
- **Transform**: The property value in the source node is transformed using an expression that you enter and then copied to the target node during a merge.

> **Note:**
>
> The Transform operation is supported for unregistered data sources only. For registered data sources, use the node type converter on the target node to configure how property values are transformed from the source and copied to the target. See Working with Node Type Converters.

> **Tip:**
>
> Transforming property values can be helpful in these use cases:
>
> - You have a null value for a property in the source node that you want to replace with a non-null value.
> - You want to perform a lookup of a source value to a target value using a lookup set.
> - You want to populate additional properties that are not directly loaded from the data source.

Considerations

You must have *Owner* or *Metadata Manager* permission on the application or dimension that contains the node type in order to create, edit, or delete survivorship rules.

Creating Survivorship Rules

1. Inspect the node type that you want to create a survivorship rule for. See Inspecting a Node Type.
2. Navigate to the **Rules** tab, and on the **Survivorship** sub-tab click **Create**.
3. Enter a name and, optionally, a description for the survivorship rule.
4. In the **Data Sources** drop down menu, select the data source that will use this survivorship rule, and then click **Create**. The data source that you select must have at least one matching rule enabled for the node type in order to use it for a survivorship rule.

   > **Note:**
   >
   > To use survivorship rules when deduplicating nodes in a viewpoint, create the survivorship rule for the registered data source of the application that contains the viewpoint that you are deduplicating.

Editing Survivorship Rules

After creating a survivorship rule, you can edit it to change the name or description, enable it, or edit the properties to merge after a match is accepted. You cannot edit the data source for a survivorship rule after it has been created.

1. Inspect the survivorship rule that you want to edit.
2. On the **General** tab, click **Edit** and perform an action:
   - **(Optional)** Edit the name or description.
   - Use the **Enabled** checkbox to enable or disable the survivorship rule.

     > **Note:**
     >
     > Only one survivorship rule for a data source can be enabled for a particular node type.

3. On the **Definition** tab, for each property in the node type, select an option:
   - **None**: Ignores the property value for this property during the merge.
   - **Copy**: Merges the property value from the source node to the target node during a merge.

     > **Note:**
     >
     > If the property that you are copying is not editable for this node type, a **Caution** ⚠ icon is displayed.

- **Transform**: Transforms the property value from the source node and then merges it to the target node during a merge.

  After you select **Transform**, click **Edit Expression** *fx* in the Transform column to open the expression editor and create an expression. Then, click **Apply** to return to the Definition tab. For help on transforming properties, see Property Transformations.

4. Click **Save**.

Deleting Survivorship Rules

1. Inspect the node type that contains the survivorship rule that you want to delete.
2. From the **Rules** tab, select the **Survivorship** sub-tab, and then click **Edit**.
3. In the Action column click the **Action** menu ⋮ for the rule that you want to delete, and then click **Delete**.
4. Click **Yes** to confirm that you want to delete the survivorship rule.

> **Note:**
>
> You cannot delete a survivorship rule that has any match results that have already been merged using the rule. If the survivorship rule is no longer needed, you can disable it instead.

28

Working with Roles and Permissions

Security ensures that each user has the right access to functions and data.

Videos

Your Goal	Watch This Video
Learn about security in Oracle Fusion Cloud Enterprise Data Management.	Overview: Understanding Security in Enterprise Data Management Cloud Part 1 Overview: Understanding Security in Enterprise Data Management Cloud Part 2

There are two levels of security:

1. In Oracle Fusion Cloud EPM, users are created and assigned predefined roles.
2. In Cloud EDM, users and groups are assigned application roles and permissions. After a user has access to a viewpoint, the viewpoint data security controls the allowed actions on nodes and properties.

The first level of security is managed in Cloud EPM through My Services when your Identity Domain Administrator creates users. Each user is assigned a predefined role.

There are two Cloud EDM predefined roles:

- User

  A user with the *User* predefined role can log into Cloud EDM and can be assigned roles and permissions to view and manage application data.

- Service Administrator

  A user with the *Service Administrator* predefined role can perform most functional activities and can access views, applications, and data including:

  - Provisioning roles in Cloud EDM and creating groups, see Access Control in *Administering Access Control*.
  - Migrating artifacts across test and production environments, see Using Migration.
  - Performing daily maintenance, see Using the Maintenance Snapshot in *Getting Started Guide for Administrators*.

  However, Service Administrators cannot be assigned to a request or a request subscription unless they have *Participant(Write)* permission on the data objects in that request.

If you do not have either the User or Service Administrator predefined role, you are not able to access Cloud EDM, and the following message is displayed:

"**No valid role for this user.** You must have either the User or Service Administrator predefined role assigned in Oracle Fusion Cloud Enterprise Performance Management in order to log into Enterprise Data Management Cloud. Contact your system administrator to have a role assigned."

Working with Groups in Cloud EDM

You can use groups to grant permissions to your applications, dimensions, node types, hierarchy sets, and views. Enterprise Data Management offers three types of groups:

- PREDEFINED: These groups are automatically created for each predefined role. All users are assigned to a predefined group based on their predefined role (e.g., User).
- EPM: These are the groups that you create in Access Control in Tools.
- IDCS: These are the groups that you create in the Oracle Cloud Identity Console. You can view them in Access Control and assign them to application roles and EPM groups.

For more information, see Managing Groups in *Administering Access Control*.

Understanding Application Roles and Permissions

The second level of security is managed in Oracle Fusion Cloud Enterprise Data Management using a combination of application roles and permissions that are assigned to users. By default, Service Administrators can perform all functions and access data. They do not need application roles or permissions.

Videos

Your Goal	Watch This Video
Learn about setting up users and groups.	▶ Managing Users and Roles using IAM

Application Roles

Application roles control the functions users can perform, such as creating views or registering applications and dimensions. Application roles are compatible and independent from permissions. See Oracle Enterprise Data Management Cloud and Enterprise Data Management in *Administering Access Control*.

> **Note:**
>
> If a user is renamed in Access Control, then all references to that user (such as in approval policies) become invalid.

The following application roles are available:

- **Access Control - Manage**: Enables you to manage groups, assign application roles to users, and report on user security. See Access Control in *Administering Access Control*
- **Access Control - View**: Enables you to view reports on user security such as the Role Assignment Report, User Login Report, and User Group Report (see Access Control in *Administering Access Control*). However, you cannot assign application roles, manage groups, or perform any other update operation in Access Control.
- **Application - Create**: Enables you to register an application in Cloud EDM. After you register an application, you are assigned *Owner* permission to the application.
- **Audit**: Enables you to view changes made to data in all applications. It does not grant the ability to make any changes to data. If you want to make changes to data, you must be

granted at least *Participant (Write)* permission on a data object. See Auditing Transaction History.

- **Migrations - Administer**: Enables you to import and export migration snapshots, import and export applications and dimensions to templates, and clone environments. See Migrating Across Environments.
- **Views - Create**: Enables you to create views. After you create a view you are assigned *Owner* permission to the view.

Permissions

Permissions secure access to applications, dimensions, data chain objects (node set, hierarchy set, and node type), and data. You can assign the following permission levels:

- *Owner*
- *Metadata Manager*
- *Data Manager*
- *Participant*

> **Note:**
>
> By default, the *Participant* permission grants *Read* access to data chain objects. You can grant *Write* access to those objects by configuring data access. See Configuring Data Access.

You assign these permissions on applications, dimensions, hierarchy sets, and node types. Applications and dimensions support all permission levels, while hierarchy sets and node types support the *Participant* permission only. See Working with Permissions.

> **Note:**
>
> You can also assign the *Owner* permission on a view. This permission enables you to configure the view and to assign the *Owner* permission to other users and groups for that view.

Data Access

For users with *Participant* permission, data access enables you to specify which actions they can take and which properties they can view or edit for specific data chain objects.

See Configuring Data Access.

Viewpoint Actions and Properties

In addition to assigning roles and permissions to users, you can specify the actions that users can perform and the properties that they can update in a viewpoint. For example, you can specify that users can add nodes, but not delete them, or that users can edit the description of a node, but not the name. These allowed actions and editable properties for a viewpoint are enforced for all users, regardless of their role or permission.

> **Note:**
>
> You must have *Data Manager* permission on a dimension in a viewpoint and *Owner* permission on the view that contains that viewpoint in order to specify the allowed actions and editable properties for a viewpoint.

See Changing Viewpoint Permissible Actions and Configuring How a Viewpoint Displays Properties.

Inclusive and Additive Permissions

Permission levels are either inclusive (higher level permissions include all of the permissions of the lower levels), or additive (the permission grants additional access without being included in other permission levels).

Inclusive Permissions

There are three inclusive permission levels that you can assign to data objects:

- **Owner** (includes all Inclusive and Additive permissions)
- **Data Manager**
- **Participant**

> **Note:**
>
> The **Participant** permission automatically grants **Read** data access to a data chain object. You can modify the permission to grant **Write** data access to data chain objects by selecting the allowed actions and property access for that object. See Configuring Data Access.

When you assign a higher level permission (such as *Owner*) to a user or group, that grant includes all of the lower level permissions (*Data Manager*, *Participant (Write)*, and *Participant (Read)*). You do not need to assign a user or group multiple permissions on the same data object.

Applications and dimensions support all inclusive permission levels, while hierarchy sets and node types support the *Participant* permissions only. You can assign multiple permission levels (such as *Participant* and *Metadata Manager*) to applications and dimensions. See Combining Permissions.

You can further refine the *Participant* permission by specifying data access to a hierarchy set or node type. See Configuring Data Access.

Additive Permissions

Additive permissions provide more granular access to applications and data chain objects. They can be assigned by themselves or on top of an inclusive permission.

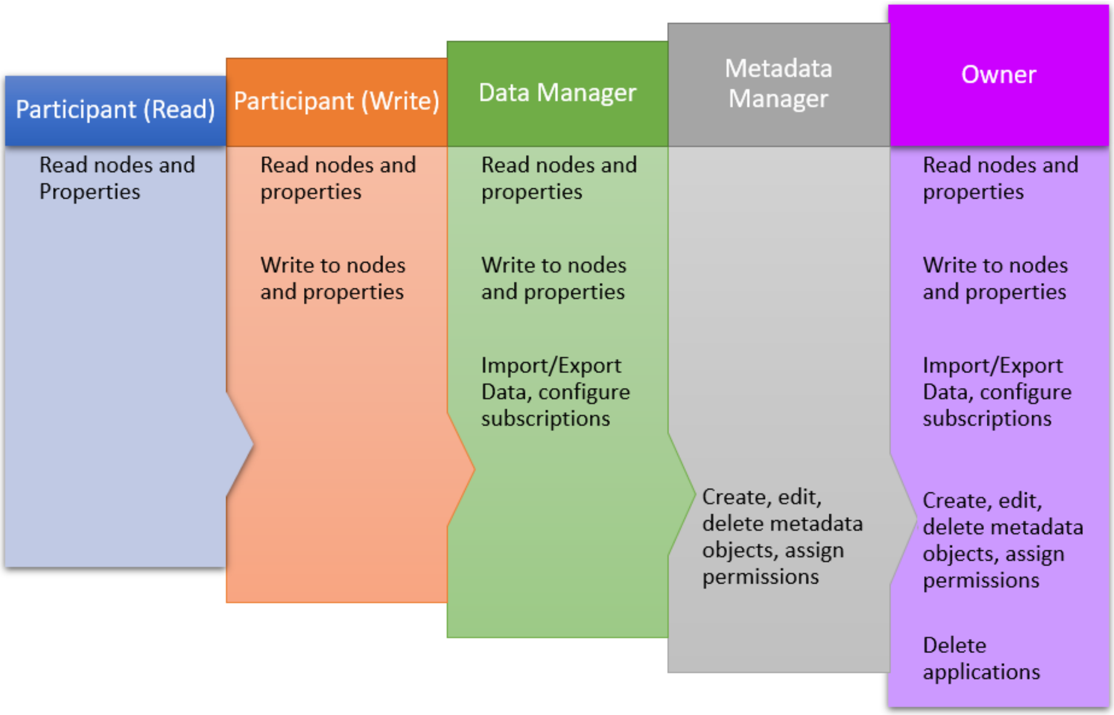
The additive permission that you can assign is **Metadata Manager**. This permission can be assigned at the application or dimension level, and it allows users to create, edit, and delete all metadata objects in an application or dimension including node types, hierarchy sets, converters, custom validations, permissions, and policies.

Note:

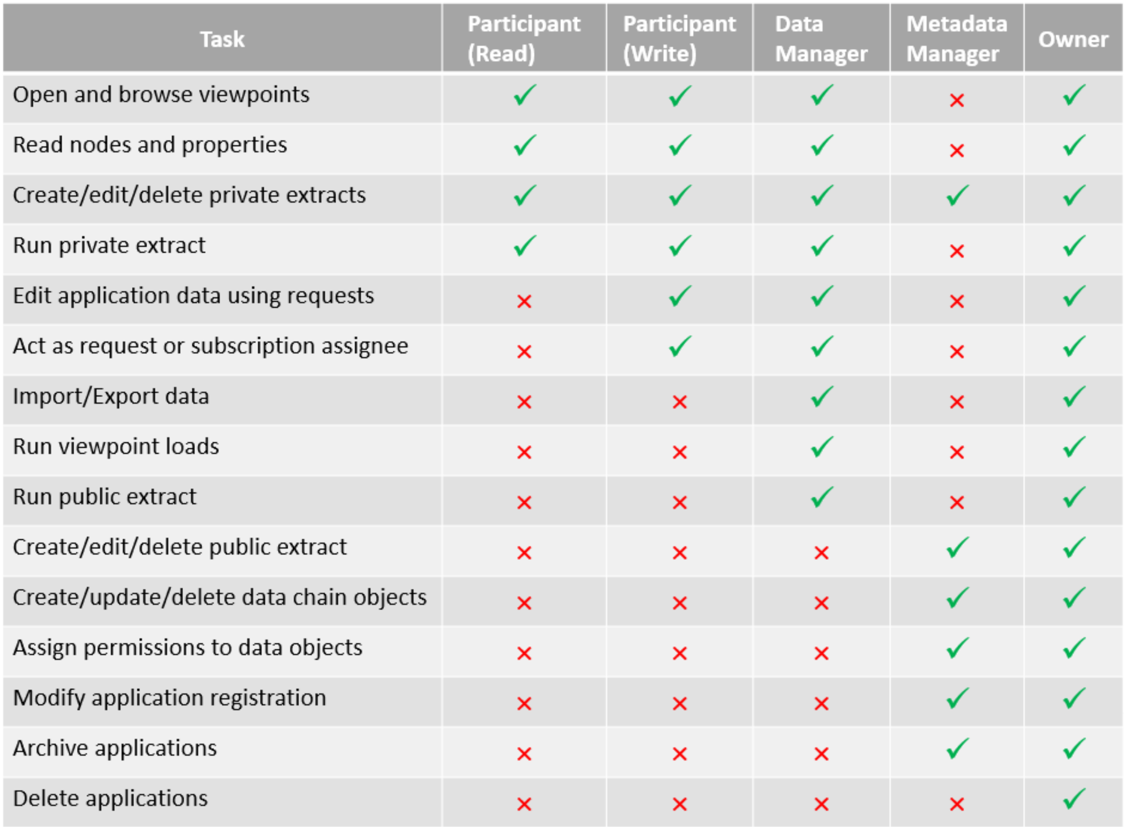
Metadata Manager permission does not give a user access to create or delete applications.

Metadata Manager permission does not give access to data. Users with this permission must be assigned one of the inclusive permission levels (such as *Participant (Read)*) in order to view data.

The following graphic illustrates how the inclusive and additive permission levels work. The three inclusive permissions build on each other, with *Participant (Write)* including the access of *Participant (Read)*, and *Data Manager* including the access of *Participant (Write)*. *Metadata Manager* is additive, so it does not include any of the lower level permissions. *Owner* includes all of the access in both types of permission levels, plus the ability to delete applications.


Participant (Read)
Read nodes and Properties
Participant (Write)
Read nodes and properties
Write to nodes and properties
Data Manager
Read nodes and properties
Write to nodes and properties
Import/Export Data, configure subscriptions
Metadata Manager
Create, edit, delete metadata objects, assign permissions
Owner
Read nodes and properties
Write to nodes and properties
Import/Export Data, configure subscriptions
Create, edit, delete metadata objects, assign permissions
Delete applications

The following table displays some commonly performed tasks and the permission level needed to perform them:



Task	Participant (Read)	Participant (Write)	Data Manager	Metadata Manager	Owner
Open and browse viewpoints	✓	✓	✓	×	✓
Read nodes and properties	✓	✓	✓	×	✓
Create/edit/delete private extracts	✓	✓	✓	✓	✓
Run private extract	✓	✓	✓	×	✓
Edit application data using requests	×	✓	✓	×	✓
Act as request or subscription assignee	×	✓	✓	×	✓
Import/Export data	×	×	✓	×	✓
Run viewpoint loads	×	×	✓	×	✓
Run public extract	×	×	✓	×	✓
Create/edit/delete public extract	×	×	×	✓	✓
Create/update/delete data chain objects	×	×	×	✓	✓
Assign permissions to data objects	×	×	×	✓	✓
Modify application registration	×	×	×	✓	✓
Archive applications	×	×	×	✓	✓
Delete applications	×	×	×	×	✓

Combining Permissions

You can combine inclusive and additive permissions to provide a finer level of control over a user's access. Some of the combinations include:

- *Data Manager* plus *Metadata Manager*: This combination provides a user access to perform most tasks around both data and metadata (such as creating, updating, and deleting data chain objects, creating and running public extracts, and running imports and exports) but does not allow a user to delete an application.
- *Participant (Read)* plus *Metadata Manager*: This combination grants data access to a metadata manager so that they can browse and validate data in viewpoints. This lets them, for example, ensure that the expressions that they create are working as intended.

Permission Cascading

Best Practice

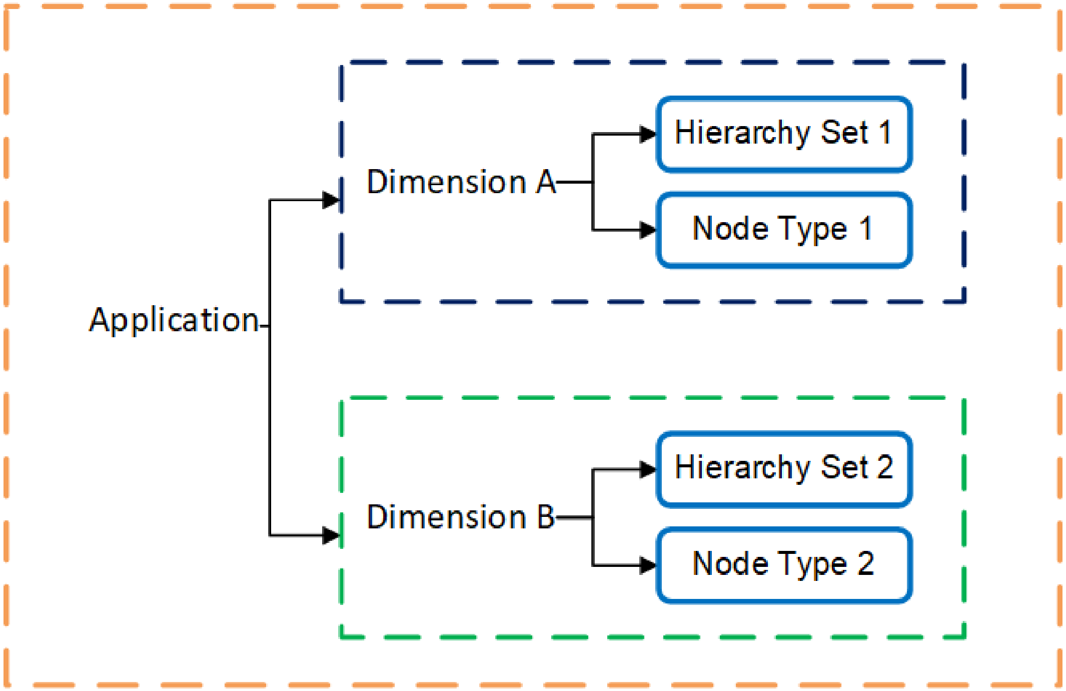
A best practice is to assign permissions at the most general level first (for example, at the application or dimension level), and then to assign permissions at more specific levels (such as hierarchy sets or node types) only if there are specific business requirements that must be met.

Considerations

- Permissions assigned to an application will be applied to all dimensions in that application. For example, in the following image, if you assign *Data Manager* permission to a user on the application, that user will have *Data Manager* permission on Dimension A and Dimension B as well.

- Permissions assigned to a dimension will be applied to all node types and hierarchy sets in that dimension. For example, in the following image, if you assign a user *Participant (Write)* permission on Dimension A, that user will have *Participant (Write)* permission on both Hierarchy Set 1 and Node Type 1.

The following diagram illustrates these concepts:



Note:

You can also assign the *Owner* permission to a view. However, data access is controlled at the data object (application, dimension, hierarchy set, and node type) level. The *Owner* permission on a view enables a user to configure the view and to assign the *Owner* permission to other users and groups for that view, but it does not grant access to the data objects in that view.

Working with Permissions

Permissions secure access to applications, dimensions, data chain objects, and data.

Videos

Your Goal	Watch This Video
Learn about permissions.	Setting Up Roles and Permissions in Oracle Enterprise Data Management Cloud

Permission Levels and Data Objects

The following table lists the permissions that you can assign to data objects and the actions that can be performed. For more information on the permission levels and cascading, see Permission Cascading.

Table 28-1 Data Object Permissions and Actions

You must have at least this permission	On this data object	To perform these actions:
Metadata Manager	Application	• Create, update, and delete all data chain objects • Assign permissions for all data objects to other users and groups • Modify application registration
Owner	Application	Delete an application
Metadata Manager	Dimension	• Create, update, and delete node sets, hierarchy sets, and node types in this dimension • Assign permissions to data chain objects in this dimension to other users and groups
Metadata Manager	Dimension	Archive and unarchive a dimension
Data Manager	Application	Import or export data for all dimensions in the application
Data Manager or *Metadata Manager*	Application	• Create and update a viewpoint for all dimensions in the application **Note:** You also need *Owner* permission on the view that contains the viewpoint. • Create and update a viewpoint subscription for all dimensions in the application **Note:** You also need *Owner* permission on the view that contains the viewpoint.

Table 28-1 (Cont.) Data Object Permissions and Actions

You must have at least this permission	On this data object	To perform these actions:
Data Manager or *Metadata Manager*	Dimension	- Create and update a viewpoint for this dimension **Note:** You also need *Owner* permission on the view that contains the viewpoint. - Create and update a viewpoint subscription for this dimension **Note:** You also need *Owner* permission on the view that contains the viewpoint.
Data Manager	Dimension	Import, export, and update data for this dimension
Participant (Write)	Application	Create a new request or act as a request assignee for any dimension in an application
Participant (Write)	Dimension	Create a new request or act as a request assignee for requests that use this dimension
Participant (Write) **Note**: The user or group can perform only the request actions that their data access permits.	Hierarchy Set	Create a request or act as a request assignee for requests that insert, move, remove, and reorder nodes in a hierarchy set **Note:** When you grant a user *Participant (Write)* permission on a hierarchy set, that user is also granted implicit *Participant (Read)* permission on any node type in that hierarchy set.

Table 28-1 (Cont.) Data Object Permissions and Actions

You must have at least this permission	On this data object	To perform these actions:
Participant (Write) **Note**: The user or group can perform only the request actions that their data access permits.	Node Type	Create a request or act as a request assignee for requests that add or delete nodes or that update node properties **Note:** When you grant a user *Participant (Write)* permission on a node type, they are not granted any implicit permissions on the hierarchy sets that use that node type. Thus, that user cannot add or delete nodes in a hierarchy set unless they are also granted *Participant (Write)* permission to that hierarchy set.
Participant (Read)	Application	Browse a viewpoint that contains data for any dimension in the application
Participant (Read)	Dimension	Browse a viewpoint that contains data for this dimension
Participant (Read)	Hierarchy Set	Browse a viewpoint that contains data for this hierarchy set and the node types in this hierarchy set **Note:** When you grant a user *Participant (Read)* permission on a hierarchy set, that user is also granted *Participant (Read)* permission on the node type in that hierarchy set.
Participant (Read)	Node Type	Browse a viewpoint that contains data for this node type in a list only

Adding, Removing, and Editing Permissions

You add, remove, and edit permissions to a data object from the Permissions tab of the object inspector. You must have *Owner* or *Metadata Manager* permission on the application or dimension that contains the data object in order to do so.

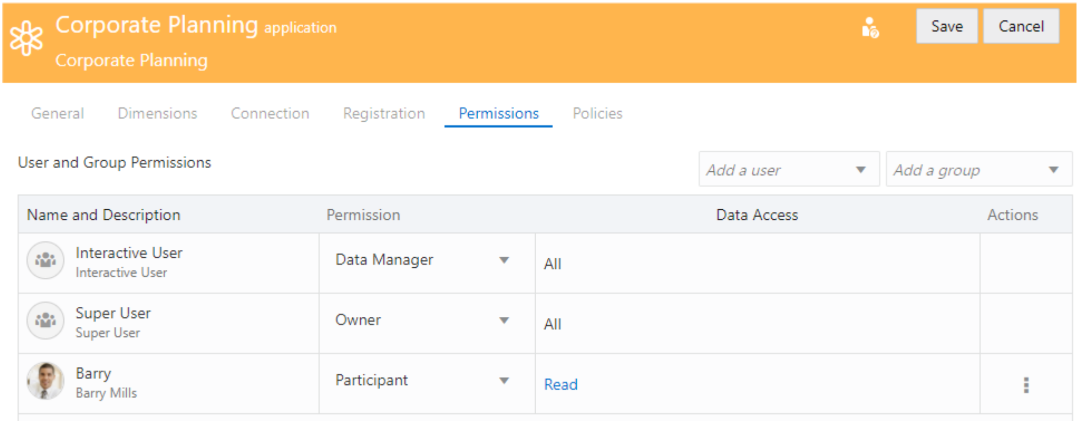
> **Note:**
>
> You can run a report to determine the permissions that have been assigned across all applications. For more information, see Working with Reports.

To add permissions to a user or group on a data object:

1. Open the object inspector for the data object. See, for example:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Hierarchy Set
   - Inspecting a Node Type
   - Inspecting a View
2. On the Permissions tab, click **Edit**.
3. In the **Add a user** or **Add a group** drop down list, select the user or group that you want to grant permission to.

   > **Note:**
   >
   > Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control*.

4. (Optional) To display the users in a group, click ⓘ.
5. In the **Permission** drop down list, select the permission level to grant to the user or group.



Name and Description	Permission	Data Access	Actions
Interactive User Interactive User	Data Manager	All	
Super User Super User	Owner	All	
Barry Barry Mills	Participant	Read	⋮

6. (Optional) For the *Participant* permission, you can also specify data access. See Configuring Data Access.
7. Click **Save**.

To remove a permission from a user or group on a data object:

1. Open the object inspector.
2. On the Permissions tab, click **Edit**.
3. Navigate to the permission that you want to remove, click ⋮ in the Actions column, and then select **Remove**.
4. Click **Save**.

To edit a permission on a data object:

1. Open the object inspector.
2. On the Permissions tab, click **Edit**.
3. In the **Permission** drop down list, select a new value for the permission that you want to edit.
4. Click **Save**.

To see how roles and permissions work together, see Security Examples.

Configuring Data Access

For users with *Participant* permission, data access enables you to specify which actions they can take and which properties they can view or edit for specific data chain objects.

You can specify access to data in two ways:

- **Allowed Actions**: You can specify the actions that can be performed on a data chain object in a request.
- **Property Access**: You can specify which properties are displayed or hidden, and which properties are able to be edited for a data chain object.

By default, when you assign the *Participant* permission to a user or group, their data access is set to **Read**. This means that their allowed actions on the data chain object are set to None, and their property access is set to Display only. If you grant the participant any allowed action or Edit access on at least one property, their data access changes to **Write** for that data chain object.

The following table lists the data access for allowed actions and property access that you can set on each data chain object for users with *Participant* permission.

Table 28-2 Data Access on Data Chain Objects

Data Chain Object	Allowed Actions	Property Access
Application	• None • All	• Display All • Edit All
Dimension	• None • All	• Display All • Edit All
Hierarchy Set	• None • All • Specified – Insert – Move – Remove – Reorder	Not applicable **Note:** You cannot set access to properties at the hierarchy set level. Use one of the other data chain objects, such as node type, to control access to properties.

Table 28-2 (Cont.) Data Access on Data Chain Objects

Data Chain Object	Allowed Actions	Property Access
Node Type	• None • All • Specified – Add – Delete	• Display All • Edit All • Specified – Display – Edit – Hide

Considerations

- Data access can be configured for users with *Participant* permission only. Users with *Owner* or *Data Manager* permission on a data chain object are automatically granted All access to all actions and properties on that data chain object. For example, Owners and Data Managers are always able to see properties that are set to Hidden.
- For applications and dimensions, you can specify only All or None for the allowed actions and Display All or Edit All for the property access. If you want to specify more granular actions or property access, such as allowing only adds and deletes or displaying only certain properties, you must specify those at the hierarchy set or node type level.

  > **Note:**
  >
  > This means, for example, that a property cannot be hidden at the application or dimension level. You hide properties at the node type level only.

- You cannot assign Edit access to properties that are never editable (for example, those in the Core namespace other than `Core.Name` or `Core.Description`, or any properties in the CoreStats namespace). You also can't set the `Core.Name` property to Hide.

Data Access Cascading

Just as with permissions, data access cascades from higher to lower level data chain objects (for example, if a user has an allowed action of Add in a dimension, they can add in the hierarchy sets and node types in that dimension.) See Permission Cascading.

For allowed actions and editable properties, the *least* restrictive setting is used. For example, if a user has no allowed actions at the dimension level but an allowed action of Add at the node type level, that user can perform add actions for that node type.

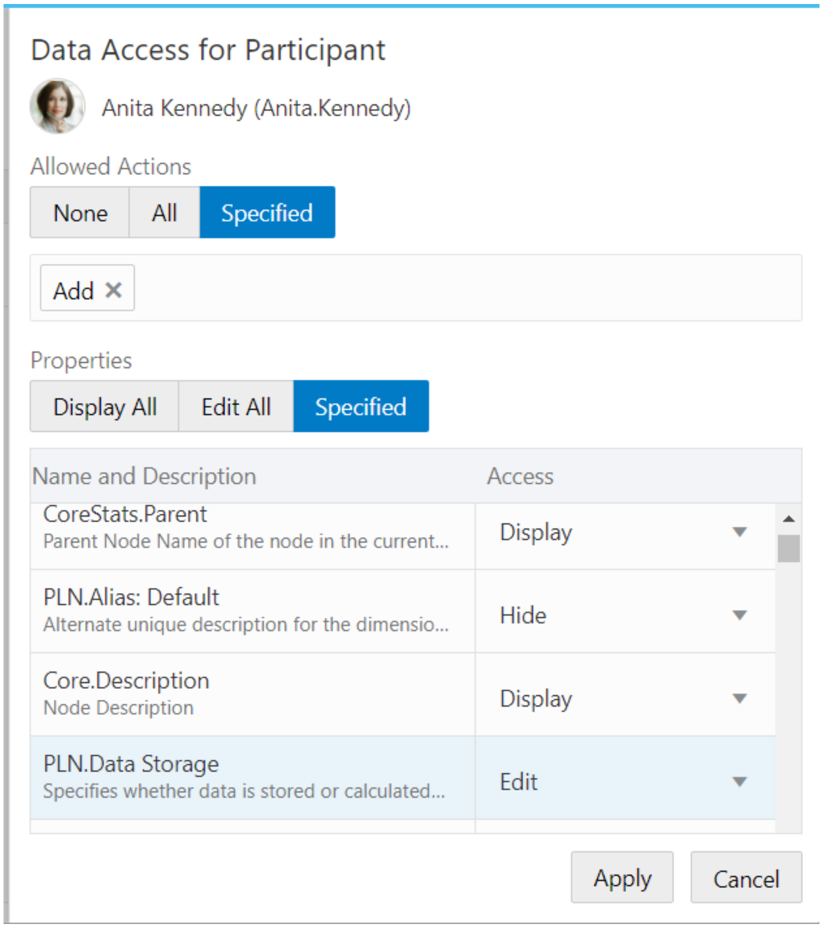
For hidden properties, the *most* restrictive setting is used. If a property is hidden on a node type, that setting overrides any other permission. For example, if a user with *Participant* permission has Display All on an application's properties but the Cost Center property is hidden at the node type, that user is unable to see that property in a viewpoint.

Configuring Data Access

1. Inspect the data chain object that you want to configure data access for:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Hierarchy Set
   - Inspecting a Node Type

2. On the Permissions tab, click **Edit**.
3. Perform an action:
   - To edit a permission:
     a. For the permission that you want to modify, perform either one of these actions to display the data access panel:
        – In the Data Access column, click the permission level (**Read** or **Write**)
        – In the Actions column, click ⋮ and select **Edit Actions**.
     b. In the Data Access for Participants panel, select the Allowed Actions and Displayed Properties settings for the user or group. See the Table 1 table, above, for details on what settings can be applied for each data chain object.
     c. Click **Apply**, and then click **Save**.
   - To remove a permission: In the Actions column, click ⋮, and then select **Remove**.

As an example, the following screenshot shows a Participant permission configured with **Add** as an allowed action, the `CoreStats.Parent` and `Core.Description` properties set to **Display**, the `PLN.Alias:Default` property set to **Hide**, and the `PLN.Data Storage` set to **Edit**.



Data Access for Participant

Anita Kennedy (Anita.Kennedy)

Allowed Actions

None | All | Specified

Add ✕

Properties

Display All | Edit All | Specified

Name and Description	Access
CoreStats.Parent Parent Node Name of the node in the current...	Display
PLN.Alias: Default Alternate unique description for the dimensio...	Hide
Core.Description Node Description	Display
PLN.Data Storage Specifies whether data is stored or calculated...	Edit

Uploading and Downloading Permissions and Data Access from a File

When assigning permissions at the node type level, you can upload an Excel file with the permissions and data access instead of manually assigning them in the node type inspector.

You can also download permissions and data access on a node type to an Excel file for offline reference.

Considerations

- You can download and upload permissions and data access on node types only.
- You must have *Owner* or *Metadata Manager* permission on the application or dimension that contains the node type in order to upload permissions for it.
- When uploading permissions, the permissions from the file are merged with existing permissions. Existing permissions and data access for user and groups are overwritten with the ones in the file.

Uploading Permissions and Data Access from a File

Your permissions file must be in the correct format. See Permissions File Format

> **Tip:**
>
> You can download the current permissions on a node type to use as a template that you can modify with additional changes and then upload back to the system. See Downloading Permissions and Data Access to a File.

To upload permissions:

1. Inspect the node type that you want to upload permissions for. See Inspecting a Node Type.
2. Click **Edit**.
3. Click **Load Permissions from File** ⭱.

   > **Note:**
   >
   > You must be in Edit mode to see the **Load Permissions from File** button.

4. Browse to the permission file, and then click **Open**.
   The permissions are loaded to the system.
5. **Optional**: In the confirmation dialog box, click the link to download the permission file and review the **Status** and **Message** columns to see the results of each permission.
6. Click **OK** on the confirmation dialog box, and then click **Save** to save your changes.

   > **Note:**
   >
   > If you click **Cancel**, the permissions that you uploaded are not saved to the node type.

Downloading Permissions and Data Access to a File

To download permissions:

1. Inspect the node type that you want to download permissions for. See Inspecting a Node Type.
2. On the Permissions tab, click **Download Permissions to File** ⤓.

> **Note:**
>
> The **Download Permissions to File** button is not visible if you are in Edit mode.

After the file is downloaded, you can view it in Excel.

Permissions File Format

This topic describes the format of Microsoft Excel spreadsheets used to load permissions and data access.

The permissions file must be an Excel file with a sheet named **Permissions**.

> **Note:**
>
> The file can contain other sheets. For example, when you download permissions the file contains a sheet named Summary with summary information. This summary sheet is informational and is not required for upload.

Permissions Sheet Format

Considerations

- The header columns must be in the order specified below. The node type properties columns can be in any order.
- If any of the information in the header columns is invalid, the record is skipped.
- Properties in the file that are not valid for the node type are ignored, but the record is still processed.
- If there are multiple records for the same user or group in the file, the first record in the file is processed. Any subsequent records are skipped and marked with a skipped status and a corresponding message.

Header Columns

- **Access Type**: Select **User** or **Group**
- **Name**: Enter the user or group name for which permissions are being created.
- **Permission**: Select a valid permission. For node types, this is **Participant**.
- **Allowed Actions**: Select **None**, **All**, or **Specified**
- **Specified Actions**: If you selected **Specified** in Allowed Actions, enter the list of actions separated by commas (for example: **Add**, **Delete**).
- **Properties Access**: Select **Edit All**, **Display All**, or **Specified**

Node Type Property Columns

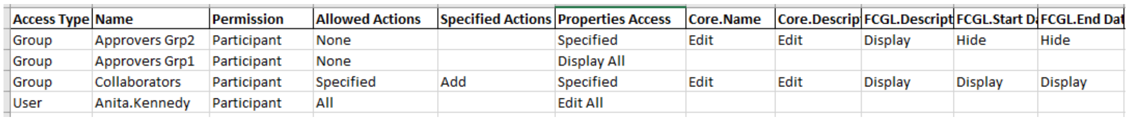
If you selected **Specified** in Properties Access, enter the fully-qualified (Namespace.Property) node type properties , one per column, and one of these values for each property: **Display**, **Edit**, or **Hide**.

> **Note:**
>
> You cannot specify Hide on the Core.Name property.

Status Columns

- **Status**: System reserved column that displays the status of the permission (**Success** or **Skipped**) after you upload it.
- **Message**: System reserved column that displays an informational message on permissions that were skipped.

The following example shows the correct format for the Permissions sheet:



Access Type	Name	Permission	Allowed Actions	Specified Actions	Properties Access	Core.Name	Core.Descrip	FCGL.Descript	FCGL.Start D	FCGL.End Dat
Group	Approvers Grp2	Participant	None		Specified	Edit	Edit	Display	Hide	Hide
Group	Approvers Grp1	Participant	None		Display All					
Group	Collaborators	Participant	Specified	Add	Specified	Edit	Edit	Display	Display	Display
User	Anita.Kennedy	Participant	All		Edit All					

Security Examples

These examples show the combined roles and permissions required for:

- Security for Applications and Dimensions
- Security for Hierarchy Sets and Node Types
- Security for Views and Viewpoints
- Security for Requests
- Security for Access Control, Migration, and Daily Maintenance

Security for Applications and Dimensions

This table lists the minimum combined user roles and permissions to work with applications and dimensions.

Application or Dimension Task	You need at least this permission
To register an application, a user needs the *Application - Create* role. **Note**: After the application is registered and created, the user is automatically assigned the *Owner* permission for the application and the default view.	
Assign permissions on an application or dimension to another user or group	*Metadata Manager* on the application or dimension
Inspect an application or modify an application registration	*Metadata Manager* on the application.

Application or Dimension Task	You need at least this permission
Delete an application	*Owner* on the application
Open and browse an application	*Participant (Read)* on the application.
Edit application data	*Participant (Write)* on the application or dimension **Note:** While the *Participant (Write)* permission enables you to edit application data in a viewpoint, the data access that is specified and the viewpoint allowed actions and editable properties have final control over the actions that can be done. See Configuring Data Access and Changing Viewpoint Permissible Actions.
Import or export data	*Data Manager* on the application or dimension
Create or edit dimensions	*Metadata Manager* on the application or dimension
Create, edit, and delete global connections	*Metadata Manager* on the application
Create, edit, and delete constraints	*Metadata Manager* on the application
Create and edit bindings	*Metadata Manager* on the application or dimension
Create, edit, copy, promote, delete public extracts	*Metadata Manager* on the application or dimension

Note:

A Service Administrator can register, import, export, and modify all applications and dimensions, as well as assign all permissions to applications, data objects, and views.

Security for Hierarchy Sets and Node Types

This table lists the minimum combined user roles and permissions required to create, view, and edit hierarchy sets and node types.

Data Chain Objects	You need at least this permission
Create, inspect, edit, archive, or delete a hierarchy set or node type	*Metadata Manager* on the application or dimension that contains the data chain objects
View the application data in hierarchy sets or node types	*Participant (Read)* on the application or dimension will let you browse all data chain objects in that application or dimension. *Participant (Read)* on the hierarchy set will let you browse the hierarchy set, as well as the node type that is used in that hierarchy set. *Participant (Read)* on the node type will let you browse that node type only.

Note:

A Service Administrator can see and update all data chain objects.

Security for Views and Viewpoints

This table lists the minimum combined user roles and permissions required for views and viewpoints.

Viewpoints	You need this permission	You need this View permission
Create a view **Note:** You need the *Views - Create* role. After you create a view, you are automatically assigned the *Owner* permission to the default view.		
Assign permissions on a view		*Owner* on the view
Update application data in a viewpoint using requests **Note:** The assigned data access, viewpoint allowed actions, and editable property settings determine the actions you can take on nodes and properties.	• *Participant (Write)* for the application or dimension can make any update to data. • *Participant (Write)* for the hierarchy set can insert, move, reorder and remove nodes in that hierarchy set • *Participant (Write)* for the node type can add and delete nodes and update node properties	
Create, edit, archive, or delete a viewpoint	*Data Manager* or *Metadata Manager* for the application	*Owner* for the view containing the viewpoint
Create a subscription in a viewpoint	*Data Manager* or *Metadata Manager* for the dimension	*Owner* for the view containing the viewpoint
Copy a viewpoint	*Owner* for the dimension	*Owner* for the view containing the viewpoint

Viewpoints	You need this permission	You need this View permission
Browse, inspect, compare, validate, or download viewpoints	Complete *Participant (Read)* permission on the node set, as follows: • **List-type node set**: must have *Participant (Read)* access to all node types. • **Hierarchy type node set**: must have *Participant (Read)* access to the hierarchy set. **Note:** If you do not have complete *Participant (Read)* permission on a node set, the viewpoints for that node set are not displayed when you open a view.	No view permission is needed to browse a viewpoint **Note:** *Owner* permission on the view that contains a viewpoint does not enable you to see the viewpoints in that view. You must have complete *Participant (Read)* permission to a node set in a viewpoint in order to see that viewpoint when you open a view. For example, if you have *View* permission on a view that contains five viewpoints, but you have *Participant (Read)* permission to data objects in only three of those viewpoints, when you open the view you will see only those three viewpoints.
Open or inspect a view	*Participant (Read)* permission on any data object in any viewpoint in that view	*Owner* permission on the view will enable you to open the view; however, you need *Participant (Read)* permission on a data object in a viewpoint in that view in order to see any viewpoints in that view.
Edit or archive a view		*Owner* for the view

Note:

A Service Administrator can see and update all views and viewpoints and assign permissions to a view.

Security for Requests

Permissions and Data Access

The following table describes the permissions and data access that are required for request workflow actions.

Table 28-3 Request Actions and Permissions

To perform this request workflow action:	You need this permission:
Assign a request	You must have *at least one* of these roles or permissions: • Current request assignee • *Owner* permission on the view • Service Administrator role

Table 28-3 (Cont.) Request Actions and Permissions

To perform this request workflow action:	You need this permission:
Be assigned a request	*Participant (Write)* access on at least one data chain object in *every* viewpoint in a request, as follows: • For Add, Delete, or Update Properties actions, the user must have *Participant (Write)* on the node type. • For Insert, Move, Reorder, or Remove actions, the user must have *Participant (Write)* on the hierarchy set.
Be added as a collaborator on a request	*Participant (Write)* on at least one data chain object in *at least one* viewpoint in the request. The request actions and property access for request items in a request are determined by the data access permissions on both the request assignee and the collaborator, as follows: • For allowed actions, both the assignee and the collaborator must be able to perform the request action in order for that action to be available. For example, in order to insert a node into a hierarchy, both the assignee and collaborator must have Insert as an allowed action on the hierarchy set. If *either* user does not have the Insert permission, then the option to insert a node is not available. • Similarly, a property can be edited only if both the assignee and the collaborator have Edit access to that property.
Make changes to a request using a request load file	• The user uploading the request file must have *Participant (Write)* access (either directly or through permission cascading) to the hierarchy set or node type in order to load a request file. • If the load file contains an action or a property update that the user does not have data access to perform (for example, the load file contains a Delete action and the user has data access for Add actions only), the request action is loaded as a request item but it is identified as a validation error when the request is validated. • If the load file contains an update for a property that is hidden from the user, that request item is not loaded in the request but it is included in the attached file.

Table 28-3 (Cont.) Request Actions and Permissions

To perform this request workflow action:	You need this permission:
Create a subscription	You must have *all* of these permissions: • *Participant (Read)* access (either directly or through permission cascading) to the hierarchy set or node type in the source viewpoint. • *Data Manager* permission on the dimension in the target viewpoint • *Owner* permission on the target view
Be eligible to be assigned as a default or alternate assignee of a subscription	• *Participant (Write)* access (either directly or through permission cascading) to the hierarchy set or node type in the target viewpoint. **Note:** If you are assigning a group, at least one member of the group must have *Participant (Write)* permission to the hierarchy set or node type in the target viewpoint in order for that group to be available as a selection. • If the subscription source request contains an action or a property update that the user does not have data access to perform (for example, the source request contains a Delete action and the user has data access for Add actions only), the request action is loaded as a request item in the target subscription request but it is identified as a validation error when the request is validated. • If the subscription source request contains an update for a property that is hidden from the user, that request item is not loaded in the target subscription request but it is included in the attached file.
Approve a request	None, initially. When you add user or group to a policy for a data object, that user or group is granted implicit *Participant (Read)* permission on that data object. See Configuring Policies
Enrich a request during approval stage	Because enrichers are approvers, they are automatically granted implicit *Participant (Read)* permission on the data objects in the approval policy. In order to make changes during enrichment, enrichers must also have *Participant (Write)* on the data chain objects in the request that they want to change. Enrichers can edit request items according to their data access and permissions.

Users

This section describes actions users and Service Administrators can perform on completed and draft requests.

Draft Requests

- **Current assignees**:
  - Perform request actions according to their permissions and data access
  - Load request items
  - Delete request items
  - Submit the request
  - Download the request items to a file
  - Add, edit, or delete comments and attachments
- **Previous participants**:
  - View request items, comments, and attachments
  - Add request comments and attachments
  - Inspect the request
  - Validate the request
  - Inspect, validate, and compare viewpoints in the request
  - Download the request items to a file

The creator of a request comment or attachment can edit the comment or attachment while the request is in Draft status.

Completed Requests

Users can see completed requests if they have *Participant (Read)* access to the view in which the request was made.

> **Note:**
>
> Completed requests cannot be modified, as they provide a historical audit trail.

Service Administrators

Service Administrators can view all requests.

Service Administrators can modify or delete a draft request if they are the current assignee.

> **Note:**
>
> A Service Administrator can be designated as the assignee for a request if they have *Participant (Write)* permission on the data for which the request was made.

Service Administrators can not modify or submit requests that they are not assigned to, nor can they approve, reject, or push back requests that they are not approvers on.

Security for Access Control, Migration, and Daily Maintenance

The following table lists the permission or role needed to perform access control, migration, and maintenance tasks.

Task	Permission or Role Needed
Provision roles in Oracle Fusion Cloud Enterprise Data Management and create groups, see Access Control.	Service Administrator or Access Control - Manage
Migrate artifacts across test and production environments, see Using Migration.	Service Administrator or Migrations - Administer
Perform daily maintenance, see Using the Maintenance Snapshot.	Service Administrator

29 Configuring Policies

You can set up policies at the application, dimension, node type, or hierarchy set level. Approval policies are applied during the Approve workflow stage, and they enable approvers to review requests and approve or reject their contents. Commit policies are applied during the Commit workflow stage, and they enable a user on the commit policy to do a final review and commit of all changes in a request. Notification policies are applied at the Closed workflow stage when a request is committed, and they enable users to receive email notifications when a completed request is submitted by other users.

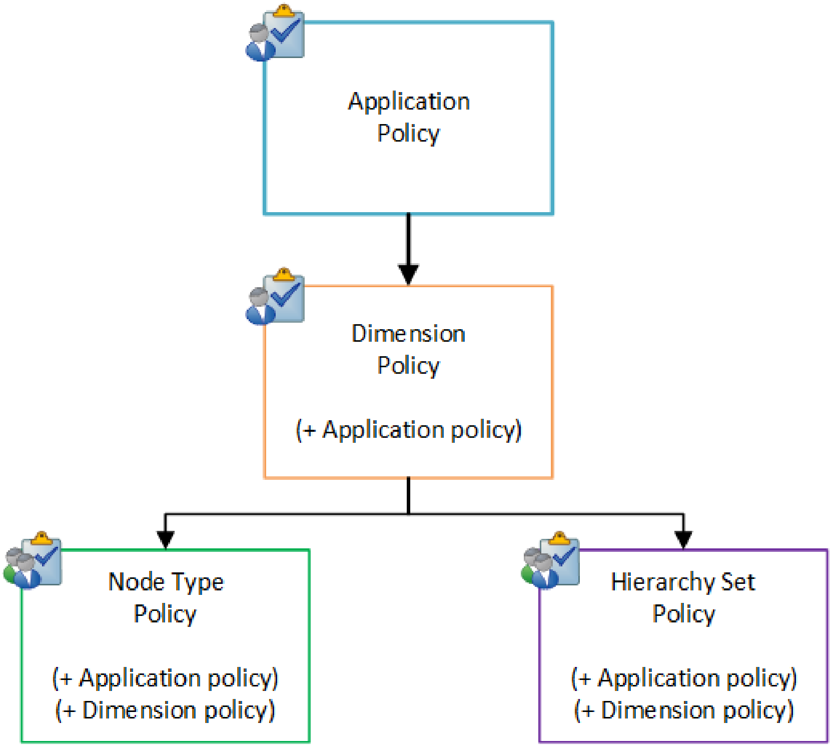
For more information, see

- Approval Policies
- Commit Policies
- Notification Policies

Policies and Data Objects

Policies are enabled at the application, dimension, hierarchy set, or node type level. As with permissions, policies propagate from applications to dimensions, and then to hierarchy sets and node types. That is, a policy that is configured on an application will be applied to all of the dimensions in that application, and policies configured on dimensions will apply to the node types and hierarchy sets in that dimension. See Permission Cascading for more information.

> **Note:**
>
> You can run a report to determine the polices that have been assigned across all applications. For more information, see Working with Reports.



Policies and Request Actions on Data Objects

Use the **Included Actions** filters on your policies (see Creating and Enabling Approval Policies, Creating and Enabling Commit Policies, and Creating and Enabling Notification Policies) to specify if a policy should triggered only for specific request actions. The type of action that you specify is not affected by the data chain object that the policy is on. For example, you can specify a Move action on a policy for a node type, even though the Move action is a hierarchy set action. When you do so, the policy will be triggered for all Moves in all hierarchy sets for that node type.

Propagating Policies on Data Objects

Policies propagate from higher level objects (such as applications) to lower objects in a data chain (such as dimensions, node types or hierarchy sets). A data chain object may be affected by policies directly enabled for it as well as propagating policies from objects above it in the data chain. In this case, each policy is evaluated independently.

To illustrate this, suppose you have the following approval policies enabled:

- An approval policy at the application level requires three approvers from Accounting.
- An approval policy at the dimension level requires four approvers from Accounting.

These policies do not get combined, so a total of seven approvers from Accounting are required. Instead, each policy is evaluated separately, and since both policies require approval from the same Accounting group, the first three approvals from Accounting are applied to both policies, as follows:

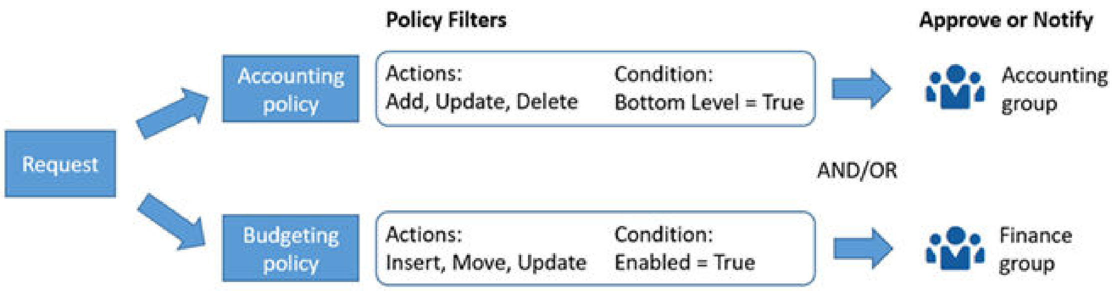
- The application-level policy is satisfied when three approvers from Accounting approve the request.
- The dimension-level policy is satisfied when one additional approver from Accounting approves the request.

> **Note:**
>
> If an approver is in multiple approval groups, they approve a request only once. That approval will count for all approval policies that the user is a member of. For example, if Barry is a member of both the Accounting group and the Cost Center group, and an approval policy states that two approvals from each group are required, Barry's approval counts as one approval from each group.

Multiple Policies for the Same Data Object

You can define multiple policies for the same data chain object to allow different users to approve or be notified for different types of requests. Additional policies can be created with different filters to handle conditional approvals or notifications of specific data sets by different users. For example, the following diagram illustrates a dimension with two policies:

- An accounting policy sends approval requests or notifications to the Accounting group if bottom level nodes are added, updated, or deleted
- A budgeting policy sends approval requests or notifications to the Finance group if nodes are inserted, moved, or updated.



Policies and Permissions

You must have *Owner* or *Metadata Manager* permission on a data object in order to configure a policy for that object.

When you add an user or group to a policy for a data object, that user or group is granted implicit *Participant (Read)* permission on that data object. Because permissions propagate, if you add a user or group to a policy for an application or a dimension, that user or group is also granted implicit *Participant (Read)* permission to the data chain objects that the application or dimension contains (dimensions, node types, and hierarchy sets).

When a user or group is added to a policy and granted implicit *Participant (Read)* permission on a hierarchy set, they are also granted implicit *Participant (Read)* permission on the node type used in that hierarchy set. This enables them to open a view and browse the hierarchy set when approving or viewing completed requests. However, granting users implicit *Participant (Read)* permission on a node type by adding them to a policy does *not* grant them implicit *Participant (Read)* permission on any hierarchy sets that use that node type. If a user has *Participant (Read)* permission on a node type but not a hierarchy set, they are unable to open the view and browse the hierarchy set in the viewpoint to approve or view the request. Instead, they must approve the request in the Request Inspector.

If you remove a user from a policy, that user's implicit *Participant (Read)* permission on the data object in that policy is revoked, but they retain any permissions that were explicitly granted to that data object. See Working with Permissions.

> **Note:**
>
> If a user’s name is changed in Access Control, that user becomes invalid and can no longer participate in any policies that they were assigned to.

Approval Policies

Approval policies enable approvers to review requests and approve or reject their contents. You configure the request by specifying who can approve the request and how many approvals are needed. You can also allow approvers to enrich the request by adding, editing, and removing request items.

After the approval policy requirements are met, the request can be committed. Reviewers can also reject the request, which closes the request and sends a notification to the request submitter, or push the request back to the submitter so that the submitter can make edits and resubmit.

For more information, see

- Creating and Enabling Approval Policies
- Policy Reminders and Escalations
- Modifying Policies
- Approval Policy Examples

Videos

Your Goal	Watch This Video
Learn about configuring approval policies.	Configuring Approval Policies in Oracle Enterprise Data Management Cloud

Creating and Enabling Approval Policies

You create and enable approval policies on the Policies tab of the inspector for applications, dimensions, hierarchy sets, and node types. The Policies tab for each of these data chain objects contains an empty approval policy for that object, which is disabled by default. You must enable the policy on the data chain object in order to require approvals for that object. You can also create additional approval polices that have different filters to handle conditional approvals of specific data sets by different users.

To create and enable an approval policy:

1. Open the object inspector for the data object. See, for example:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Hierarchy Set

- Inspecting a Node Type

2. On the Policies tab, perform an action:
   - To enable the default approval policy, click the name of the approval policy. By default, this is `Approval`.

     The approval policy is displayed in the policy inspector.
   - To create a new approval policy, click **Create**. From **Policy Type**, select **Approval** and enter a name and optional description for the policy. Click **Create**.
3. (Optional) On the General tab, click **Edit** and then change the policy name or description. Click **Save**.
4. On the Definition tab, click **Edit**, and define the policy settings.

Field Name	Valid Values
Enabled	Allows you to enable or disable a policy without affecting the policy settings. By default, this is set to **False**.
Approval Method	• **Parallel**: Approvers can approve in any order at the same time. • **Serial**: Approvals must be given one at a time, in the order of the users and groups listed in the **Policy Groups** list. • **Ownership**: Approvers are invited to approve based on an ownership property on each node. See Understanding Ownership Approvals. • **Management Hierarchy**: Approvers are invited to approve based on a management hierarchy from a Users type application. See Understanding Management Hierarchy Approvals.
One Approval Per Group (Serial and Parallel policies only)	**Serial policies**: • **Selected**: Only one approval per policy group is required. The **Total Required** field is automatically set to the number of groups listed in **Policy Groups**. • **Deselected**: More than one approval from each group is required. Specify the approvals required for each group in the **Approvals Per Group** column in the **Policy Groups** list. **Parallel policies**: • **Selected**: Only one approval per policy group is required. The **Total Required** field is automatically set to the number of groups listed in **Policy Groups**. • **Deselected**: More than one approval from each group is required. Specify the total approvals required across all groups in **Total Required**.
Ownership Property (Ownership policies only)	The property that determines the node owner who will approve changes on each node. The property must be a Node data type property with the User subtype (see Node and Node List Data Type Subtypes).
Management Hierarchy Node Set (Management hierarchy policies only)	The node set that contains the management hierarchy to use for approvals for this policy

Field Name	Valid Values
Fulfillment Type (Management hierarchy policies only)	• **Fixed**: The policy is fulfilled after a specified number of approval levels has been met. Enter the number of approval levels required to fulfill the policy in **Fulfillment Levels**. • **Variable**: The policy is fulfilled when a specified fulfillment expression returns a value of True for the user node in the management hierarchy. Enter an expression to evaluate after each approval in **Fulfillment Expression**. If the expression returns False, the policy invites the next user in the hierarchy. If the expression returns True, the policy is fulfilled.
Total Required	The total number of approvers required across all users and groups. • If **One Approval Per Group** is selected, this field is automatically set to the number of groups listed in the **Policy Groups** list. • If **One Approval Per Group** is not selected: – **Serial policies**: This field is automatically set to the total number of approvers across all groups that you specify in **Policy Groups**. Use the **Approvals Per Group** field to specify how many approvals are required for each group. – **Parallel policies**: Specify the total number of approvers required across all groups. For Ownership and Management Hierarchy policies, this field is automatically set to one approval per owner (for Ownership) or user (for Management Hierarchy) and cannot be changed.
Include Submitter	Select to specify that if the request submitter is also an approver, the submitter's approval is granted automatically. For example, if you select this option on a policy that requires three approvals and the submitter is part of an approval group on that policy, the submitter's approval is granted automatically and the policy will require only two additional approvals before it can be committed. **Note:** If **Include Submitter** is disabled, the request submitter is not invited to be an approver even if they are listed on the approval policy. This setting is not applicable to Management Hierarchy policies.
Allow Enrichment During Approval	Select to enable approvers to enrich the request by adding, modifying, and removing request items according to their permissions and data access. See Understanding Request Enrichment.
Prevent Auto-Approval for Enriched Requests	When using serial policies or policy order, select to prevent requests from being automatically approved by users who have already approved it if the request has subsequently been enriched. When enabled, if a user approves a request and then the request is enriched (see Understanding Request Enrichment), the user is invited to approve the request again instead of that approval being granted automatically.

Field Name	Valid Values
Allow Setting Future Time Label	Select to enable invitees to set or remove a future time label for a request. See Working with Future Dated Requests.
Reminder Notification	Enter the number of days that will elapse before a reminder email is sent to the current invitees. **Note:** If this is set to zero, reminders are not sent to current invitees if they do not take action on an approval.
Approval Escalation	Enter the number of reminders that will be sent before the request approval is escalated to a user with *Data Manager* permission to the data object in the policy. **Note:** If this is set to zero, requests will not be escalated for timeouts. See Policy Reminders and Escalations
Policy Order	Assign a number for the order of the policy. Policies with the same number are fulfilled as a group. When all of the policies in a group have been fulfilled, policies with the next order number sequentially are evaluated. See Understanding Policy Order.

5. In **Policy Groups**, click the **Add a policy group** drop down list to add users and groups to the approval list.

> **Note:**
>
> Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control*.

If you selected **Serial** as the approval method, approvals must be given in the order specified in this list. You can drag and drop to reorder the users and groups in the list. You must click the user or group first, and then drag it to the desired position.

> **Note:**
>
> The users and groups in this list are granted *Participant (Read)* permission on the data objects in the policy.

> **Note:**
>
> For Ownership and Management Hierarchy policies, the **Policy Groups** list is empty and cannot be changed.

6. (Optional): For Serial policies where **One Approval per Group** is not enabled, use the **Approvals per Group** field to specify the total number of approvals required for each group in the list.
   The **Total Required** field is automatically updated with the total of all approvers across all groups.
7. Click **Save**.
8. (Optional) On the Filters tab, click **Edit**, and define action filters and node conditions for the policy.
   - In **Included Actions**, clear the **All** check box and, from the **Included Actions** list, select actions to add. Action filters enable you to set up approvals for only the actions that you select.
   - In **Properties**, perform an action:
     - Select **All** to invite users to approve whenever any properties are updated.
     - Select **Include** and then select specific properties to invite users to approve only when those properties are updated.
     - Select **Exclude** and then select specific properties to invite users to approve when any property *except* the ones that you selected are updated.
     - Select **None** to specify that users should not be invited to approve based on property updates.

     > **Note:**
     >
     > The request item base action does not have to be Update in order to use the Properties filter. For example, if your Properties filter includes the Core.Description property, then any update to the description property will qualify for the property filter, even if the base request action was an Add or an Insert.

   - In **Node Condition**, click **Define Expression** *fx* to launch the expression builder. Because you are evaluating nodes in the viewpoint where the action was performed, you use the `node` object in the expression builder.

     The expression builder guides you to create an expression that evaluates to a Boolean value only. See Building Expressions.
   - In **Request Types**, select the type of requests that this policies applies to, or select the **All** check box to apply the policy to all request types. See Request Types and Process Flow.
     - Consolidation
     - Import
     - Interactive
     - Load

- Subscription

- Click **Save**.

Understanding Request Enrichment

Request enrichment enables approvers on a policy to modify request items before approving the request. The items and actions that can be modified depend on the approver's permissions and data access in the view.

When you enable enrichment on an approval policy, any approver on the policy who has *Participant (Write)* permission on at least one data object in the view can perform enrichment during the approval phase. The actions that an enricher can take depend solely on the permissions and data access of the enricher in the view.

> **Note:**
>
> The data access of the request submitter is not taken into account. This means that an enricher can potentially perform actions in a request that the original request submitter could not.

The changes that an enricher can make are not limited to the policy or the data objects in the original request. If enrichment is enabled on any policy on any data object in a request, enrichers can perform any request action in the view that their permissions and data access allows.

For example, suppose you have a maintenance view that contains viewpoints for a general ledger application and a Planning application, and your general ledger application has an approval policy with enrichment enabled. When a request to add a cost center to the general ledger is submitted, if an approver on the GL policy also has *Participant (Write)* access to the Planning application, they can add the cost center to the Planning application as well before approving.

Considerations

- If an approver is included in more than one policy on a data chain object, the approver can enrich the request if *any* of the policies has enrichment enabled on it.
- The enricher must be currently invited in order to make changes to a request. If the request is not in the Approval stage (for example, if another approver pushed the request back), the enrichers can not make changes to the request.
- When determining the request actions and property updates that an enricher can perform, the permissions and data access in the view of the enricher are taken into account (see Configuring Data Access). For example:
  - If an enricher has data access to perform only Adds in a node type, they cannot add or delete a request item to Delete a node.
  - If an enricher has Display only access to a property, they cannot add or delete a request item to modify that property.
- Enrichers can perform request actions that negate or change previous actions performed by the submitter or other enrichers.

Validating and Approving Enriched Requests

Whether or not an enricher makes changes to a request, since enrichers are also approvers they must approve the requests that are assigned to them. When an enricher approves the request, data validations are performed based on the enricher's permissions and data access. The validation and approval process is as follows:

- If a validation error occurs:
  - If the approver is an enricher and is responsible for any actions that failed validation, the request is not approved. The request remains open in the view, and a message indicating the validation failure is displayed. The enricher must take action on the request to correct the validation failure, either by correcting any data issues, pushing the request back to the submitter, or rejecting the request.
  - If the approver is not responsible for any actions that failed validation, the request is approved and new invitees are calculated.
- If no validation errors occur, the request is approved and new invitees are calculated.

When all approval policies have been satisfied, if there are still validation errors the submitter is notified. The submitter can delete any request items that are causing the validation issues, including request items that contain actions or properties that the submitter's permissions and data access don't include.

When no validation errors remain, the request is committed.

Understanding Ownership Approvals

The Ownership approval method allows you to invite approvers based on an ownership property that you define for nodes in a viewpoint.

The ownership property on a node identifies the user who is responsible for approving requests for that node. After you assign an ownership property on the policy, the users that you identify are invited to approve any changes in request items that include that node. Each invited node owner must approve the request in order to fulfill the policy.

> **Note:**
>
> The term "owner" in the context of Ownership approval methods is not the same as the *Owner* permission on a data chain object. Node owners get invited as approvers on an approval policy, which grants only the ability to approve requests for that node. Node owners are not automatically granted any additional permissions for that node, such as the ability to view or edit that node.

Considerations

- You can configure the Ownership method on Approval policies only. You cannot configure the Ownership method on Commit or Notification policies.
- You can define multiple ownership properties on a node, which enables you to have more than one node owner as approvers for that node. Each property must be on a separate approval policy, since you can assign only one ownership property per policy.
- The users that you designate as owners are chosen from a Users application (see Working with Users Applications) using Node data type properties with the User subtype (see Node and Node List Data Type Subtypes).

- If the owner of a node is changed while a request is In Flight, the new owner is invited to approve and the previous owner is no longer invited to approve. If the previous owner has already approved the request, that approval is retained in the request history.
- If a node owner is not available (for example, the owner is not a valid user or is out of the office), the request is escalated. See Policy Reminders and Escalations.

Process Flow

The following describes the general process flow when setting up ownership approvals. It is not intended as a step-by-step procedure, but rather an overview of the steps that are required in order to utilize the ownership approval method.

1. **Set up a Users application to store your users.** (See Working with Users Applications.) **Important**: In order to be assigned as a node owner, a user in your Users application must be a valid user in Oracle Fusion Cloud Enterprise Data Management (`CoreStats.EDM User` = True). The system uses the `User.Email Address` property in the Users application to determine valid users. See Predefined Properties for Users Applications.
2. **Create a Node data type property with the User subtype to use as an ownership property.** See Creating Properties Manually.
3. **Associate the property with a node type in the viewpoint.** See Adding a Property to a Node Type.

   > **Tip:**
   >
   > For the **Assigned Node Set** and **Allowed Node Types**, select the appropriate node set and node types from the Users application that you created in step 1.

4. **Configure an approval policy to use the Ownership method**. (See Creating and Enabling Approval Policies).
   a. For Approval Method, select **Ownership**.
   b. For Ownership Property, select the property that you configured as an ownership property (created in step 2, above).
   c. Enable and save the policy.
5. **Assign the node owners:**
   a. In the viewpoint, create a request and set the value of the ownership property (created in step 2, above) for each node to a user in your Users application.

      > **Tip:**
      >
      > Setting the value of this property to be inherited or derived can simplify this process.

   b. Submit and complete the request.

Request Processing

After you have configured your Users application, viewpoint, and approval policies for ownership approvals, invitees and policy fulfillment are calculated based on the value of the ownership property. When a request with an ownership policy reaches the Approve stage:

1. The Ownership policy is evaluated on the node for each request item that is associated with the policy.
2. For each node, the value of the ownership property is checked, and each user referenced by the ownership property is invited to approve the request. The users must be valid users in Cloud EDM.
3. Each invitee must approve the request in order to fulfill the policy.
4. If an invitee is not a valid user or is out of the office, the request is escalated.

Best Practices

- Node owners can always approve requests via the request inspector (see Request Inspector Actions). However, in order to be able to visualize the request changes in a viewpoint, node owners must have at least *Participant (Read)* access to the viewpoints that contains the nodes that they are owners on. See Security for Views and Viewpoints.
- Enable the **Ownership Approval Permission** predefined validation to display a validation error or warning when a request or viewpoint is validated if the node owner does not have at least *Participant (Read)* access to the viewpoint. See Predefined Validations.
- When you create the property that you will be using as the ownership property, set it as Required in the viewpoint so that node owners are always assigned to each node.
- Set up ownership properties as inherited (in a hierarchy viewpoint) or derived from a lookup (in a list viewpoint) so that when new nodes are added the ownership property is automatically populated.
- When you configure your ownership property, use the **Selection Criteria Property**, **Select Criteria Operator**, and **Selection Criteria Value** fields to ensure that only valid users are able to be assigned as node owners. See Node and Node List Data Type Parameters.
  For example:
  - **Selection Criteria Property**: `EDM User`
  - **Select Criteria Operator**: `Equal`
  - **Selection Criteria Value**: `1` (True)
- Create a separate list viewpoint specifically to maintain node ownership. This enables you to govern node ownership separately from any data changes.
- If you add a user in Access Control that you want to use as a node owner, you must also add them in your Users application before they can be used as an ownership approval user.

Understanding Management Hierarchy Approvals

The Management Hierarchy approval method enables you to invite approvers based on the management hierarchy of the request submitter.

When creating the approval policy you select a hierarchy node set from a Users application that contains your management hierarchy. Then, after a request is submitted the request submitter's parent is invited to approve the request. When that user approves the request, the policy continues up the management hierarchy until the policy is fulfilled.

Considerations

- You can configure the Management Hierarchy method on Approval policies only. You cannot configure it on Commit or Notification policies.

- The node set that you select for the management hierarchy must meet the following conditions:
  - It must be from a Users application. (See Working with Users Applications)
  - It must be a hierarchy node set and not a list node set (that is, the node set must have an associated hierarchy set).
  - It cannot support shared nodes.
- The users in your management hierarchy must be assigned at least *Participant (Read)* permission on the viewpoint associated with the policy in order to be able to approve the request. See Security for Views and Viewpoints.

> **Note:**
>
> Unlike node owners in a policy using the Ownership approval method, users that do not have at least *Participant (Read)* permission on the viewpoint cannot approve requests via the request inspector. Instead, if a user in the management hierarchy does not have the appropriate permission, the request is escalated. See Policy Reminders and Escalations.

- Use the **Fulfillment Type** setting to determine how the management hierarchy policy is fulfilled:
  - **Fixed**: The policy is fulfilled after a specified number of approval levels has been met.
  - **Variable**: The policy is fulfilled when a specified fulfillment expression returns a value of True. The node context of the expression is of User type.

  For example, you could use a Fixed setting of `2` to specify that the policy is fulfilled when the request submitter's parent and the next higher ancestor in the hierarchy approves the request, or you could use the Variable setting to specify that the policy is fulfilled when an approval is received where the `CoreStats.Level` property on the hierarchy node is greater than `2`.

Request Processing

For Management Hierarchy policies, invitees and policy fulfillment are calculated based on the node structure of the management hierarchy node set. When a request with a Management Hierarchy policy reaches the Approve stage:

1. The Management Hierarchy node set is evaluated and the node associated with the request owner is located.
2. The user associated with the parent node of the request owner node is invited to approve the request.
3. When each invitee approves the request, the policy is evaluated for fulfillment:
   - If Fulfillment Type is **Fixed**, the policy is fulfilled if the number of approvals specified by the Fulfillment Levels setting has been met.
   - If Fulfillment Type is **Variable**, the Fulfillment Expression is evaluated and the policy is fulfilled if the expression returns True.
4. If the policy is fulfilled, the request advances to the next stage. If the policy is not fulfilled, the next higher ancestor in the node set is invited to approve the request.

Request Escalation

Requests are escalated if any of these conditions are true:

- There is no valid User type hierarchy node set associated with the policy.
- The request owner is not found in the management hierarchy.
- The invitee is not a valid user in Oracle Fusion Cloud Enterprise Data Management (that is, the `EDM User` property on the user does not equal `True`. See Predefined Properties for Users Applications).
- The invitee does not have at least *Participant (Read)* permission on the viewpoint associated with the policy.
- The top node invitee has approved the request but the policy is not fulfilled. For example, the policy requires five approval levels but there are only four levels in the hierarchy chain.
- The invitee is out of office or does not have an email address configured.

Understanding Withdrawing Request Approvals

Users who have approved a request can withdraw their approvals while the request is still In Flight. For serial policies, when a user withdraws an approval all subsequent approvals are also removed from all approval policies associated with the request.

Considerations

- The user must have contributed an Approve action to the request.
- The request must still be In Flight.
- The request submitter cannot withdraw their approval. They must recall the request to make changes to it.

When an approver withdraws an approval from a request, the withdrawer's approval is removed from all policies associated with the request. For serial policies, all subsequent approvals are also removed from all policies associated with the request. This may affect policies that the original withdrawer was not a part of.

> **Note:**
>
> For parallel approval policies, withdrawing an approval does not affect the other approvals, regardless of whether those other approvals were given before or after the approval that was withdrawn.

Serial Policy Approval Withdrawal Example

Suppose you had two serial approval policies. For policy 1, Alex, Barry, and Anita approved the request. For policy 2, Casey, Anita, and Amy approved the request (**bold** means an approval has been given):

Approval Policy 1 (Serial)

1. **Alex**
2. **Barry**
3. **Anita**
4. Henry

Approval Policy 2 (Serial)

1. **Casey**

2. **Anita**
3. **Amy**
4. Denise

Next, Barry withdraws his approval from policy 1. Now the policies look like this (**bold** means an approval has been given):

Approval Policy 1 (Serial)

1. **Alex**
2. Barry
3. Anita
4. Henry

Approval Policy 2 (Serial)

1. **Casey**
2. Anita
3. Amy
4. Denise

When Barry withdraws his approval from policy 1, Anita's approval is cleared in policy 1 as well, because her approval was subsequent to Barry's in the serial policy. But, Anita's and Amy's approvals in policy 2 are *also* cleared, even though Barry withdrew his approval in a different policy. And if Anita or Amy are approvers on any other policies associated with the request, those approvals are also cleared, and so on.

After all approvals are cleared, approvers are calculated and invitations to approve are sent out again. Approvers whose approvals were not cleared (Alex and Casey in the above example) are not invited again, as their approvals were not affected.

> **Note:**
>
> Approvers are calculated based on the current policies and permissions on the data chain objects in the request. Rarely, this may result in the approver who initially withdrew their request not being invited to approve again if a change was made after the withdraw action.

Approval Policy Examples

The following examples illustrate approval policies at the application, dimension, node type, and hierarchy set levels and demonstrate how the approvals are processed using various policy settings.

Example 1: Application Level Approval Policy

First, let's look at a simple example to show how approvals work on a basic level. In this example, there is an approval policy at the application level that states that at least two people from the GL Govern group must approve all changes to the chart of accounts.

Table 29-1 Example 1: Application Level Policy Settings

Fusion GL Application	Dimension	Node Type	Hierarchy Set
Policy A • **Approval groups**: GL Govern • **Method**: Parallel • **Total Required**: 2 approvers	Account dimension	Account node type	Account hierarchy set

The GL Govern group consists of Barry, Julie, and Jane. Tom is the owner of the Fusion GL application.

Approval workflow:

1. Mark submits a request to add an account and update a cost center description.
2. Because the approval method is parallel, a request is sent to all three members of the GL Govern group for approval.
3. Julie approves the request.
4. Barry approves the request.
5. The policy requirements are met, and the request is committed.

Example 2: Deadlock Escalation

Now, let's look at that same example again, but this time, Barry and Jane have been transferred out of the GL Govern group.

Table 29-2 Example 2: Deadlock Escalation Policy Settings

Fusion GL Application	Dimension	Node Type	Hierarchy Set
Policy A • **Approval groups**: GL Govern • **Method**: Parallel • **Total Required**: 2 approvers	Account dimension	Account node type	Account hierarchy set

The GL Govern group consists only of Julie. Tom is the owner of the Fusion GL application.

Approval workflow:

1. Mark submits a request to add an account and update a cost center description.
2. An approval request is sent to Julie.
3. Julie approves the request.
   Although the policy requires two approvers from the GL Govern group, Julie is the only person in that group. Since there are no more approvers available to meet the policy requirements, this results in a deadlock. As a result, the request is escalated to users who have *Data Manager* permission on the application. Because Tom is the owner of the application, his *Owner* permission includes the *Data Manager* permission.
4. The request is escalated to Tom.
5. Tom approves the request.

6. The policy requirements are met, and the request is committed.

Example 3: Dimension Level Serial Approval Policy

Next, let's look at a serial-type policy at the dimension level. In this example, Josh must approve the request, then Frank, and finally someone from the Accounting group.

Table 29-3 Example 3: Dimension Level Serial Policy Settings

Application	Dimension	Node Type	Hierarchy Set
Planning application	Account dimension **Policy A** • **Approval groups**: Josh, Frank, Accounting • **Method**: Serial • **Total Required**: 3 groups	Account node type	Account hierarchy set

The Accounting group consists of James and Heather.

Approval workflow:

1. Mark submits a request to move three accounts.
2. An approval request is sent to Josh.
3. Josh approves the request.
4. An approval request is sent to Frank.
5. Frank approves the request.
6. Approval requests are sent to the Accounting group (James and Heather).
7. Heather approves the request.
8. The policy requirements are met, and the request is committed.

Example 4: Node Type and Hierarchy Set Level Approval Policy

While approval polices at the application and dimension level are applied on all request actions, policies at the node type and hierarchy set level are applied only for specific request actions. Polices on a node type are applied only for requests that add or delete nodes, or that update node properties. Policies on a hierarchy set are applied only for requests that insert, remove, move, or re-order nodes in a hierarchy set, or that update node relationship properties.

To illustrate these principles, let's look at two requests for an example that has policies on both the node type and the hierarchy set. The first request updates a node property, so only the policy on the node set is applied. The second request adds accounts, which affects both the node type and the hierarchy set, and therefore both policies are applied.

Table 29-4 Example 4: Node Type and Hierarchy Set Level Policy Settings

Application	Dimension	Node Type	Hierarchy Set
Planning application	Account dimension	Account node type **Policy A** • **Approval groups**: Accounting, TaxUsers • **Method**: Parallel • **One Approval per Group**: True	Account hierarchy set **Policy B** • **Approval groups**: EssAdmins • **Method**: Parallel • **One Approval per Group**: False • **Total Required**: 5 approvals

Some additional background about these requests:

- The Accounting group consists of James and Heather.
- The TaxUsers group consists of Brian, Jane, and Rachel.
- The EssAdmins group has seven administrators in it (EssAdmin1 through EssAdmin7).

First, let's look at a request to update node properties. Node property updates are affected by policies on the node type only.

Request 1 Approval workflow:

1. Mark submits a request to update three account descriptions (node property update).
2. Approval requests are sent to the Accounting and TaxUsers groups.

> **Note:**
>
> Because a node property update does not affect the hierarchy set, the EssAdmins group does not get an approval request.

3. James approves the request for the Accounting group.
4. Rachel approves the request for the TaxUsers group.
5. The policy requirements are met, and the request is committed.

Next, let's look at a second request, this time to add nodes. As before, the policy on the node type is applied because the request action adds nodes. But for this request the policy on the hierarchy set is applied as well, because add actions create insert actions in hierarchy-based viewpoints.

Request 2 Approval workflow:

1. Mark submits a request to add three new accounts.
2. Approval requests are sent to the Accounting and TaxUsers groups.
3. Because an add action in a hiearchy-based viewpoint also creates insert actions in the hierarchy set, approval requests are also sent to the EssAdmins group.
4. James approves the request for the Accounting group.

> **Note:**
>
> Because James has implicit *Participant (Read)* permission on the node type but not the hierarchy set, he must approve the request in the Request Inspector. See Policies and Permissions.

5. Rachel approves the request for the TaxUsers group.

> **Note:**
>
> Because Rachel has implicit *Participant (Read)* permission on the node type but not the hierarchy set, she must approve the request in the Request Inspector. See Policies and Permissions.

6. Five Essbase administrators approve the request.

> **Note:**
>
> Because the EssAdmins group has implicit *Participant (Read)* permission on the hierarchy set, they are also granted implicit *Participant (Read)* permission on the node type. They can open a view and browse the hierarchy set to approve the request. See Policies and Permissions.

7. The policy requirements are met, and the request is committed.

Example 5: Approval with Enrichment Enabled

If enrichment is enabled on an approval policy, approvers with *Participant (Write)* access on any data object in the view for the request can modify the request before approving.

In this example, a request is made in a maintenance view with viewpoints for three applications: General Ledger, Planning, and Consolidation. Each application has an approval policy at the application level, and the GL and Planning policies have enrichment enabled.

Table 29-5 Example 5: Approval with Enrichment

General Ledger Application	Planning Application	Consolidation Application
GL Approval Policy • **Approval Groups:**Josh (General Ledger administrator) **Note:** Josh has *Participant (Write)* permission to the Planning application. • **Method:** Serial • **Total Required:** 1 • **Enrichment Enabled?** Yes	**Planning Approval Policy** • **Approval Groups:** Julie (Planning administrator) **Note:** Julie has *Participant (Write)* permission to the Consolidation application. • **Method:** Serial • **Total Required:** 1 • **Enrichment Enabled?** Yes	**Consolidation Approval Policy** • **Approval Groups:** Accounting • **Method:** Serial • **Total Required:** 1 • **Enrichment Enabled?** No

Approval Workflow:

1. Mark submits a request to add a cost center in the General Ledger application.
2. An approval request is sent to Josh, the GL administrator.
3. Josh enriches the request by adding the cost center to the Planning application and then approves the request for the GL application.
4. Because Josh added a node to the Planning application, the Planning approval policy is activated and an approval request is sent to Julie, the Planning administrator.
5. Julie further enriches the request by adding the cost center to the Consolidation application, and then approves the request for the Planning application.
6. The Consolidation approval policy is activated, and an approval request is sent to the Accounting group.
7. Barry, a member of the Accounting group, approves the request for the Consolidation application. Because enrichment is not enabled in the Consolidation policy, Barry cannot enrich the request further.
8. With the policy requirements of all of the approval policies met, the request is committed and the cost center is added to all three applications.

Commit Policies

Commit policies provide a final review stage for a request during which all request items are reviewed and committed by a user specified in the commit policy in order to enforce a segregation of duties. While you can specify multiple users on a commit policy, only one commit action is required to fulfill each policy.

> **Note:**
>
> No changes to the request are allowed during the commit stage.

After all approval policies for a request are fulfilled, if commit policies are enabled for any of the request items, users configured in the policy are notified via email that they must review and commit the request in order to finalize the changes. Committers can also reject the request, which closes the request and sends a notification to the request submitter, or push the request back to the submitter so that the submitter can make edits and resubmit for approval and then committing.

Considerations

- Users who are listed on both an approval policy and a commit policy are invited to commit even after they have approved the request.
- Enrichment is not allowed during the commit stage, but you can add comments and attachments to the request.
- When a request is in the commit stage:
  - The request submitter can still recall the request, which moves the request back to the Submit stage with a status of Recall.
  - Request approvers can still withdraw their approvals, which moves the request back to the Approve stage with a status of In Flight.
- You can have multiple commit policies on a data chain object. The request is not completed until all commit policies are fulfilled.

Creating and Enabling Commit Policies

You create and enable commit policies on the Policies tab of the inspector for applications, dimensions, hierarchy sets, and node types. You must enable the policy on the data chain object in order to require a commit stage for request items for that object.

To create and enable a commit policy:

1. Open the object inspector for the data object. See, for example:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Hierarchy Set
   - Inspecting a Node Type
2. On the Policies tab, click **Create**.
3. From **Policy Type**, select **Commit** and enter a name and optional description for the policy. Click **Create**.

   The new policy is displayed in the policy inspector.
4. (Optional) On the General tab of the policy inspector, click **Edit** and then change the policy name or description. Click **Save**.
5. On the Definition tab, click **Edit**, and define the policy settings.

Field Name	**Valid Values**
Enabled	Allows you to enable or disable a policy without affecting the policy settings. By default, this is set to **False**.
Commit Method	Automatically set to **Parallel** (multiple committers are invited at the same time) and cannot be changed for commit policies.
Total Required	Automatically set to **1** and cannot be changed for commit policies. Commit policies only require one commit in order to be fulfilled.
Include Submitter	Select to specify that if the request submitter is also an committer, the submitter's commit is granted automatically. **Note:** If **Include Submitter** is disabled, the request submitter is not invited to be an committer even if they are listed on the commit policy.
Allow Setting Future Time Label	Select to enable invitees to set or remove a future time label for a request. See Working with Future Dated Requests.
Reminder Notification	Enter the number of days that will elapse before a reminder email is sent to the current invitees. **Note:** If this is set to zero, reminders are not sent to current invitees if they do not take action on a commit.
Commit Escalation	Enter the number of reminders that will be sent before the request commit is escalated to a user with *Data Manager* permission to the data object in the policy. **Note:** If this is set to zero, requests will not be escalated for timeouts. See Policy Reminders and Escalations
Policy Order	Assign a number for the order of the policy. Policies with the same number are fulfilled as a group. When all of the policies in a group have been fulfilled, policies with the next order number sequentially are evaluated. See Understanding Policy Order.

6. In **Policy Groups**, click the **Add a policy group** drop down list to add users and groups to the commit list.

> **Note:**
>
> Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control*.

> **Note:**
>
> The users and groups in this list are granted *Participant (Read)* permission on the data objects in the policy.

7. Click **Save**.
8. (Optional) On the Filters tab, click **Edit**, and define action filters and node conditions for the policy.
   - In **Included Actions**, clear the **All** check box and, from the **Included Actions** list, select actions to add. Action filters enable you to set up commits for only the actions that you select.
   - In **Properties**, perform an action:
     - Select **All** to invite users to commit whenever any properties are updated.
     - Select **Include** and then select specific properties to invite users to commit only when those properties are updated.
     - Select **Exclude** and then select specific properties to invite users to commit when any property *except* the ones that you selected are updated.
     - Select **None** to specify that users should not be invited to commit based on property updates.

     > **Note:**
     >
     > The request item base action does not have to be Update in order to use the Properties filter. For example, if your Properties filter includes the Core.Description property, then any update to the description property will qualify for the property filter, even if the base request action was an Add or an Insert.

   - In **Node Condition**, click **Define Expression** *fx* to launch the expression builder. Because you are evaluating nodes in the viewpoint where the action was performed, you use the `node` object in the expression builder.

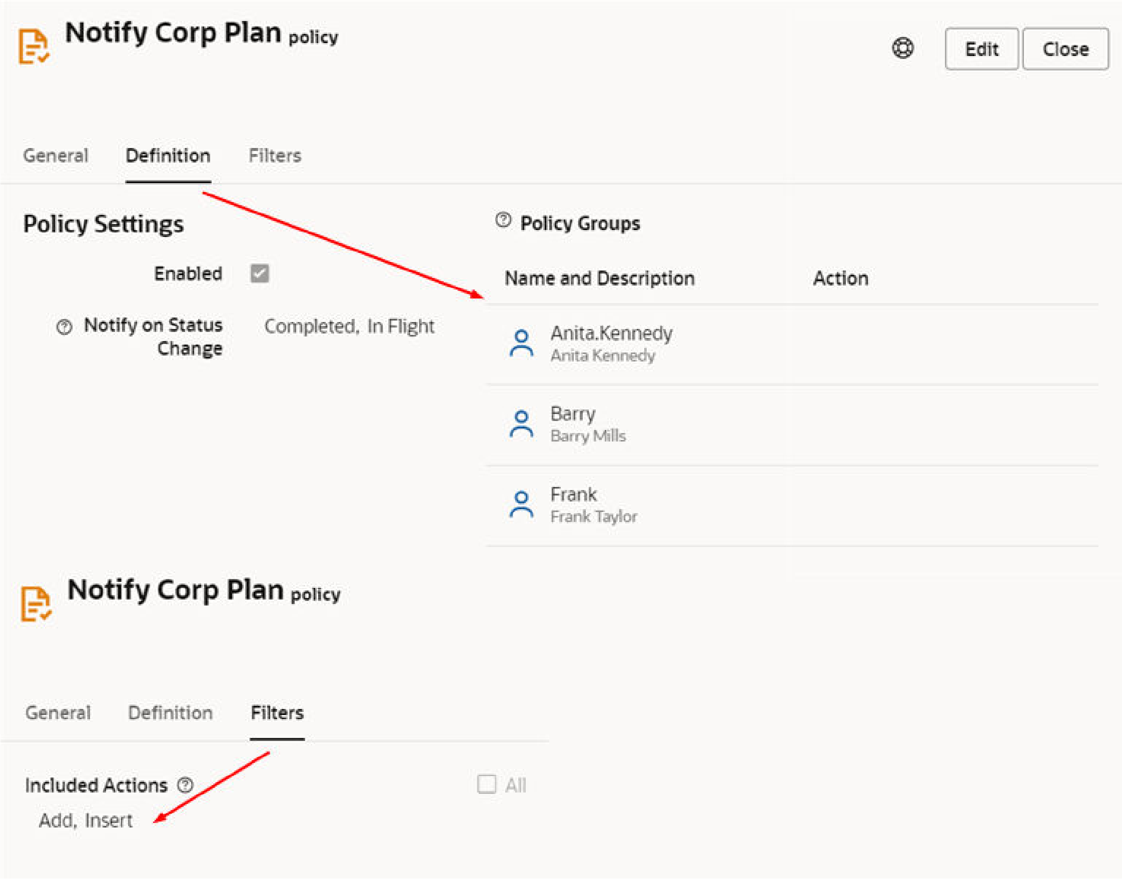
     The expression builder guides you to create an expression that evaluates to a Boolean value only. See Building Expressions.
   - In **Request Types**, select the type of requests that this policies applies to, or select the **All** check box to apply the policy to all request types. See Request Types and Process Flow.
     - Consolidation
     - Import

  - Interactive
  - Load
  - Subscription
- Click **Save**.

Notification Policies

Notification policies enable application or dimension owners to notify policy users when the request is changed to specific status. You can configure the policy to send an email notification to policy users when the request status is changed to In Flight, Completed, or both.

For example, this policy notifies members of the Finance department (Anita, Barry, and Frank) of both Completed and In Flight requests for nodes that were Added or Inserted in the Corporate Planning application.



Notify Corp Plan policy

Policy Settings

Enabled ☑

Notify on Status Change: Completed, In Flight

Policy Groups

Name and Description	Action
Anita.Kennedy Anita Kennedy	
Barry Barry Mills	
Frank Frank Taylor	

Notify Corp Plan policy

Included Actions: Add, Insert

□ All

For more information, see Creating and Enabling Notification Policies.

Creating and Enabling Notification Policies

You create and enable notification policies on the Policies tab of the inspector for applications, dimensions, hierarchy sets, and node types. You must enable the policy on the data chain object in order to send notifications for that object. You can create additional notification policies that have different filters to handle conditional notifications of specific data sets by different users.

To create and enable a notification policy:

1. Open the object inspector for the data object. See, for example:
   - Inspecting Applications
   - Inspecting Dimensions
   - Inspecting a Hierarchy Set
   - Inspecting a Node Type
2. On the **Policies** tab, click **Create**.
3. From **Policy Type**, select **Notify** and enter a name and optional description for the policy.
4. Click **Create**.
5. On the **Definition** tab, click **Edit** and define the policy settings:
   - Select the **Enabled** checkbox to enable the policy. By default this is set to **False**.
   - In **Notify on Status Change**, select one or both of these statuses. By default, both statuses are selected. Click the **X** to remove a status.
     - **In Flight**: notifies policy users when a request is submitted.
     - **Consolidated**: notifies policy users when a request is consolidated. See Understanding Request Consolidation.
     - **Completed**: notifies policy users when a request is closed.
   - In **Policy Groups**, click the **Add a policy group** drop down list to add users and groups to the notify list.

     > **Note:**
     >
     > Groups that do not currently contain any users are indicated by a ⚠ icon. Service Administrators can assign users to the groups in Access Control. See Overview of Access Control in *Administering Access Control*.

     > **Note:**
     >
     > The users and groups in this list are granted *Participant (Read)* permission on the data objects in the policy.
6. Click **Save**.
7. (Optional) On the Filters tab, click **Edit**, and define action filters and node conditions for the policy.
   - In **Included Actions**, clear the **All** check box and, from the **Included Actions** list, select actions to add. Action filters enable you to set up notifications for only the actions that you select.
   - In **Properties**, perform an action:
     - Select **All** to notify users whenever any properties are updated.
     - Select **Include** and then select specific properties to notify users when those properties are updated.

  - Select **Exclude** and then select specific properties to notify users when any property *except* the ones that you selected are updated.
  - Select **None** to specify that users should not be notified based on property updates.

> **Note:**
>
> The request item base action does not have to be Update in order to use the Properties filter. For example, if your Properties filter includes the Core.Description property, then any update to the description property will qualify for the property filter, even if the base request action was an Add or an Insert.

- In **Node Condition**, click **Define Expression** *fx* to launch the expression builder. Because you are evaluating nodes in the viewpoint where the action was performed, you use the `node` object in the expression builder.

  The expression builder guides you to create an expression that evaluates to a Boolean value only. See Building Expressions.
- In **Request Types**, select the type of requests that this policies applies to, or select the **All** check box to apply the policy to all request types. See Request Types and Process Flow.
  - Consolidation
  - Import
  - Interactive
  - Load
  - Subscription
- Click **Save**.

Understanding Policy Order

You can assign priorities to approval and commit policies by specifying the order in which the policies are executed. When a request is approved or committed, policies with the same order are fulfilled as a group before moving on to the next group.

For example, you can assign policy orders so that enrichment policies take place before approval policies, or node type policies are enforced before dimension policies.

You specify the policy order on the Definition tab of an approval or commit policy. (See Creating and Enabling Approval Policies).

Policy Order Processing

When a request with multiple policies enters the approval or commit stages, the following processing takes place:

1. Approvals (or commits, in the Commit stage) for all policies with a policy order with the lowest order (for example, 1) are processed as a group. Invites to approve (or commit) are sent to all invitees in all of the policies that group.

2. When all of the policies with the lowest policy order have been fulfilled, policies with the next highest sequential policy order (for example, 2) are processed as a group. Invites are sent to all invitees in the current group, as well as any unfulfilled policies in any earlier groups (to account for an unfulfilled policy being modified to a lower number while the request is being processed).

   > **Note:**
   >
   > If a current invitee has already provided an approval or commit earlier in the policy order, that user auto-approves for each of their policies in the current group.

3. Policies are processed in groups by policy order number until no policies remain unfulfilled.
4. The request moves to the next stage (for example, if the request was in the Approval stage and there are Commit policies, the request is moved to the Commit stage).
5. Policies with the lowest number policy order in the Commit stage are processed as a group, and so on.
6. When there are no stages remaining, the request is attempted to be completed and closed.

Request Enrichment and Policy Order

If a request is enriched in a way that activates a policy, the new policy will be included in the next workflow cycle according to its policy order. In other words, if a request is enriched with data that is affected by policies in a group earlier than the current group (for example, policies that were not initially activated or policies that have already been approved), invitees for all of the policies in that earlier group are invited to approve the request along with the invitees in the current group.

As an example, suppose you had three groups of policies ordered 1, 2 ,and 3:

1. A request is submitted with data that is affected by the policies in groups 1 and 3, but not 2.
2. The policies in group 1 are fulfilled, but during the processing of group 3, the request is enriched to include data that is affected by the policies in group 2.
3. Invitees in group 2 are included along with the current invitees in group 3. All policies from groups 2 and 3 must now be fulfilled before the request can transition to the next stage.

Request Pushback, Recall, and Withdrawing Approvals

The request stage changes resulting from pushback, recall, and withdrawing approvals is handled as follows:

- For request pushback or recall, all approvals are cleared. When the request enters the approval or commit stage again, the policy order begins again at the lowest order.
- When approvals are withdrawn from a request, approvals that occurred before the withdrawn approval are retained, and approvals that occurred after the withdrawn approval are cleared. For example, if an approval was withdrawn in policy group order 2, approvals from policy group order 1 are retained and approvals from policy group order 3 are cleared. For approvals in the same policy group as the withdrawn approval:
  - If the withdrawn approval is in a serial policy, approvals after the withdrawn approval are cleared.

  - Approvals before the withdrawn approval in a serial policy and all approvals in parallel policies are retained.

Viewing the Policy Execution Plan

The Policy tab of the request inspector displays an execution plan of the policies on a request. See Policy Execution Plan.

Understanding Request Delegation

When request approvers and committers mark themselves as out of the office to take themselves out of the request workflow (see Setting Your Preferences), they can assign a request delegate to approve or commit requests for them.

The delegate must have the appropriate permissions to view content and approve or commit requests (see Security for Requests). The system does not automatically assign them the necessary permissions to view and take action on requests. If the delegate has permissions on the data objects in a request but not the view, they can approve the request using the request inspector.

Request delegates are handled as follows:

- When the invitation to approve or commit a request is sent to the approver or committer, an invitation is also sent to the delegate.
- The delegate can perform the approve and commit tasks on behalf of the out of office user.

> **Note:**
>
> Both the original invitee and the delegate remain invited to the request, but only one of them can approve. When one of them approves, the other is removed from the invitee list. However, the system marks the out of office invitee as a request participant so that they can access the request at a later point in time.

- When the delegate approves or commits a request, the invitee table on the Workflow tab in the request inspector displays the user details on whose behalf the delegate was invited. See Inspecting Requests.
- If a delegate is the same user for multiple out of office approvers or committers, or if the delegate is also an invitee on the request, the delegate is sent only one invitation to approve or commit.
- If the delegate is also out of the office, the request is escalated to a user with *Data Manager* permission to break the deadlock.

Policy Reminders and Escalations

When a request is submitted on a data object that has an approval or a commit policy enabled on it, notification emails are sent to the invitees to approve or commit the request. You can also configure the policy so that reminder and escalation emails are sent if an invitee does not take action on a request or if an escalation is needed.

> **Note:**
>
> Reminder and escalation emails are only sent if you have configured these settings in the policy.

Reminders

After a request is submitted and notifications are sent to invitees, reminder emails can be sent if an invitee has not taken action after 24 hours. You can specify how many reminders will be sent before the request gets escalated. See Creating and Enabling Approval Policies or Creating and Enabling Commit Policies.

Escalations

Requests are escalated for resolution in these scenarios:

- **Timeout**: When an invitee has not responded to the number of reminder emails that you specified in the **Approval Escalation** or **Commit Escalation** field when you configured the policy.

  > **Note:**
  >
  > If escalation is set to zero, requests will not be escalated for timeouts.

- **Deadlock**: When there are not enough policy groups or users available to satisfy a policy requirement. For example, if a user has been removed from a group and there aren't enough remaining users to satisfy the **Total Required** setting for that policy, the request is deadlocked.
- **Management Hierarchy Escalations**: There are several scenarios when a request with a management hierarchy approval policy would get escalated, such as when the invitee does not have at least *Participant (Read)* permission on the viewpoint associated with the policy. For the full list, see Request Escalation in "Understanding Management Hierarchy Approvals".

In these scenarios, the request is escalated to users that have *Data Manager* permission to the data objects in the affected policy, and notifications are sent to those users so that they can intervene and resolve the timeout or conflict.

> **Note:**
>
> Because the *Owner* permission includes the *Data Manager* permission, requests are escalated to users with *Owner* permission on the data objects as well.
> If a user is invited to a request because of an escalation, that is information indicated on the Workflow and Policies tabs of the request inspector (see Inspecting Requests).

After a user with *Data Manager* permission takes action on the request, the normal request workflow is resumed.

For example, suppose you have a serial approval policy in which someone from Accounting has to approve, then Barry, and then Jane. A user in Accounting approves the policy, but Barry is out of the office and is not available to approve the request. In this case, after the reminder emails are sent to Barry, the request is then escalated to users with *Data Manager* permission to the data objects in the request. After one user with *Data Manager* permission approves the policy, it then moves to Jane for her approval.

> **Note:**
>
> If a request is escalated multiple times, a user with *Data Manager* permission has to approve the request only once.

If a request contains items that are affected by multiple approval policies and one of the policies is escalated, the other policies are not affected. For example, if a request has items from an Account dimension that has a policy that is escalated after five reminders and items from a Cost Center dimension that has a policy that is escalated after two reminders, after the second reminder the request is escalated to users with *Data Manager* permission on the Cost Center dimension only. Users with *Data Manager* permission on the Account dimension will not receive escalation notices until after the fifth reminder is sent.

Handling Out of Office Status for Approvers and Committers

Approvers and committers can mark themselves as out of the office to take themselves out of the request workflow. (See Setting Your Preferences). If they have specified a request delegate to approve or commit requests for them, invitations are sent to the delegate as well. (see Understanding Request Delegation). If they have not designated a request delegate, the out of office status is handled as follows:

- Approvers and committers who are out of the office are still invited to approve requests. However, they are not included in the remaining approvers or committers count for the policy. If the out of office user has assigned a request delegate, that delegate is counted towards the remaining count for the policy.
- If the remaining approvers or committers are not enough to satisfy the policy, the request is automatically escalated to a user with *Data Manager* permission to break the deadlock.
- The out of office approver or committer is still invited to approve or commit the request and can take action on it when they return to the office. If the out of office user has assigned a request delegate, both the original invitee and the request delegate are invited to approver or commit the request, but when one of them approves, the other is removed from the invitee list. See Understanding Request Delegation.

Modifying Policies

You can modify approval or commit policies even if there are requests that have already been sent for approval or commit based on that policy. When you modify a policy, a daily scheduled process evaluates in flight requests for that policy and takes appropriate action.

> **Note:**
>
> These changes apply to in flight requests only. They do not affect draft or completed requests, or requests that are pushed back.
>
> If you modify a policy that affects requests that are currently in flight, the Remaining Approvals or Commits and Invitee counts on the Workflow tab of the request inspector may display inaccurate counts until the daily scheduled process runs to synchronize the counts.

Modified Policy Setting	Action Taken
Enabled	• **Changed from Disabled to Enabled**: New requests that use this data object will require approvals or commits. If a request is pending approval or commit for other policies, the users in this newly-enabled policy are invited to approve or commit the request. • **Changed from Enabled to Disabled**: New requests that use this data object will no longer require approvals or commits. If a request is pending approval or commit only for this newly-disabled policy, it is approved or committed. **Note**: You can disable workflow at the application level instead of disabling individual policies. This can be helpful, for example, in a test environment. See Disabling Workflow for Applications.
Approval Groups or **Commit Groups**	The new group is compared to the existing group, and the following actions are taken: • New users who are not in the existing group are added as invitees. • Users who are no longer in the existing group are removed as invitees.
Approval Method, **One Approval Per Group**, **Total Required**, **Include Submitter**	Remaining users who have not yet approved are invited to approve or commit based on the new settings. **Note:** **Commit Method** and **Total Required** cannot be changed for commit policies.
Allow Enrichment During Approval	Remaining users who have not yet approved are able to enrich the request based on their permissions and data access before approving. **Note:** Does not apply to commit policies.
Reminder Notification	Reminder notifications are sent to current invitees based on the new settings.
Approval Escalation	Requests are escalated to users with *Data Manager* permission to the data objects in the request, based on the new settings.

Deleting Policies

Considerations

- You must have *Owner* or *Metadata Manager* permission on the data chain objects in the policy in order to be able to delete it.
- The policy cannot be in use by an in flight or completed request.
- The policy must be disabled before it can be deleted.

To delete a policy:

1. Inspect the data chain object that contains the policy that you want to delete.
2. On the Policies tab, click **Edit**.
3. In the Action column, click **Action** ⋮, and then select **Delete**.
4. Click **Yes** on the confirmation message, and then click **Save**.

30 Integrating Data

Integrating data consists of running imports, exports, extracts, and viewpoint loads.

See the following topics for more information:

Using OAuth2 in Cloud EPM Applications

Application connections and global connections to Cloud EPM applications support using Open Authorization (OAuth) 2 for authentication.

Perform the following steps to set up OAuth2 authentication. See Authentication with OAuth 2 in *REST APIs* for details.

1. Register the OAuth2 client.

   > **Note:**
   >
   > For Oracle Fusion Cloud Enterprise Data Management, you must register separate OAuth2 clients (each with a client ID, access, and refresh tokens) for each application or global connection. For example, if you have two application connections and two global connections that you want to use OAuth2 authentication for, you must register four OAuth2 clients.

2. Obtain and securely store the first refresh token.

   > **Tip:**
   >
   > Set the refresh token expiry longer than the frequency of running your imports and exports in order to keep the tokens valid.

3. Obtain an access token from the refresh token.

Creating Application Connections

Application connections enable you to import and export data to external applications.

You create application connections when you register an application. See:

- Registering Planning and FreeForm Applications
- Registering Financial Consolidation and Close Applications
- Registering Enterprise Profitability and Cost Management Applications
- Registering Oracle Financials Cloud General Ledger Applications

You can also inspect an application and navigate to the **Connections** tab in order to create an application connection, modify connection details such as the connection URL or the username and password, or delete a connection. See Inspecting Applications.

Application connections to Oracle Cloud EPM applications support basic and OAuth2 authentication. You must register an OAuth2 client for each application connection. See Using OAuth2 in Cloud EPM Applications.

Application connections to Oracle Financials Cloud General Ledger applications support basic authentication only.

> **Note:**
>
> You cannot use application connections in extracts or extract packages. Instead, you must use a global connection. See Creating Global Connections.

Importing Dimensions

You import data from external application dimensions into Oracle Fusion Cloud Enterprise Data Management which populates nodes, properties and hierarchical relationships in the dimension's registered bound data chain.

From the Applications list, import dimensions:

- Importing Planning and FreeForm Dimensions.
- Importing Financial Consolidation and Close Dimensions
- Importing Enterprise Profitability and Cost Management Dimensions
- Importing Tax Reporting Dimensions
- Importing Oracle Financials Cloud General Ledger Dimensions
- Importing E-Business Suite General Ledger Dimensions
- Importing a User Defined Dimension.

Videos

Your Goal	Watch This Video
Learn about importing and exporting dimensions.	Importing and Exporting Application Data

> **Note:**
>
> All application types support importing and exporting from comma-delimited files.
>
> For Planning and Financial Consolidation and Close applications, if you enter valid connection information to an external application during registration, you can import and export from the external application inbox by using a connection. You can then import that file into your Planning application.
>
> For Oracle Financials Cloud General Ledger applications, if you enter valid connection information to an external application during registration, you can export to the external application server by using a connection. You can then import that file into your Oracle Financials Cloud General Ledger application.

Import Considerations

- To import, you need at least the *Data Manager* permission on the application or the dimension that you are importing to.
- You can import to dimensions with an Active status only.
- You cannot import to unbound or lookup dimensions.
- You must first register the dimension you want to import.
- For all import modes, the import source file should contain the full data set for a viewpoint. If you want to perform an incremental update where the incoming source file contains only changes to a viewpoint, use the request file upload instead of import. See Making Changes Using a Load File.
- For imports in Replace or Reset mode, the data that you import into Cloud EDM does not have to pass all system, application-specific, and custom validations before you import it. By design, imports enable you to bring data into Cloud EDM and then cleanse it. For example, you can bring in data that fails any of these validations:
  - String values that exceed the maximum allowed length for a property
  - Integer or float values that are outside of the allowed values for minimum or maximum values
  - Data that fails custom or application-specific validations
  - Node and node list data type values that do not reference an existing node in a node set

  > **Tip:**
  >
  > After you bring this data in, you should validate it (for example, by running viewpoint validations) in order to cleanse it. See Validating a Viewpoint.

  For imports in Merge mode, validations prevent you from importing data that does not pass all system, application-specific, and custom validations.
- For imports in Replace or Reset mode, you can import properties that are marked as read-only in Cloud EDM. This enables you to bring in reference data from an external system that cannot then be changed in Cloud EDM.

For imports in Merge mode, validations prevent you from importing properties that are read-only in Cloud EDM.

- For Planning, Planning Modules, and Financial Consolidation and Close applications, you can also import attribute dimensions. You must first register the attribute dimension. See Understanding Attributes.
- When you import to an existing dimension, data in the dimension's bound data chain is affected. Unbound objects are not updated by the import process; however, they may be indirectly affected by changes to other objects in their data chain
- For a hierarchy format, nodes and hierarchical relationships are imported and displayed based on the grouping and sorting options on the hierarchy set. You can adjust the sorting of parent and bottom nodes to match the external application by opening the hierarchy set and using the **Use Custom Order**, **Group Parent Nodes First**, and **Sort Bottom Nodes By** settings to determine the sort order for parent and bottom nodes before you run the import. See Creating Hierarchy Sets.

  > **Note:**
  >
  > For Merge Imports, enable **Process Reorders** to set the order of the nodes in the viewpoint being imported into to the order in which they appear in the import source file. See Managing Import Options.

- When you import to an existing dimension in Merge or Replace mode, the system checks the node type of the nodes that you are importing, as follows:
  - If the node already exists in the bound dimension, the system finds and uses the existing node and node type when it runs the import.
  - If the node does not already exist in the bound dimension, the system imports the node using the node type that was created during the initial registration.

  > **Note:**
  >
  > There are additional considerations for node types with imports in Universal applications. See Importing a User Defined Dimension.

- Import files should be encoded in UTF-8.

  > **Tip:**
  >
  > To preserve Unicode characters, save your import file with the following option in Excel: `CSV UTF-8 (Comma delimited) (*.csv)`.

Import Modes

When you import data, there are three import modes that you can select. The following table describes the import modes.

Import Mode	Hierarchy Viewpoint Updates	List Viewpoint Updates
Merge	• Adds nodes that are in the import source but not in the bound viewpoint • Updates node property values that have different values in the import source and the bound viewpoint. • For nodes that have different parents in the import source than the bound viewpoint: – If shared nodes are enabled, nodes are inserted. – If shared nodes are not enabled, nodes are moved. • Removes nodes that are in the bound viewpoint but not in the import source. • Depending on your merge import options, deletes nodes that are in the bound viewpoint but not in the import source. See Managing Import Options. • For relationship property values: – If the relationship property value exists in the import source but not the bound viewpoint, the relationship property value is updated. – If the relationship property values are the same in the import source and the bound viewpoint, no update takes place. – If the import source does not contain a relationship property value, no update takes place. • Select **Process Reorders** to reorder the nodes to the order in the import file. See Managing Import Options. **Note:** Reorder actions in Merge imports are supported only with inserts or adds.	• Adds nodes that are in the import source but not in the bound viewpoint • Updates node property values that have different values in the import source and the bound viewpoint. • Nodes are displayed in descending order by the date that they were created or updated.

Import Mode	Hierarchy Viewpoint Updates	List Viewpoint Updates
Replace	• Nodes are imported and the hierarchy is recreated as follows: – Hierarchy relationships that are in the bound viewpoint and the import source are retained. – Hierarchy relationships that are in the bound viewpoint but not the import source are deleted. • For existing relationship properties: – If the import file contains both the relationship and the relationship property value, the existing value is overridden with the value from the import. – If the import file contains the relationship but not the relationship property value, the existing value is retained. This enables property values that were defined for these relationships directly in Cloud EDM to remain in the application. – If the import file does not contain the relationship, the relationship property is deleted. **Caution:** Ensure that your import file contains your full hierarchy set before you run the import.	• Adds nodes that are in the import source but not in the bound viewpoint • Updates node property values that have different values in the import source and the bound viewpoint. • Nodes are displayed in descending order by the date that they were created or updated.
Reset	• Clears all nodes in all bound node types in the dimension • Clears all relationships for all bound hierarchy sets in the dimension • Re-imports nodes and rebuilds relationships	• Clears existing bound nodes and imports nodes • Nodes are imported and displayed in the order in which they are listed in the import file or external application.

Note:

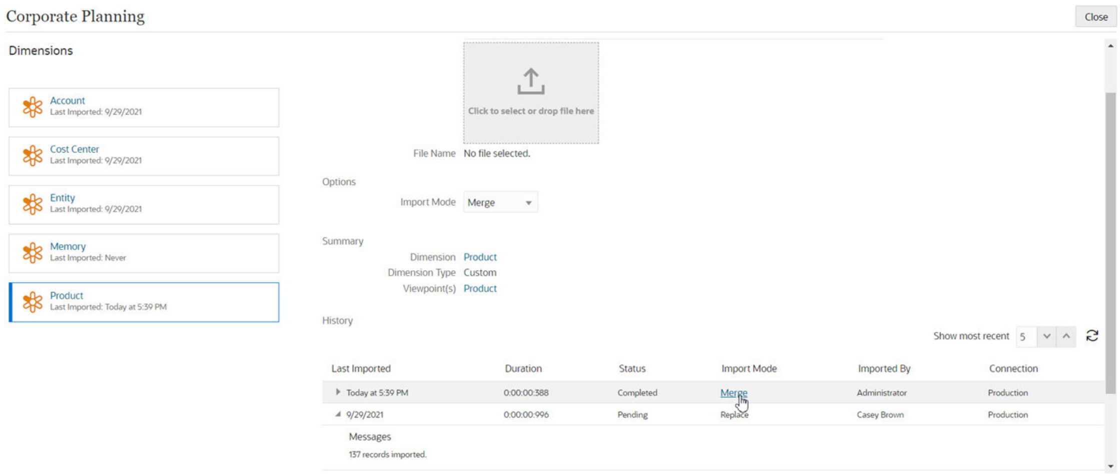
The first time you import into a dimension, you must use the Reset import mode. You cannot use a Merge or Replace import in an empty dimension.

Reviewing Import History

On the import screen, review the **History** section to see the dates and times of the most recent imports that were run for this dimension. By default, the last five are displayed. You can adjust the number of imports to display (up to 50) in the **Show most recent** field. Click ⟳ to update the table.

The table displays how long it took to run each import, the status and mode of the import, the user who ran it, and the name of the connection that was used. Imports from a file will display "None" for the connection.

Click the expand arrow on a row to view the number of records that were imported or any errors that were encountered during the import process. For Merge imports, click the link in Import Mode column to inspect the request that was generated from the import.



Corporate Planning

Dimensions

- Account — Last Imported: 9/29/2021
- Cost Center — Last Imported: 9/29/2021
- Entity — Last Imported: 9/29/2021
- Memory — Last Imported: Never
- Product — Last Imported: Today at 5:39 PM

File Name No file selected.

Options

Import Mode Merge

Summary

Dimension Product
Dimension Type Custom
Viewpoint(s) Product

History

Show most recent 5

Last Imported	Duration	Status	Import Mode	Imported By	Connection
Today at 5:39 PM	0:00:00:388	Completed	Merge	Administrator	Production
9/29/2021	0:00:00:996	Pending	Replace	Casey Brown	Production

Messages
137 records imported.

Working with Merge Imports

Merge imports capture changes in an import request instead of importing the data directly (as with Replace and Reset). This enables you to identify the incremental changes that were made in the import, view the import changes in transaction history, and synchronize import changes across viewpoints by using subscriptions.

When you run a Merge import, the data is imported following this process:

1. The import source data is compared to existing data to identify differences.
2. A draft request for the view of the bound viewpoints is created that contains request items and actions from the import differences.
   If there are no differences, the import request is not created and the user is notified.

   > **Note:**
   >
   > The request item limit for merge imports is 20,000 items.

3. Predefined validations are run on the request:

- By default, if no validation errors are found, the request is automatically submitted. You can deselect **Auto-Submit** to prevent the request from being automatically submitted. See Managing Import Options.
- If validation errors are found, the request remains in Draft status in the Submit stage and is assigned to the user who ran the import. Email notification is sent to the user who is assigned to the request.

> **Note:**
>
> For Planning and Financial Consolidation and Close imports, the `Cannot add a Predefined Member` and `Cannot change a Predefined Member property` validations are not run.

4. By default, the Approve and Commit stages are bypassed, request items are committed, and the request is set to Closed. Deselect **Bypass Approvals and Commits** to prevent the stages from being bypassed. See Managing Import Options.
5. If the import contains request actions for a source viewpoint of a subscription, subscription requests are generated for the target viewpoints. Subscriptions for the same target view have their request items grouped into the same subscription request. See Subscribing to Viewpoints.

After the import is completed, you can click the import request link to navigate to the import request. If the request is in Submit or Approve stage, it is displayed in the request view. If it is closed, it is displayed in the inspector. You can view the import results in Transaction History.

Managing Import Options

Import options enable you to specify at the dimension level how empty incoming values are handled, how clearing data on import is managed, and how deletes and reorders are processed during merge imports.

To manage the import options for a dimension, inspect the dimension that you want to import to (see Inspecting Dimensions) and then select the Import/Export tab.

Importing Empty Values

For properties with a data type that supports empty values (see Understanding Null and Empty Values in Properties), the **Ignore Empty Fields on Import** field determines how empty values are handled. When importing a file that does not contain a value for a property:

- If **Ignore Empty Fields on Import** is enabled, then no value is imported for the property. (The value for the property will be null if it does not inherit a value and there is no defined or default value.)
- If **Ignore Empty Fields on Import** is not enabled:
  - For property data types that support defined empty values (such as String, List, Memo, or Numeric String): A defined empty value is stored for the property.
  - For property data types that do not support a defined empty value (such as Boolean, Date, Float, or Integer): The import will display a warning in the messages section of import history and no value will be imported for that property.

Clearing Values on Import

The **Clear Value Keyword for Import** setting enables you to define a keyword that will clear the value for a property when a file with that keyword is imported. For example, if you enter `<none>` in this field, when you import a file that contains `<none>` in the field for a property, the existing value for that property will be cleared. (The value for the property will be null if it does not inherit a value and there is no default value.)

It is a best practice to always use a clear value keyword in your import file to clear out existing values. Disabling **Ignore Empty Fields on Import** and then importing blank values for a property will not overwrite the property values with blank values if the property data type doesn't support blank values. You can't, for example, overwrite existing Date values with blank values by importing blanks.

Expanding Allowed Values on Import

For Reset and Replace import modes only, select **Expand Allowed Values on Import** to add new distinct allowed values to properties with string and list data types when you run an import. If there are currently allowed values at the node type override level, the imported allowed values are added to them. Otherwise, the new allowed values are added at the application override level.

> **Note:**
>
> The **Expand Allowed Values on Import** option is not available for Merge imports or dynamic allowed values lists.

Merge Import Options

Deleting or Reordering Nodes during Merge Imports

By default, merge imports do not delete nodes from the dimension that you are importing to. If a node exists in the bound dimension but not the import file, the node is removed from the hierarchy set that you are importing into, but it is not deleted from the node set. You can choose to delete all nodes in the bound dimension that are not in the import file, or you can delete only the nodes in the bound dimension that are not referenced by any other hierarchy node sets in the dimension. You can also choose to reorder the nodes in your dimension based on the order of the nodes in your import file.

To specify how deleting and reordering nodes is handled during merge imports:

1. In the **Merge Import Options** section, for **Process Deletes**, select from these options:

Mode	Description	Use Case
Disabled (Default setting)	Removes nodes from a bound viewpoint if they do not exist in the import source. No nodes are deleted from any bound node type.	Do not delete any nodes that are in the bound viewpoint but not the import source.
Safe Mode	Removes nodes from a bound viewpoint if they do not exist in the import source. The removed nodes are also deleted from bound node types only if the nodes are not used in any other hierarchy node set or hierarchy set for the dimension.	Prevent removed nodes used in other hierarchy sets or hierarchy node sets from being deleted.

Mode	Description	Use Case
Forced Mode	Deletes nodes from a bound viewpoint if they do not exist in the import source. These nodes are deleted from node types used by any node set and hierarchy set in the dimension.	Delete nodes from a bound viewpoint which are no longer in the import source regardless of whether they are used in another hierarchy set or hierarchy node set for the dimension.

2. Select **Process Reorders** to set the order of the nodes in the viewpoint being imported into to the order in which they appear in the import source file.

> **Caution:**
>
> This option may generate a large number of request items if there are significant differences between the existing order and the order in the import file.

3. Click **Save**.

Configuring Workflow Options for Merge Imports

By default, if there are no validation errors on a merge import request it is auto-submitted and it bypasses any approval or commit policies. Use the following options to adjust the default behavior:

- Deselect **Auto-Submit** to prevent merge import requests from being auto-submitted
- Deselect **Bypass Approvals and Commits** to prevent auto-submitted merge import requests from automatically bypassing all approval and commit policies. (This option is cleared automatically if you deselect the **Auto-Submit** field).

Exporting Dimensions

You can export from a viewpoint that is bound to a dimension back to an external application. Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States, and all states. If the node set's top node is Texas, only Texas and the nodes under it are exported.

Best Practice

It's a best practice to validate the viewpoint bound to the dimension prior to export to ensure data integrity. Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

> **Note:**
>
> All system, application-level, and custom validations as well as all constraints are run when you use the **Validate Before Export** option. See Understanding Validations and Constraints.

From the Applications list, export dimensions:

- Exporting Planning and FreeForm Dimensions
- Exporting Oracle Financials Cloud General Ledger Dimensions

- Exporting Enterprise Profitability and Cost Management Dimensions
- Exporting Tax Reporting Dimensions
- Exporting E-Business Suite General Ledger Dimensions
- Exporting Financial Consolidation and Close Dimensions
- Exporting a User Defined Dimension.

Videos

Your Goal	Watch This Video
Learn about importing and exporting dimensions.	Importing and Exporting Application Data

Considerations

- All application types support importing and exporting from comma-delimited files.
- For Planning applications, if you enter valid connection information to an external application during registration, you can import and export from the external application inbox by using a connection. You can then import that file into your Planning application.
- For Oracle Financials Cloud General Ledger applications, if you enter valid connection information to an external application during registration, you can export to the external application server by using a connection. You can then import that file into your Oracle Financials Cloud General Ledger application.
- Export options are available in Oracle Fusion Cloud Enterprise Data Management to include additional post-export tasks that initiate the import and publishing processes in the external Financials Cloud environment and complete the synchronization of segment values and trees. See Managing Export Options.

Before you begin:

- To export, you need at least the *Data Manager* permission on the application or dimension that you are exporting from.
- You can export from dimensions with an Active status only.
- You cannot export from unbound or lookup dimensions.
- For Planning, Planning Modules, and Financial Consolidation and Close applications, you can also export attribute dimensions. See Understanding Attributes.
- The sort order for nodes and hierarchical relationships in exports is based on the grouping and sorting options on the hierarchy set. See Creating Hierarchy Sets.
- Export files are encoded in UTF-8.

> **Tip:**
>
> For all application types except Oracle Financials Cloud General Ledger applications, you can change the order of the columns in your export file and control which columns are exported by editing the binding keys for the dimension that you are exporting. See Editing Binding Keys.

Reviewing Export History and Downloading Export Files

On the export screen, review the **History** section to see the dates and times of the most recent exports that were run for this dimension. By default, the last five are displayed. You can adjust the number of exports to display (up to 50) in the **Show most recent** field. Click ⟳ to update the table.

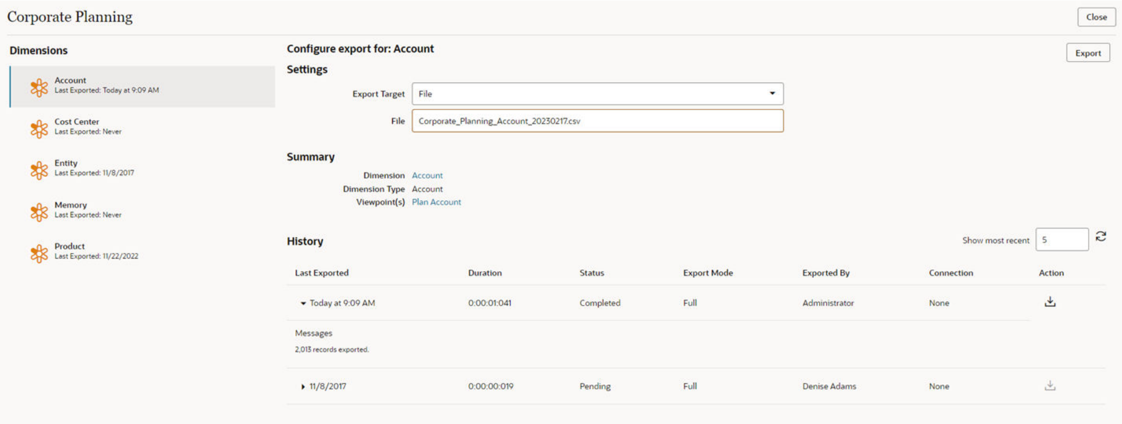
The table displays how long it took to run each export, the status and mode of the export, the user who ran it, and the name of the connection that was used. Exports to a file will display "None" for the connection. Click the expand arrow on a row to view details about the export, such as the number of records that were exported, the status of any post-export tasks, or any validation errors.

In the **Action** column, click **Download to File** ⤓ to download the export file from the server. This can be useful if your browser times out during long exports or to compare export results to previous export files. You can download export files that were run to file only. If the export was run to a connection, it is not available to download from the Export screen.

If the file is no longer available on the server, a message is displayed.

> **Note:**
>
> Exports that were run from a REST API are not available to be downloaded from the export screen. Use EPMAutomate or a REST API to download those export files from the staging area. See Download a File from the Staging Area in *REST API for Cloud EDM*.

For Oracle Financials Cloud General Ledger exports to a connection, if you have enabled any post-export tasks in the Financials Cloud environment (such as processing account hierarchies), the status of those tasks is also displayed. See Managing Export Options.



Corporate Planning

Dimensions

Account
Last Exported: Today at 9:09 AM

Cost Center
Last Exported: Never

Entity
Last Exported: 11/8/2017

Memory
Last Exported: Never

Product
Last Exported: 11/22/2022

Configure export for: Account

Settings

Export Target: File

File: Corporate_Planning_Account_20230217.csv

Summary

Dimension: Account
Dimension Type: Account
Viewpoint(s): Plan Account

History

Show most recent: 5

Last Exported	Duration	Status	Export Mode	Exported By	Connection	Action
▾ Today at 9:09 AM	0:00:01:041	Completed	Full	Administrator	None	⤓
Messages 2,013 records exported.						
▸ 11/8/2017	0:00:00:019	Pending	Full	Denise Adams	None	⤓

Managing Export Options

Export options enable you to define keywords to export instead of null or empty values, as well as how validations are run before exporting.

To manage the export options for a dimension, inspect the dimension that you want to export from (see Inspecting Dimensions) and then select the Import/Export tab.

Exporting Keywords Instead of Null and Empty Values

You can export keywords instead of null and empty values for properties. When you export a property that has a null or an empty value, the keyword that you specified in **Export Keyword for Null Values** or **Export Keyword for Empty Strings** (for example, `<none>`) is exported for that property instead of an actual blank value.

> **Note:**
>
> For Planning, Planning Modules, and Financial Consolidation and Close applications, the default keyword for null and empty values is `<none>`. You can change this keyword, but the target application may not recognize the keyword as a null or empty value. The keyword that you specify is applied only to the properties that support it in the target application:
>
> - Aliases
> - Attributes
> - Description (unless it is a shared node)
> - Formula (always exports the keyword if Data Storage is Shared)
> - Formula Description (always exports the keyword if Data Storage is Shared. If it is not shared then it will export a blank value if defined or the keyword if default is blank)
> - Smart List (unless it is a shared node)
> - UDA
>
> Additionally, the null or empty value keyword is not exported for any predefined members.

> **Tip:**
>
> If the application that you are exporting to does not differentiate between null and empty values, you should use the same keyword for both **Export Keyword for Null Values** and **Export Keyword for Empty Strings**.

Validating Viewpoints before Exporting Dimensions

To ensure data integrity, you can use the **Validate Before Export** option to specify whether or not the viewpoint for a dimension gets automatically validated when you export the dimension or mapping. Choose from these options:

- **Disabled** (Default): Validations are not run before export.
- **Alert on Error**: Validations for the bound viewpoints for the dimension with a severity of Error (see Configuring Validation Enforcement and Severity) are run before the export is performed. If there are validation failures, the user is alerted but the export will still be performed. Validation failures and export results are displayed in the History section of the export screen.

- **Discontinue on Error**: Validations for the bound viewpoints for the dimension with a severity of Error (see Configuring Validation Enforcement and Severity) are run before the export is performed. If there are validation failures, the user is alerted and the export is discontinued. Validation failures and export results are displayed in the History section of the export screen.

> **Note:**
>
> All system, application-level, and custom validations as well as all constraints are run when **Validate Before Export** is performed. See Understanding Validations and Constraints.

If there are export validation issues, click the **Viewpoint(s)** link to open the viewpoint and resolve the issues. See Resolving Validation Issues.

When you have finished resolving the validation issues, from the viewpoint pane click **Back to Export** to return to the export screen.

Oracle Financials Cloud General Ledger Export Options

For Oracle Financials Cloud General Ledger dimension exports to a connection only, you can configure export options to include additional post-export tasks that initiate the import and publishing processes in the external Financials Cloud environment and complete the synchronization of segment values and trees. After you enable the export options, they are executed each time you export the dimension to an Oracle Financials Cloud General Ledger connection.

The following options are available:

- **Import Segment Values and Hierarchies**: Updates the Financials Cloud segment values and hierarchies in the external application with the values being exported.

  > **Note:**
  >
  > The user running the export must have the `GL_LOAD_ACCOUNT_VALUES_AND_HIERARCHIES_PRIV` privilege in Oracle ERP Cloud.

- **Process Segment Values and Hierarchies**: Submits processes in the external system related to account hierarchies, such as row and column flattening, tree audit and activation, maintain value sets and account hierarchies processes, and publish hierarchies process.

  > **Note:**
  >
  > The user running the export must have the following privileges in Oracle ERP Cloud:
  >
  > – `GL_LOAD_ACCOUNT_VALUES_AND_HIERARCHIES_PRIV`
  >
  > – `FND_APP_MANAGE_FLEXFIELD_VALUE_SET_PRIV`

- **Publish Hierarchies to Cube**: Creates and updates chart of accounts dimension members and hierarchies for the balances cubes.

> **Note:**
>
> The user running the export must have the following privileges in Oracle ERP Cloud:
>
> – `GL_LOAD_ACCOUNT_VALUES_AND_HIERARCHIES_PRIV`
>
> – `FND_APP_MANAGE_FLEXFIELD_VALUE_SET_PRIV`

> **Note:**
>
> For the **Process Segment Values and Hierarchies** and **Publish Hierarchies to Cube** options, the integration tasks are executed in Oracle Financials Cloud General Ledger only on the trees and tree versions that were exported from Oracle Fusion Cloud Enterprise Data Management.

By default, all export options are deselected, and only the first option can be selected. When you select each option, the next option becomes available to be selected. For example, you must select **Import Segment Values and Hierarchies** before you can select **Process Segment Values and Hierarchies**.

If you deselect an option, all selected options below that option are cleared.

Incremental Exports

Incremental exports enable you to export new or updated nodes from a specified date and time or export type. They are available as REST API methods for list viewpoints in certain application types only.

Considerations

- Use the Incremental Export for a Dimension REST API method to run incremental exports.
- Incremental exports are available for list viewpoints in these application types:
  - Oracle Financials Cloud General Ledger
  - Oracle E-Business Suite General Ledger
  - Universal
- You can run the export to a file for the above application types. For Oracle Financials Cloud General Ledger applications only, you can also run the export to a connection.
- The export file format is based on the application type's export format.

You can define your incremental export by using these settings:

Table 30-1 Incremental Export Settings

Setting	Description
`nodeChangeTypes`	The type of node changes that you want to export: • **New**: Only nodes added after the date and time that you specify are exported. • **Updated**: Only nodes updated after the date and time that you specify are exported. **Note:** You can select both **New** and **Updated** in the same export.
`since`	The starting timestamp (in Epoch time) that you want to export from.
`sinceLastExportOfType`	Specify an export type (**Full** or **Incremental**). Nodes are exported that were added or updated since the specified export type was last run for this binding. **Note:** You must specify either `since` or `sinceLastExportOfType`.

Exporting Mapping Data

Exporting mapping data is designed to work with Data Management for Enterprise Performance Management systems, such as:

- Planning, Planning Modules, and FreeForm target dimensions
- Financial Consolidation and Close target dimensions
- Enterprise Profitability and Cost Management target dimensions

You can also export mapping data for Universal target dimensions.

Videos

Your Goal	Watch This Video
Learn about mapping data across applications.	Mapping Data Across Applications

Before you begin:

- Ensure that you have the *Data Manager* permission on the dimension that you are exporting from.
- Ensure that the dimension that you are exporting from has an Active status.

- Ensure one or more mapping keys are defined for your source and target nodes in the mapping hierarchy. You will need the location name from the mapping key, see Defining Mapping Keys.
- If you are using a Universal application you can export the mapping data to a file only.
- If you are using Planning, Planning Modules, FreeForm, or Financial Consolidation and Close you can export the mapping data to:
  - A file
  - The Data Management inbox, using the external application connection.
    You can view connection settings by inspecting the application. From **Applications**, scroll to your application, click ⋮, then select **Inspect**, and then select **Connection**.

    After the export, you can use Data Management , within your external application, to import the mapping data, see Importing Member Mappings in *Administering Data Management* .

You can export mapping data for one or more mapping keys. For example, if you mapped two source applications to one target application and defined two mapping keys, export the mapping data twice using the two mapping key location names. To export mapping data:

1. From **Applications**, find your application, click ⋮, and then select **Export Mapping**.
2. From **Location**, select a location name defined in your mapping keys for the selected dimension.

   You need to have already set up mapping keys for the dimension, see Defining Mapping Keys.
3. Select the target dimension.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:
   - If you selected **Connection**, select a connection to export to, and specify an export file name.
   - If you selected **File**, a file name is generated. The default file name is *Application Name_Dimension Name_Date_Location Name*.csv, for example Planning_Account_20200123_loc1.csv, and can be edited before running the export.
5. Click **Export**.

You can also export dimensions using EPM Automate (see exportDimensionMapping in *Working with EPM Automate*), a REST API (see Export Dimension Mappings in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

Mapping Data File Formats

The exported data file from a Universal application dimension is a .csv file with these columns:

- Source node name
- Target node name
- A column that contains `True` or `False` taken from the `Core.Change.Sign` property in the source node type. If the `Core.Change.Sign` property is not used, this column defaults to False.

The exported data file from a Planning application dimension is a .csv file with these columns:

- Source—Name of source node

- Target—Name of target node

> **Note:**
>
> If the source node has the `Core.Change.Sign` property, it is applied to the target node. For example, if `Core.Change.Sign` is `True`, a negative sign is applied to the target node.

- Rule Name—Blank
- Rule Description—Blank
- Load Rule—Blank

For example, if these nodes are in your Planning mapping viewpoint and

The source node is: `CC11100`

The `Core.Change.Sign` property on the source node is set to `True`

The target node is: `100`

The mapping export row will be: `CC111000,-100 , , ,` the columns are separated by commas.

After the export, navigate to your external application and import the mapping data. Within your external application, open the Data Management module and import the mapping data. During the load the negative sign is applied to values in the target node for the selected mapping key which is the location in Data Management.

Creating Global Connections

Global connections enable you to share data with external applications using extracts.

The following connection types are supported:

- **Oracle Cloud EPM** connections send data to an Enterprise Performance Management inbox for use by a metadata import process.
- **Oracle Cloud ERP** connections send data to a Universal Content Management (UCM) document account for use by an File Based Data Import (FBDI) process.
- **Oracle Object Storage** connections serve as technology adapters for more general purpose external systems.

Use Cases

- Global connections to Oracle Cloud ERP can be used to transfer data such as account combinations, mapping sets, and related value sets for import into Financials Cloud.
- Global connections to Oracle Cloud EPM can be used in the following ways:
  - Universal applications that manage the Account Reconciliation business process can use global connections to send profiles to the application inbox.
  - Universal applications that manage dimensions for the Profitability and Cost Management business process can use global connections to send metadata files to the profitinbox.
  - To send mapping files to a Data Management inbox

> **Note:**
>
> Global connections are not used for application imports and exports. Instead, the application connection information that you specify when registering the application is used.

You can restrict connection types so that they can be used only by specific applications in Oracle Fusion Cloud Enterprise Data Management.

Creating a Global Connection

> **Note:**
>
> You must be a Service Administrator in order to create global connections.

Considerations

- After you create a global connection, you can use it in an extract. You can use a global connection in multiple extracts, but each extract can use only one global connection.
- You cannot change the connection type after you create a global connection.
- You can use global connections for extracts only. You cannot run a dimension export using a global connection.
- The parameters that you specify for a global connection are determined by the connection type. For example, when setting up a global connection to an Oracle Object Storage, you must specify the Default Bucket.
- Global connections to Oracle Cloud EPM applications support basic and OAuth2 authentication. You must register an OAuth2 client for each global connection. See Using OAuth2 in Cloud EPM Applications.
- Global connections to Oracle Cloud ERP and Oracle Object Storage support basic authentication only.

To create a global connection:

1. From the Connections cluster, click **Create**.
2. Select the connection type:
   - Oracle Cloud EPM
   - Oracle Cloud ERP
   - Oracle Object Storage
3. Enter a name and, optionally, a description for the connection, and then click **Create**. The global connection is displayed in the connection inspector.
4. Click **Edit** and specify the parameters for the connection. The parameters that you specify are dependent on the connection type:
   - **Oracle Cloud EPM parameters**:

Field	Description
Location	The URL for the EPM inbox. For example: `https://epm.mycloud.com` **Caution:** If you are pasting the address from your browser, do not include the context (for example, `/epmcloud`) in your URL.
Authentication Type	Select an option: – Basic: Use the provided user credentials to authenticate. – OAuth: Use OAuth2 IDs and tokens to authenticate. (Available for environments on Oracle Cloud Infrastructure (OCI) / Gen 2 architecture only.) Using OAuth2 in Cloud EPM Applications
Basic:	
User Name/Password	Credentials required for basic authentication
Identity Domain (Optional)	Identity domain to append in front of the user name when authorizing.
OAuth:	
Client ID	Client ID for the client that you created for global connections.
Access Token	Access token for the client that you created for global connections.
Refresh Token	Refresh token for the client that you created for global connections.
Default Folder (Optional)	Default folder to send extract files to in Cloud EPM environments: – Planning, FreeForm, Financial Consolidation and Close, Enterprise Profitability and Cost Management default application inbox – Data Management folder (for example, `inbox/repository`) – Profitability and Cost Management inbox (for example, `profitinbox`)

- **Oracle Cloud ERP parameters**:

Field	Description
Location	The URL for the EPM server. For example: `https://erp.mycloud.com`
User Name/Password	Credentials required for basic authentication
Identity Domain (Optional)	Identity domain to append in front of the user name when authorizing.
Default Document Account	Folder to extract file to in Oracle Cloud ERP. For example, `fin/generalLedger/import`.

- **Oracle Object Storage parameters**:

Field	Description
Location	The Swift URL of the Oracle Object Storage Cloud bucket. The format of the URL is: `https://swiftobjectstorage.region_identifier.oraclecloud.com/v1/namespace`, where: – `host name` (using the Swift API type), for example, `swiftobjectstorage`. – `region_identifier` is the hosting Oracle Cloud Infrastructure (OCI), for example `us-phoenix-1`. See Regions and Availability Domains. – `namespace` is the top-level container for all buckets and objects. Each Oracle Cloud Infrastructure tenant is assigned a unique system-generated and immutable Object Storage namespace name at account creation time. **Tip:** The Oracle Object store Swift URL should include the version and namespace. **Do not include a trailing forward slash in the Location parameter.**
User Name/Password	Credentials required for authentication, as follows: a. Generate an Auth token (see To create an Auth token in *Oracle Cloud Infrastructure Documentation*). b. For User Name, enter the OCI user name that the Auth token was generated for. c. For Password, enter the generated Auth token. This parameter is case-sensitive.
Default Bucket	Container to store the extracted file within the namespace for the object storage. This parameter is case-sensitive.
Default Subfolder (Optional)	Subfolder under the Default Bucket to write the extract file to. If the subfolder that you specify does not exist, it will be created. This parameter is case-sensitive.

5. (Optional): In **Supported Applications**, click **Specified**, and then select the applications that can use this global connection.
6. Click **Test Connection** to validate the connection details.

   > **Note:**
   >
   > For Oracle Object Storage connections, you must enter a Default Bucket in order to test the connection.

7. Click **Save**.

> **Note:**
>
> All issues must be corrected in all connections before you can save the connections. If there are any missing required fields in the connection details of any connections, an indicator is displayed on that connection and an error is displayed that one or more connections has issues that must be fixed.

To delete a global connection, from the Actions column click ⋮, and then click **Delete**. You cannot delete a global connection that is being used by an extract.

Working With Extracts

Extracts enable you to retrieve data from a viewpoint within a dimension into a formatted file or a global connection. You can extract from any viewpoint and for any properties. The viewpoints and properties do not have to be bound to be included in an extract.

Your Goal	Watch This Video
Learn more about how to use extracts to retrieve data from a viewpoint.	▶ Overview of Data Extracts

Extracts Versus Exports

Extracts are similar to exports in that they both retrieve data from dimensions. However, there are some key differences:

Extracts	Exports
Can be run for any viewpoint or property regardless of binding status	Limited to bound viewpoints and properties
Can output a full data set, a filtered subset of data, or incremental data changes for a viewpoint	Can output a full data set for bound viewpoints or incremental changes to a bound list viewpoint
Created and customized by dimension owners or metadata managers (for public extracts) or by users with access to a view and at least *Participant (Read)* permission on a viewpoint (for private extracts).	Created by the system based on registration settings
Formatting options are configurable by users. No constraints for external applications.	Contains target-specific logic and formatting so that they can be imported by external applications
Dimensions support multiple extracts	One export available per dimension
Extracts can be run for a single viewpoint at a time	Depending on the application type, exports can be run across multiple viewpoints
Can be extracted to a global connection or a file (depending on your permissions).	Can be exported to an application connection or a file.
Locale and time zone are fixed so that extracts always have the same output regardless of where they are run from	Depending on the application type, the date and time formatting may vary based on the user's locale.

Understanding Extract Types

The extract type enables you to extract either all data from a viewpoint, the changes to data over time, or to extract hierarchy information in separate columns.

Considerations for All Extract Types

- Extract files are encoded in UTF-8.
- Properties can be added to an extract more than once. This enables you to configure extract columns to display From and To values for nodes. See Selecting Extract Columns.
- You can create an expression to use as an inclusion or exclusion filter for the nodes in the extract. For example, you can extract all nodes with an ancestor of "A", or all nodes except those with an ancestor of "A". See Specifying Extract Options.

Full Extracts

Full extracts enable you to extract all data from a viewpoint for the current time period. The viewpoint can be bound or unbound, and the extract can include bound or unbound properties.

Incremental Extracts

Incremental extracts enable you to extract changes in a viewpoint between two time periods. The From and To time periods are compared, and the incremental differences are output in the extract file.

> **Note:**
>
> Incremental extracts are not available for time labeled viewpoints.

Considerations for Incremental Extracts

- To establish the time period for the incremental extract, you can specify From and To dates, a date offset (number of days prior to the current date), or you can extract all changes made since the last time an extract was run. See Editing Extracts.
- You can extract all actions, or you can specify the actions that you want to extract. For example, you can extract only nodes that were added or deleted, but not updated or renamed. See Specifying Incremental Options.
- The following extract options are not available for incremental extracts:
  - Pivot columns
  - Primary Key columns
  - Inverted Level specialty column

Generational Extracts

Generational extracts enable you to extract hierarchy relationships in a viewpoint as separate columns in the extract, rather than as parent and child columns as in the other extract types. You can define the maximum depth for the ancestors, and you can specify repeating options to fill in ancestor columns for nodes that are less than the maximum depth. See Specifying Generational Options.

Considerations for Generational Extracts

- You can run generational extracts on hierarchy viewpoints only.
- The **Source Type** column enables you specify that you are extracting a property for either a node or its ancestors. For example, to get the names of a node and its ancestors, you would add `Core.Name` to the extract twice and then use Source Type to specify that one is for nodes and the other is for its ancestors (see Example 1: Node and Ancestor Filters in a Balanced Hierarchy).
- For extract options, by default **Reconnect Parents** is set to enabled and **Exclude Implicitly Shared Nodes** is set to disabled. These settings cannot be changed. See Specifying Extract Options.

For examples of generational extracts, see Generational Extract Examples.

Link Extracts

Link extracts enable you to extract link information for a set of nodes in a viewpoint. Node links are relationships between nodes from different data sources, and they are established when an existing target node is updated by an incoming source node that has a defined data source. See About Node Links for more information.

Considerations for Link Extracts

- You can run link extracts on list or hierarchy viewpoints.
- If the extract is configured to extract link information (see Specifying Link Options) and a node in a viewpoint has more than one link, the extract will create a row for each link.

Generational Extract Examples

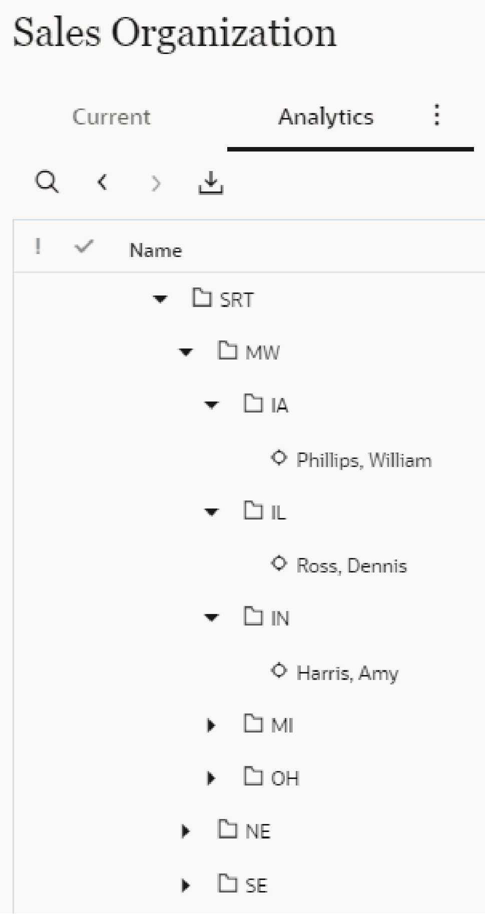
This topic provides some examples of different extract options such as the node and ancestor filters, the maximum depth, and the balance fill options in order to illustrate different ways to use generational extracts.

> **Tip:**
>
> - A *balanced hierarchy* is one in which all of the bottom nodes are at the same level.
> - An *unbalanced hierarchy* (or natural hierarchy) is one in which the bottom nodes are at different levels.
>
>   You can use the **Maximum Depth** and **Balance Fill Option** fields to balance an unbalanced hierarchy by repeating the parent, node, or top node in order to put all of the bottom nodes at the same level. This is illustrated in Example 2: Using Balance Fill Option in an Unbalanced Hierarchy.

Example 1: Node and Ancestor Filters in a Balanced Hierarchy

In this example, we will use the node and ancestor filters to extract a report of sales representatives by region. We will use a node filter to return only bottom level nodes, and then we will use an ancestor filter to reduce the report by filtering out the states to get a report by region. Because this is a balanced hierarchy, we do not need to use the Balance Fill Option.

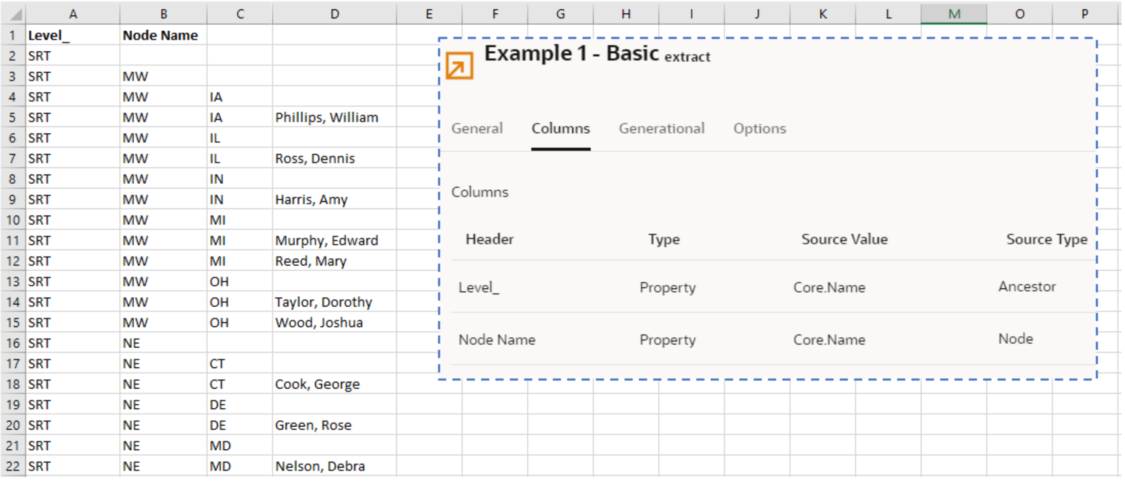
For this example, we will use a hierarchy that has a top node called Sales Representative by Territory (SRT) with several regional nodes under it: Midwest (MW), Northeast (NE), Southeast (SE), Southwest (SW), and West (W). Each region has a series of states under it (Iowa, Illinois, Indiana, Michigan, and Ohio under the Midwest, for example), and each state has a list of sales representatives as bottom nodes:



Sales Organization

- SRT
  - MW
    - IA
      - Phillips, William
    - IL
      - Ross, Dennis
    - IN
      - Harris, Amy
    - MI
    - OH
  - NE
  - SE

We will start by extracting the names of the nodes and their ancestors by adding `Core.Name` for the ancestors and `Core.Name` for the nodes to the extract (see columns in an inset, below).



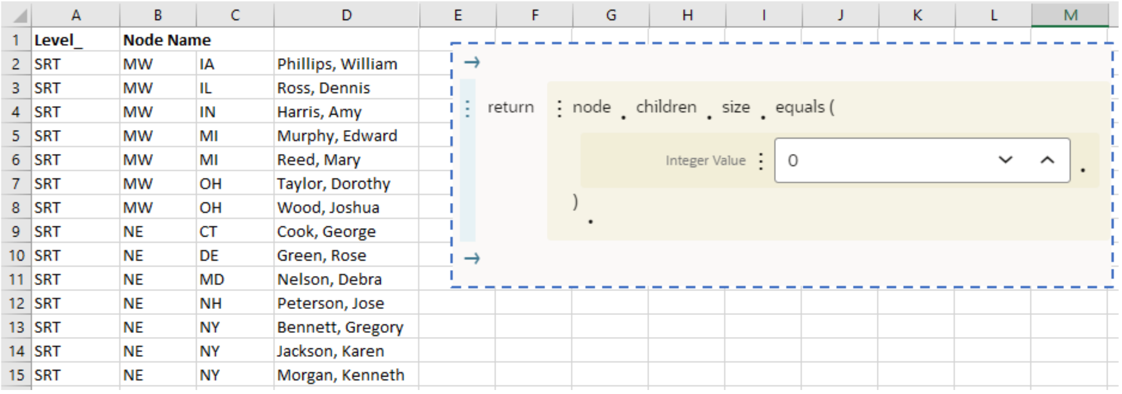
	A	B	C	D
1	Level_	Node Name		
2	SRT			
3	SRT	MW		
4	SRT	MW	IA	
5	SRT	MW	IA	Phillips, William
6	SRT	MW	IL	
7	SRT	MW	IL	Ross, Dennis
8	SRT	MW	IN	
9	SRT	MW	IN	Harris, Amy
10	SRT	MW	MI	
11	SRT	MW	MI	Murphy, Edward
12	SRT	MW	MI	Reed, Mary
13	SRT	MW	OH	
14	SRT	MW	OH	Taylor, Dorothy
15	SRT	MW	OH	Wood, Joshua
16	SRT	NE		
17	SRT	NE	CT	
18	SRT	NE	CT	Cook, George
19	SRT	NE	DE	
20	SRT	NE	DE	Green, Rose
21	SRT	NE	MD	
22	SRT	NE	MD	Nelson, Debra

Example 1 - Basic extract

Columns

Header	Type	Source Value	Source Type
Level_	Property	Core.Name	Ancestor
Node Name	Property	Core.Name	Node

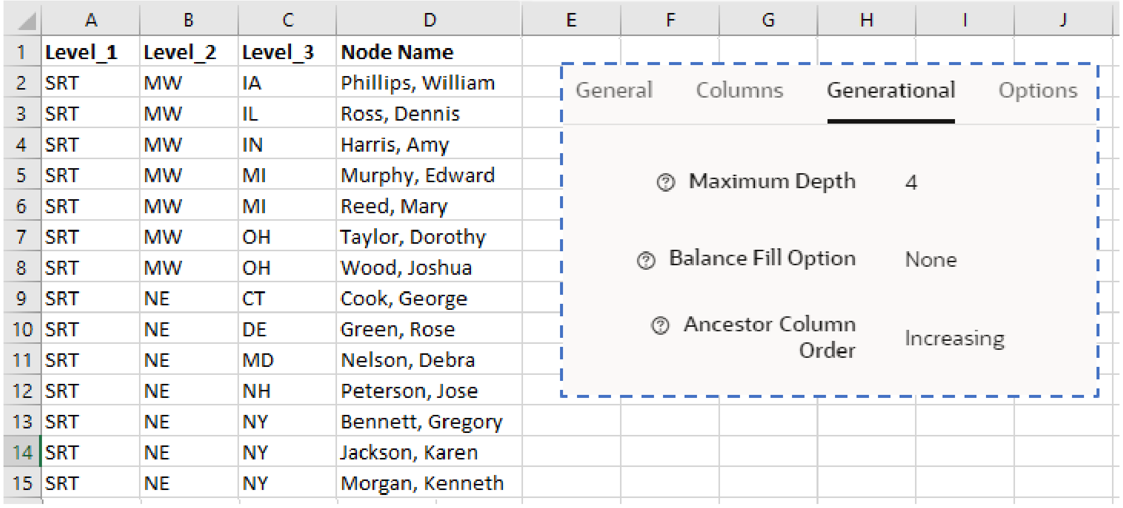
Notice that several rows in the extract have information that is not useful in our report. For example, rows 2, 3, and 4 contain ancestor information but not the bottom nodes that have the sales representative information that we are looking for. We can use a node filter expression (displayed in an inset) to return only bottom level nodes.



	A	B	C	D
1	Level_	Node Name		
2	SRT	MW	IA	Phillips, William
3	SRT	MW	IL	Ross, Dennis
4	SRT	MW	IN	Harris, Amy
5	SRT	MW	MI	Murphy, Edward
6	SRT	MW	MI	Reed, Mary
7	SRT	MW	OH	Taylor, Dorothy
8	SRT	MW	OH	Wood, Joshua
9	SRT	NE	CT	Cook, George
10	SRT	NE	DE	Green, Rose
11	SRT	NE	MD	Nelson, Debra
12	SRT	NE	NH	Peterson, Jose
13	SRT	NE	NY	Bennett, Gregory
14	SRT	NE	NY	Jackson, Karen
15	SRT	NE	NY	Morgan, Kenneth

return node . children . size . equals (
Integer Value 0
) .

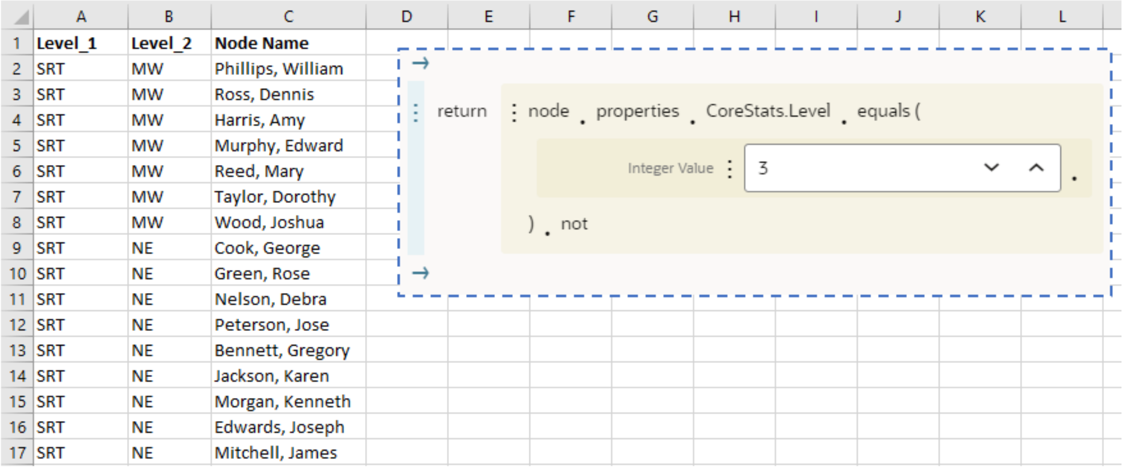
Next, we want to reduce our extract to display the sales representatives by region, but not by state. This requires two changes to our extract. First, we need to identify the ancestor level where the state information is stored. We do that by specifying the maximum depth (see inset) so that the extract adds header information to each column. Notice that columns A, B, C, and D now all have headers that identify the levels and node name.



	A	B	C	D
1	Level_1	Level_2	Level_3	Node Name
2	SRT	MW	IA	Phillips, William
3	SRT	MW	IL	Ross, Dennis
4	SRT	MW	IN	Harris, Amy
5	SRT	MW	MI	Murphy, Edward
6	SRT	MW	MI	Reed, Mary
7	SRT	MW	OH	Taylor, Dorothy
8	SRT	MW	OH	Wood, Joshua
9	SRT	NE	CT	Cook, George
10	SRT	NE	DE	Green, Rose
11	SRT	NE	MD	Nelson, Debra
12	SRT	NE	NH	Peterson, Jose
13	SRT	NE	NY	Bennett, Gregory
14	SRT	NE	NY	Jackson, Karen
15	SRT	NE	NY	Morgan, Kenneth

Maximum Depth 4
Balance Fill Option None
Ancestor Column Order Increasing

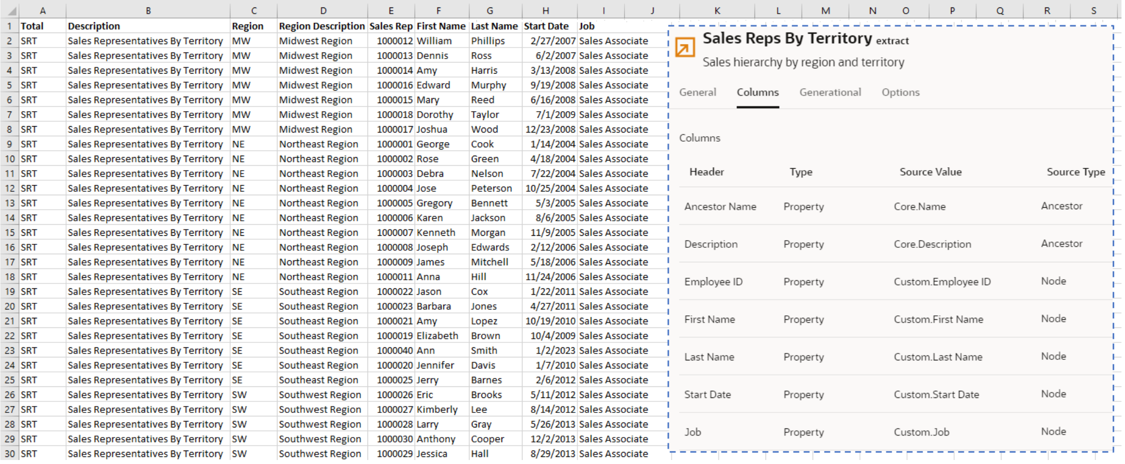
Now that we know that the state information is at ancestor level 3, we can add an ancestor filter to filter out level 3 and return the regional information that we are looking for. The ancestor filter is displayed in an inset. Remember that now that we are filtering out a level, we need to change our Maximum Depth to 3 instead of 4.



	Level_1	Level_2	Node Name
2	SRT	MW	Phillips, William
3	SRT	MW	Ross, Dennis
4	SRT	MW	Harris, Amy
5	SRT	MW	Murphy, Edward
6	SRT	MW	Reed, Mary
7	SRT	MW	Taylor, Dorothy
8	SRT	MW	Wood, Joshua
9	SRT	NE	Cook, George
10	SRT	NE	Green, Rose
11	SRT	NE	Nelson, Debra
12	SRT	NE	Peterson, Jose
13	SRT	NE	Bennett, Gregory
14	SRT	NE	Jackson, Karen
15	SRT	NE	Morgan, Kenneth
16	SRT	NE	Edwards, Joseph
17	SRT	NE	Mitchell, James

```
return : node . properties . CoreStats.Level . equals (
    Integer Value : 3 .
) . not
```

We finalize our report by adding some additional columns such as Employee ID, Start Date, and Job Title (see inset) and using custom headers to identify the information in the report.



	Total	Description	Region	Region Description	Sales Rep	First Name	Last Name	Start Date	Job
2	SRT	Sales Representatives By Territory	MW	Midwest Region	1000012	William	Phillips	2/27/2007	Sales Associate
3	SRT	Sales Representatives By Territory	MW	Midwest Region	1000013	Dennis	Ross	6/2/2007	Sales Associate
4	SRT	Sales Representatives By Territory	MW	Midwest Region	1000014	Amy	Harris	3/13/2008	Sales Associate
5	SRT	Sales Representatives By Territory	MW	Midwest Region	1000016	Edward	Murphy	9/19/2008	Sales Associate
6	SRT	Sales Representatives By Territory	MW	Midwest Region	1000015	Mary	Reed	6/16/2008	Sales Associate
7	SRT	Sales Representatives By Territory	MW	Midwest Region	1000018	Dorothy	Taylor	7/1/2009	Sales Associate
8	SRT	Sales Representatives By Territory	MW	Midwest Region	1000017	Joshua	Wood	12/23/2008	Sales Associate
9	SRT	Sales Representatives By Territory	NE	Northeast Region	1000001	George	Cook	1/14/2004	Sales Associate
10	SRT	Sales Representatives By Territory	NE	Northeast Region	1000002	Rose	Green	4/18/2004	Sales Associate
11	SRT	Sales Representatives By Territory	NE	Northeast Region	1000003	Debra	Nelson	7/22/2004	Sales Associate
12	SRT	Sales Representatives By Territory	NE	Northeast Region	1000004	Jose	Peterson	10/25/2004	Sales Associate
13	SRT	Sales Representatives By Territory	NE	Northeast Region	1000005	Gregory	Bennett	5/3/2005	Sales Associate
14	SRT	Sales Representatives By Territory	NE	Northeast Region	1000006	Karen	Jackson	8/6/2005	Sales Associate
15	SRT	Sales Representatives By Territory	NE	Northeast Region	1000007	Kenneth	Morgan	11/9/2005	Sales Associate
16	SRT	Sales Representatives By Territory	NE	Northeast Region	1000008	Joseph	Edwards	2/12/2006	Sales Associate
17	SRT	Sales Representatives By Territory	NE	Northeast Region	1000009	James	Mitchell	5/18/2006	Sales Associate
18	SRT	Sales Representatives By Territory	NE	Northeast Region	1000011	Anna	Hill	11/24/2006	Sales Associate
19	SRT	Sales Representatives By Territory	SE	Southeast Region	1000022	Jason	Cox	1/22/2011	Sales Associate
20	SRT	Sales Representatives By Territory	SE	Southeast Region	1000023	Barbara	Jones	4/27/2011	Sales Associate
21	SRT	Sales Representatives By Territory	SE	Southeast Region	1000021	Amy	Lopez	10/19/2010	Sales Associate
22	SRT	Sales Representatives By Territory	SE	Southeast Region	1000019	Elizabeth	Brown	10/4/2009	Sales Associate
23	SRT	Sales Representatives By Territory	SE	Southeast Region	1000040	Ann	Smith	1/2/2023	Sales Associate
24	SRT	Sales Representatives By Territory	SE	Southeast Region	1000020	Jennifer	Davis	1/7/2010	Sales Associate
25	SRT	Sales Representatives By Territory	SE	Southeast Region	1000025	Jerry	Barnes	2/6/2012	Sales Associate
26	SRT	Sales Representatives By Territory	SW	Southwest Region	1000026	Eric	Brooks	5/11/2012	Sales Associate
27	SRT	Sales Representatives By Territory	SW	Southwest Region	1000027	Kimberly	Lee	8/14/2012	Sales Associate
28	SRT	Sales Representatives By Territory	SW	Southwest Region	1000028	Larry	Gray	5/26/2013	Sales Associate
29	SRT	Sales Representatives By Territory	SW	Southwest Region	1000030	Anthony	Cooper	12/2/2013	Sales Associate
30	SRT	Sales Representatives By Territory	SW	Southwest Region	1000029	Jessica	Hall	8/29/2013	Sales Associate

Sales Reps By Territory extract

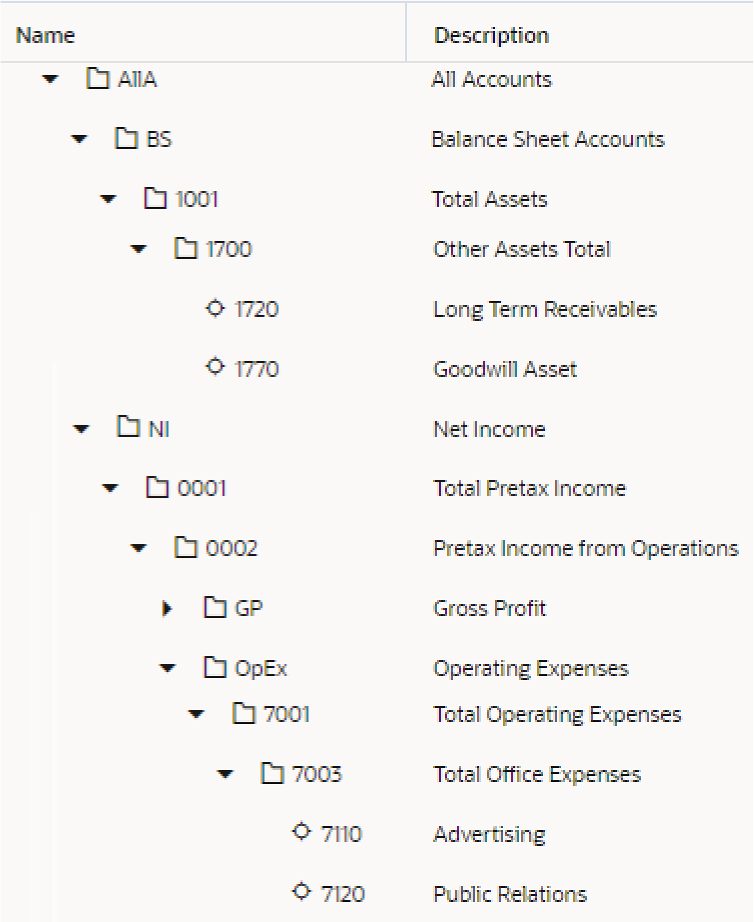
Sales hierarchy by region and territory

Columns

Header	Type	Source Value	Source Type
Ancestor Name	Property	Core.Name	Ancestor
Description	Property	Core.Description	Ancestor
Employee ID	Property	Custom.Employee ID	Node
First Name	Property	Custom.First Name	Node
Last Name	Property	Custom.Last Name	Node
Start Date	Property	Custom.Start Date	Node
Job	Property	Custom.Job	Node

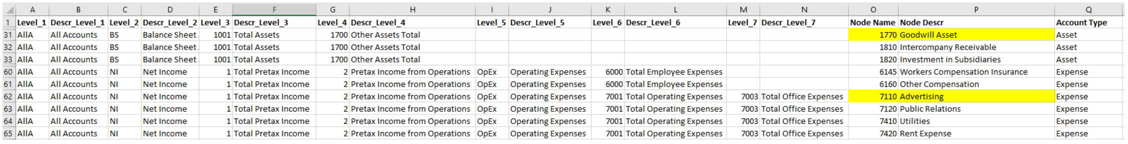
Example 2: Using Balance Fill Option in an Unbalanced Hierarchy

Next, let's look at an example of using the Balance Fill Option to balance out an unbalanced hierarchy. This time, we will use an Account hierarchy. Notice that the bottom nodes `1770 Goodwill Asset` and `7110 Advertising` are at different levels in the hierarchy.



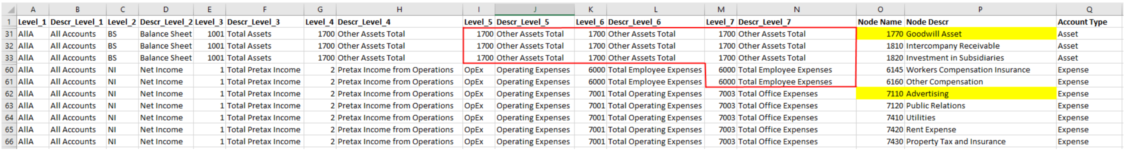
Name	Description
AllA	All Accounts
BS	Balance Sheet Accounts
1001	Total Assets
1700	Other Assets Total
1720	Long Term Receivables
1770	Goodwill Asset
NI	Net Income
0001	Total Pretax Income
0002	Pretax Income from Operations
GP	Gross Profit
OpEx	Operating Expenses
7001	Total Operating Expenses
7003	Total Office Expenses
7110	Advertising
7120	Public Relations

When we extract this hierarchy with a maximum depth of 8, the columns for levels 5, 6, and 7 for bottom node `1770 Goodwill Asset` have null values, making this hierarchy unbalanced.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Level_1	Descr_Level_1	Level_2	Descr_Level_2	Level_3	Descr_Level_3	Level_4	Descr_Level_4	Level_5	Descr_Level_5	Level_6	Descr_Level_6	Level_7	Descr_Level_7	Node Name	Node Descr	Account Type
31	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total							1770	Goodwill Asset	Asset
32	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total							1810	Intercompany Receivable	Asset
33	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total							1820	Investment in Subsidiaries	Asset
60	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	6000	Total Employee Expenses			6145	Workers Compensation Insurance	Expense
61	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	6000	Total Employee Expenses			6160	Other Compensation	Expense
62	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7110	Advertising	Expense
63	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7120	Public Relations	Expense
64	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7410	Utilities	Expense
65	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7420	Rent Expense	Expense

We select **Repeat Parent** in **Balance Fill Option** to fill in the hierarchy with the parent name and description where there are no values (red highlighted area, below). The hierarchy is now balanced.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Level_1	Descr_Level_1	Level_2	Descr_Level_2	Level_3	Descr_Level_3	Level_4	Descr_Level_4	Level_5	Descr_Level_5	Level_6	Descr_Level_6	Level_7	Descr_Level_7	Node Name	Node Descr	Account Type
31	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1770	Goodwill Asset	Asset
32	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1810	Intercompany Receivable	Asset
33	AllA	All Accounts	BS	Balance Sheet	1001	Total Assets	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1700	Other Assets Total	1820	Investment in Subsidiaries	Asset
60	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	6000	Total Employee Expenses	6000	Total Employee Expenses	6145	Workers Compensation Insurance	Expense
61	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	6000	Total Employee Expenses	6000	Total Employee Expenses	6160	Other Compensation	Expense
62	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7110	Advertising	Expense
63	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7120	Public Relations	Expense
64	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7410	Utilities	Expense
65	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7420	Rent Expense	Expense
66	AllA	All Accounts	NI	Net Income	1	Total Pretax Income	2	Pretax Income from Operations	OpEx	Operating Expenses	7001	Total Operating Expenses	7003	Total Office Expenses	7430	Property Tax and Insurance	Expense

Extract Visibility

Extract visibility defines whether the extract is Private, meaning only the user who created it can see and run it, or Public, meaning all users with appropriate data access can see and run it.

Private Extracts

When you first create an extract, it is created as a private extract. Private extracts can only be viewed and run by the user who created them. You must have access to a view and *Participant (Read)* or better access on a viewpoint in order to create or copy a private extract. There are no permission requirements to edit or delete a private extract.

You run private extracts from the Viewpoint Extract report, and they can be run to a file only. See Extracts Reports.

Public Extracts

You cannot create a public extract directly, but if you have *Owner* or *Metadata Manager* permission on the dimension, you can promote a private extract to public so that other users can see and run it. You can also copy, edit, or delete a public extract. See Creating, Copying, Promoting, and Deleting Extracts

You can run public extracts from the Applications list or the Viewpoint Extracts report, and they can be run to a file or a global connection. See Running Extracts.

Creating, Copying, Promoting, and Deleting Extracts

You create extracts from the dimension inspector. You can create a private extract if you have *Participant (Read)* permission or better on a viewpoint. You must have *Owner* or *Metadata Manager* permission on a dimension in order to be able to promote a private extract to a public extract.

Your Goal	Watch This Video
Learn how to create an extract	Creating Extracts

To create an extract:

1. Inspect the application that contains the dimension that you want to create an extract for. See Inspecting Applications.
2. From the Dimensions tab, click the dimension that you want to create an extract for. The dimension is displayed in the dimension inspector.
3. On the Extracts tab, click **Create**.
4. From the **Viewpoint** drop down menu, select the viewpoint that you want to extract the dimension for.

> **Note:**
>
> Only active viewpoints that contain the dimension that you selected are displayed. You cannot select a viewpoint that contains Legacy GL or Lookup class node types unless it is a mapping bound viewpoint.

5. Enter a name and, optionally, a description.
6. Select an extract type:
   - **Full** (default): Extract all data from a viewpoint.
   - **Incremental**: Extract only the changes to data over a specified time period for a viewpoint.
   - **Generational**: Extract hierarchy relationships in a viewpoint in separate extract columns.
   - **Link**: Extract link information in a viewpoint.

   See Understanding Extract Types.

   > **Note:**
   >
   > You cannot change the extract type after the extract is created.

7. Click **Create**.
   The extract is displayed in the extract inspector.

> **Note:**
>
> When you create an extract, it is created as a private extract and you cannot change the visibility from the Create screen. If you have the *Owner* or *Metadata Manager* permission on a dimension, you can promote your private extracts to public. See Promoting Private Extracts to Public.

Edit the extract to add columns and modify the formatting options. See Editing Extracts.

Copying Extracts

You can copy an extract and then modify it for other uses (for example, by changing the viewpoint for the extract or modifying the extract options to select different data).

Considerations

- When you copy an extract, you can create the new extract in:
  - The same dimension and viewpoint as the original
  - A different dimension and viewpoint in the same application as the original
  - A different dimension and viewpoint in a different application of the same type (for example, Planning or Financials Cloud General Ledger) as the original.
- You must have at least *Participant (Read)* access for all data chain objects in the dimension that you are copying an extract to.

To copy an extract:

1. From the Extracts tab of the dimension inspector, in the Actions column of the extract to copy, click the **Extract Action Menu** ⋮, then click **Copy**, and then click **Yes** on the message about saving changes.
2. **Optional**: Enter a new name and description. The new name must be unique in the target dimension.

3. **Optional**: Specify a different application, dimension, view, and viewpoint to create the extract in.

   > **Note:**
   >
   > You cannot specify a viewpoint that contains Legacy GL or Lookup class node types unless it is a mapping bound viewpoint.

4. Click **Copy**.
   The new extract is displayed in the extract inspector. You can modify the extract to change its settings. See Editing Extracts

   > **Note:**
   >
   > The system displays a warning if any properties in the extract do not exist in the target viewpoint. You can edit the new extract to remove the columns for the properties that do not exist in the target viewpoint. See Selecting Extract Columns.

Promoting Private Extracts to Public

When you promote a private extract to public, other users with access to the view and at least *Participant (Read)* access on the viewpoint can view, copy, and run it, and other users with the *Owner* or *Metadata Manager* permission on the dimension can edit and delete it.

To promote a private extract:

1. From the Extracts tab of the dimension inspector, in the Actions column of the extract to promote, click **Extract Action Menu** ⋮, and then click **Promote**.
2. Click **Yes** to confirm, and then click **Save**.

Deleting Extracts

You can delete any private extracts that you created, even if you no longer have *Participant (Read)* access on the viewpoint. To delete public extracts, you must have the *Owner* or *Metadata Manager* permission on the dimension in the extract.

To delete an extract:

1. From the Extracts tab of the dimension inspector, in the Actions column of the extract to delete, click the **Extract Action Menu** ⋮, and then click **Delete**.
2. Click **Yes** to confirm, and then click **Save**.

Editing Extracts

After you have created an extract, you can edit it to change the general extract settings, select the columns, and modify the formatting options. You must have *Owner* or *Metadata Manager* permission on a dimension in order to be able to edit a public extract.

To edit an extract:

1. Inspect the application that contains the extract that you want to edit. See Inspecting Applications.
2. From the Dimensions tab, click the dimension that contains the extract, and then select the Extracts tab.
3. On the General tab of the extract inspector, click **Edit** and change any of these general settings:
   - Name or description of the extract
   - **Integration Only** (Public extracts only): Select this option to specify that this extract is for system integration purposes only. Integration Only extracts are not displayed in the Viewpoint Extracts report and can only be run from the application worklist.
   - The view or viewpoint to run the extract against. Only active viewpoints within the dimension that you are creating the extract for are able to be selected. If the viewpoint has been archived, a warning icon is displayed.
     You cannot select a viewpoint that contains Legacy GL or Lookup class node types unless it is a mapping bound viewpoint.

     > **Caution:**
     >
     > If you change the viewpoint for an extract, the following may occur:
     >
     > – Any columns that were in the original viewpoint but not in the new viewpoint are removed from the extract.
     >
     > – If you set up an extract filter, it may become invalid. For example, if you change from a hierarchy viewpoint to a list viewpoint, any relationship-level properties in your filter (such as Parent) will become invalid.

   - In **Extract File Type**, select the file type (text or Excel) for the extract. If you select Excel, enter the name of the worksheet for the extract in **Sheet Name**.
   - Default file name for the text file to be created for the extract. Ensure that you enter a valid text format such as .csv, .xlsx, .txt, or .tsv.
   - **Zip Extract** option. Select the check box to extract data in a zipped format. Then, specify a zip file name ending with `.zip`.

     > **Note:**
     >
     > The **Zip File Name** field is visible only after you enable **Zip Extract**. You must enter the `.zip` file extension after the file name before you can save your changes.

4. *Public extracts only*: Specify the global connection name and any parameters for the extract. The parameters that you specify are specific to the connection type. See Creating Global Connections.

> **Note:**
>
> You cannot run private extracts to a global connection. In order to run an extract to a global connection, it must be promoted to public. See Creating, Copying, Promoting, and Deleting Extracts.

You can test the connection before saving. The **Test Connection** button is enabled only if you have set up a global connection and defined the required parameters. See Creating Global Connections.

5. *Incremental extracts only*: In **From** and **To** sections, specify the start and end periods for the extract. The From and To times cannot be the same, but the From time can be later than the To time. In that case, the action codes are reversed. For example, if a node was added and your extract has the From date after the To date, the extract contains a delete for that node.

   - **Current**: The current date and time when the extract is run is used.
   - **Specified**: In **Timestamp**, enter a specific date and time. You can select the time stamp from the date picker or manually edit the date and time.
   - **Offset**: In **Days Prior to Current field**, specify the number of days before the current date to use. In **Time of Day**, specify a time on the offset date.
   - **Since Last Extract**: The date of the last extract is used. If an extract has never been run against this viewpoint, the creation date of the viewpoint is used.

   > **Note:**
   >
   > The From and To dates use the time zone that is specified on the Options tab. See Specifying Extract Options. If no time zone is specified, the default time zone is `UTC`.

6. Click **Save**.

Selecting Extract Columns

The Columns tab of the extract inspector enables you to select the properties and constant values that you want to extract for the dimension.

To select the extract columns:

1. On the Columns tab, click **Edit**.
2. To add properties to the extract, perform the following actions:
   a. Click **Manage Properties**.
   The properties from the viewpoint are displayed.

   > **Note:**
   >
   > For Link extracts, only node level properties are displayed.

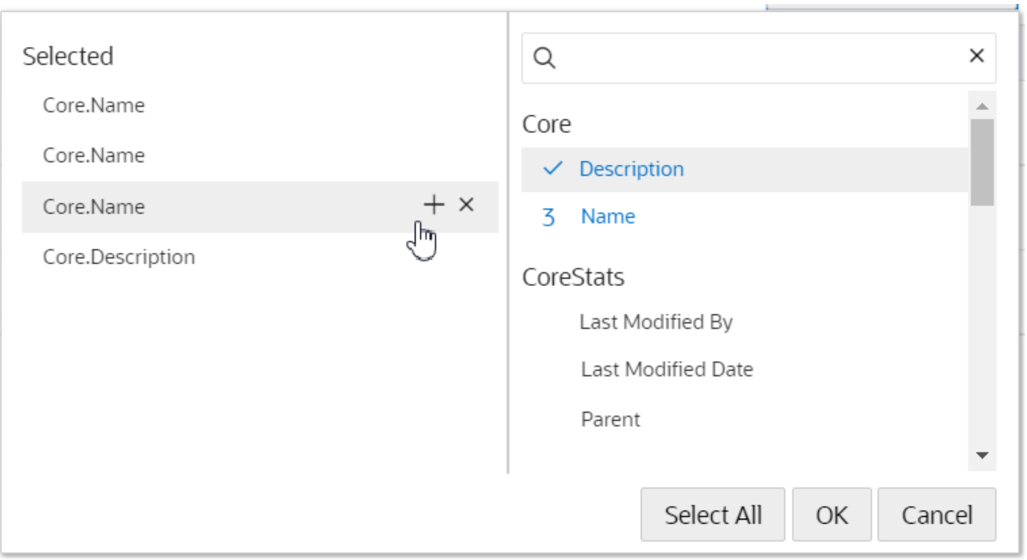
   b. Click the properties that you want to add to the extract, or click **Select All** to add all available properties.

Selected properties are indicated by a check mark. Click X to remove the property.

Incremental and Generational extracts only: On the selected properties, click + to add the property to the extract again. The number of instances of each property is indicated.

Adding a property more than once and using the **Source Type** field (see step 5, below) allows you to, for example, have different columns for the From and To values of the same property in an incremental extract, or to have different columns for the values of the same property for a Node and its Ancestors.

Click **X** to remove an instance of the property from the selected column.



Tip:

Properties that you select are added to the extract in the order that you click them. Select the properties in the order that you want them in the file to reduce re-order actions later.

c. Click **OK**.
The selected properties are added to the Columns page.

3. **Optional**: In **Header**, edit the column header name for the property.

Note:

The column header names must be unique for each property in an extract.

4. To add specialty columns to the extract, click **Add Column**, select the type of specialty column to add, and enter a column header name. The type of specialty column that you can add is dependent on the extract type.

Table 30-2 Extract Specialty Columns

Column Type	Description	Extract Type
Action Code	The action that was performed on the node. You can specify the label for the action code on the Incremental tab. See Specifying Incremental Options.	Incremental
Constant	User defined constant value **Note:** Enter the value for the constant in **Source Value**.	All
Data Source	Data source for the link	Link
Data Source Code	Data source code for the link	Link
Extract Run ID	Unique GUID for the extract being run (duplicated for all records)	All
Extract Timestamp	Timestamp of extract (duplicated for all records)	All
Inverted Level **Note:** Available for extracts run against hierarchy viewpoints only.	Bottom-up level of the node in a hierarchy set Bottom level nodes have a level of 0, their immediate parents have a level of 1, and so on. For example, in a hierarchy where a top node has one child node and that child node has one child node, the bottom node level is 0 and the top node level is 2.	Full
Node Type	Node type of the node being extracted	Link
Record Number	Incremental number starting at 1 for the first record	All
Source Name	Name of the source node for the link	Link
Source Node ID	Node ID of the source node for the link	Link
Source Node Type	Node type of the source node for the link	Link
Source Request Number	Request number for the link	Link
User Name	Name of the user running the extract (duplicated for all records)	All

5. **Optional** *(Incremental and Generational extracts only)*: In **Source Type**, specify what the value of the property is based on:
Incremental:

- **From**: The value is based on the From version of the viewpoint
- **To**: The value is based on the To version of the viewpoint
- **Combined**: The value is based on the To version of the viewpoint if it is available. If not, the value is based on the From version of the viewpoint.

> **Note:**
>
> For rename actions on the `Core.Name` property only, the **Combined** value is based on the From version of the viewpoint.

Generational:

- **Node**: The value of the column is based on the node being extracted
- **Ancestor**: The value of the column is based on the ancestor of the node being extracted

> **Note:**
>
> Ancestor columns must be grouped together.

6. **Optional** *(Full extracts only)*: Select the **Pivot** check box to specify a property to pivot on in the extract file. This will produce a separate row in the extract file for every value in the pivoting property (including null or empty values). Only one property can be pivoted in an extract.

> **Note:**
>
> You can specify List, Node List data type, or String data type properties as pivot columns. For string data types, specify a pivot delimiter on the Options tab. See Specifying Extract Options.

7. **Optional** *(Full extracts only)*: Select the **Key** check box to designate a column as a primary key column. Columns marked as primary key columns will output a single row of data for each unique instance of the key values. Duplicate instances of primary key rows are suppressed from the extract output.
You can designate more than one column as primary key columns. Duplicates are determined based on the combination of values in the key columns.

> **Note:**
>
> When using both the Pivot and Key options:
>
> - The pivot operation takes place first, and then duplicate rows of primary key values are filtered out from the extract.
> - The column that you are pivoting on should also be a primary key column. Otherwise, only the first record of the pivoting property will be extracted

8. **Optional** *(Full extracts only)*: Use the **Sort** column to configure the sort order of the rows that are written to the extract file. You can specify one primary and one secondary column, and you can sort the rows in ascending or descending order.

9. **Optional** *(Incremental extracts only)*: Use the **Compare** and **Output** columns to determine whether a property column uses compare logic to determine if a record for this node should be included and whether the column should be output, as follows:

   - Use **Compare** to determine if a record is generated when there are differences between the From and To values for the property.
   - Use **Output** to determine whether or not the column is included in the output if a record is generated for this node.

Compare Setting	Output Setting	Extract Behavior
Enabled	Enabled	*(Default behavior)* The property is compared, and if there are differences in the property between the From and the To versions of the property value, a record is generated for the node and the column is included in the extract.
Disabled	Enabled	The property is not compared. Differences in the property between the From and the To versions of the property value do not generate a record. However, if a record is generated for this node because there are differences in other properties that are being compared, then this column is also included in the extract.
Enabled	Disabled	The property is compared, and if there are differences in the property between the From and the To versions of the property value, a record is generated for this node. However, the column is not included in the extract.
Disabled	Disabled	The property is not compared, and it is not included in the extract.

> **Note:**
>
> These settings are applicable only if you have enabled **Process Property Updates** in the Incremental Options. See Specifying Incremental Options.

10. To remove columns, perform an action:

    - Click **Remove All** to remove all of the columns from the extract.
    - To remove individual columns, in the Actions column of the property or constant that you want to remove, click ⋮ and then select **Remove**
    - From **Manage Properties**, click **X** on a selected property to remove it from the extract.

11. To re-order columns, perform an action:

    - Drag and drop the row for the property or constant column that you want to move

- In the Actions column of the property or constant that you want to move, click ⋮ and then select one of the move actions.

12. Click **Save**.

Specifying Incremental Options

The Incremental tab of the extract inspector enables you to specify the incremental options for the extract, such as the actions that you want to extract or whether or not to combine renames, moves, and updates. This tab is displayed only if the extract type is Incremental.

To specify incremental options:

1. On the Incremental tab of the extract inspector, click **Edit**.
2. On each of the action types, select the check box to process the action and, optionally, specify an action code to output for that action in the extract. By default, all actions are processed in an incremental extract. The following table lists the default action codes for each action.

Action	Default Action Code
Process Adds **Note:** If you do not enable **Process Moves**, then by default the structural changes resulting from Move operations are output as Adds and Deletes (if those options are enabled). If you don't want Add and Delete operations that resulted from Moves in your output file, enable the **Exclude Moves from Adds and Deletes** option. See step 4, below.	`Add`
Process Deletes **Note:** If you do not enable **Process Moves**, then by default the structural changes resulting from Move operations are output as Adds and Deletes (if those options are enabled). If you don't want Add and Delete operations that resulted from Moves in your output file, enable the **Exclude Moves from Adds and Deletes** option. See step 4, below.	`Delete`
Process Moves	`Move`
Process Renames **Note:** If a Reset Import was performed during the incremental extract period, the From and To versions for a property cannot be determined.	`Rename`

Action	Default Action Code
Process Property Updates **Note:** Enable **Skip Missing Properties** to specify that if a property is not present in both the From and To versions of the viewpoint during an update, it is not included in the extract.	`Prop_Update`

3. Enable **Combine Renames, Moves, and Updates** to specify that if a property has multiple actions performed on it, the extract combines them into single record with the highest precedence action code. The order of precedence for action codes is **Renames**, then **Moves**, then **Updates**.
   For example, if a property was moved under a new parent and also had its description updated, if this setting is enabled then instead of two records in the extract file (one for the move and one for the update), the extract will contain only one record with an action code of Move.

4. Enable **Exclude Moves from Adds and Deletes** to exclude move actions when Adds and Deletes are processed without also processing Moves. By default, if you select to process Adds or Deletes but not Moves, the structural changes resulting from Move operations are output as Adds and Deletes (if those options are enabled) in the extract file. If you select this option, the structural changes resulting from Moves are not output in the extract file. Note the following about this option:
   - If you select to process either Adds or Deletes (or both) but not Moves, by default this option is not selected and the structural changes resulting from Moves are displayed as Adds and Deletes in the extract file. You can select the option to prevent the Moves from being output as Add and Delete actions in the extract file.
   - If you select to process either Adds or Deletes (or both) as well as Moves, this option is automatically selected and cannot be changed.
   - If you do not select to process both Adds and Deletes, this option is not visible.

Specifying Generational Options

The Generational tab of the extract inspector enables you to specify the generational options for the extract, such as the maximum depth and column order of ancestors and the balance fill options. This tab is displayed only if the extract type is Generational.

To specify generational options:

1. On the Generational tab of the extract inspector, click **Edit**.
2. In **Maximum Depth**, enter an integer (either 0 or 2 through 99) to define the number of columns to generate for the ancestor information in the extract.
   - A value of 2 through 99 will result in a balanced extract, as every record has the same number of repeated ancestor columns (the number of columns created for the ancestors is the specified maximum depth minus one).
   - A value of 0 will result in the columns and headers in the extract not being balanced, as each record might have a different number of columns.

- You cannot enter a value of 1, because that would not include any ancestor information in the extract.

> **Note:**
>
> If the actual depth of an ancestor node is greater than the maximum depth that you specify, that ancestor node is skipped in the extract. To prevent this, you can expand your maximum depth, or you can use the Ancestor Filter on the Extract Options tab to filter down your list of ancestors. See Specifying Extract Options.

3. In **Balance Fill Option**, select a value to place in the ancestor columns for records that have fewer ancestors than the specified maximum depth level.
   - **None**: The ancestor columns for levels that are not present are filled with the **Null Keyword** value in Extract Options. See Specifying Extract Options.
   - **Repeat Parent**: The ancestor columns for levels that are not present are filled with the values from the parent node for the current node being extracted.
   - **Repeat Node**: The ancestor columns for levels that are not present are filled with the values from the node being extracted.
   - **Repeat Top Node**: The ancestor columns for top levels are populated with values from the top node, and the rest of the levels are pushed down so that the last ancestor level is at the lowest level.

   For example, suppose you had a hierarchy with A as the top node, B as a child of A, C as a child of B, and then 111 as the bottom level node, and your maximum depth is 5. The following table illustrates how each of the Balance Fill Option settings would be displayed:

Balance Fill Option	First Column	Second Column	Third Column	Fourth Column	Fifth Column
None	A	B	C		111
Repeat Parent	A	B	C	C	111
Repeat Node	A	B	C	111	111
Repeat Top Node	A	A	B	C	111

4. In **Ancestor Column Order**, specify the order of the ancestor columns (Increasing or Decreasing) in the extract. So for example if your hierarchy goes from level 1 to level 10, Increasing would start at level 1 (the top of the hierarchy) and move down, while Decreasing starts at level 10 (bottom level) and moves up.
5. Select **Inclusive Ancestors** to include the node that's being extracted in the ancestor columns of the extract.
   For example, suppose you have node 111 with ancestors A, B, and C:
   - **Inclusive Ancestors de-selected** (default): The ancestor columns are populated with A, B, and C.
   - **Inclusive Ancestors selected**: The ancestor columns are populated with A, B, C, and 111.

Specifying Link Options

The Link tab of the extract inspector enables you to specify the link options for the link extract type, such as the data source and node link filters. This tab is displayed only if the extract type is Link.

To specify link options:

1. On the Link tab of the extract inspector, click **Edit**.
2. **(Optional)**: In **Data Source Filter**, add one or more data sources to include in the extract. Multiple data sources are combined with OR logic (that is, if a node link exists with any of the data sources that you specify, it is included in the extract). If you leave this field blank, all data sources are included.
3. In **Node Link Filter**, select an option to include in the extract:
   - **All Nodes**: All nodes are included in the extract, whether or not they have links.
   - **Nodes with Links**: Only nodes with links are included in the extract. Nodes without links are excluded from the extract.
   - **Nodes without Links**: Only nodes without links are included in the extract. Nodes with link information are excluded from the extract.

> **Note:**
>
> The Data Source Filter and Node Link Filter controls work together. The extract will display data source information only for the data sources that you specify, even if you select All Nodes in the Node Link Filter. For example, if you specify data source *Financials* in the Data Source Filter, and you select *All Nodes* in the Node Link Filter, the extract will contain all of the nodes in the viewpoint, but only the nodes with a data source of Financials will have link information in the extract. Link information for the nodes from other data sources is not included.

Specifying Extract Options

The Options tab of the extract inspector enables you to specify the options, such as the file and data formats, for creating the extract file.

To specify the extract options:

1. On the Options tab of the extract inspector, click **Edit**.
2. **Optional**: Define the Filter parameters.
   - **Node Filter**: Click **Define Expression** *fx* to open the expression builder and define an expression to select the nodes to be extracted. The expression must return a Boolean value, and only nodes where the filter expression is true will be included in the extract. See Using Expressions to Define Custom Business Logic

     > **Note:**
     >
     > For generational extracts, the Node Filter is not applied to ancestors.

- **Ancestor Filter** *(Generational extracts only)*: Click **Define Expression** *fx* to open the expression builder and define an expression to filter the ancestor nodes to be extracted. Only ancestor nodes where the filter expression is true are included in the extract. This enables you to reduce the levels for a generational extract to allow it to be balanced to a lower depth when the hierarchy has too many levels. See Using Expressions to Define Custom Business Logic
  Optional: Click the **X** to remove the Node or Ancestor Expression Filter.
- **Filter Logic** *(Incremental extracts only)*: Select whether to use the filter logic as an inclusion filter (that is, only nodes that match the filter are included in the extract), or an exclusion filter (nodes that match the filter are excluded from the extract).
- **Reconnect Parents** *(not available for Link extracts)*: Select whether or not to reconnect parents in the extract. See Reconnecting Parents In Extracts.

  > **Note:**
  >
  > For generational extracts, **Reconnect Parents** is always enabled and cannot be changed.

- **Exclude Implicitly Shared Nodes** (available for extracts from hierarchy viewpoints only, and not available for Link extracts): Select whether or not to exclude nodes that have already been processed under the same parent in a different location in the hierarchy. See Understanding Shared Nodes.

  > **Tip:**
  >
  > You may want to avoid excluding implicitly shared nodes when you are extracting positional information, such as ancestors, inherited values, or default derived values that use positional logic.

3. Define the File Format options:

Table 30-3   File Format Options

Field	Description	Example
Delimiter Characters (Text file type extracts only)	Character that separates text strings. Select a value, or select **Specified** and enter the delimiter characters in the **Delimiter** field. You can enter more than one character as a delimiter up to a maximum of 10, and you can use whitespace characters such as Space. **Note:** Leading and trailing whitespace characters are supported. For example, adding a space after a comma delimiter results in the extract displaying "A, B, C" instead of "A,B,C".	`Comma`
Line Feed Style (Text file type extracts only)	Character that advances to the next line	`CR+LF` (Windows) `CR Only` (Mac) `LF Only` (Linux)

Table 30-3 (Cont.) File Format Options

Field	Description	Example
Quote Columns	Columns that require a quote character: - `None` - `All Columns` - `As Needed By Value`: (Default) adds quote character to columns that contain special characters such as: – Delimiter character – Quote character – Line feed character **Note:** Quoting options are not applied to custom headers and footers.	`None`
Quote Character	Character used to indicate quote columns	`Single Quote`
Null Keyword	Keyword to replace null values in the file	`Blank`
Defined Empty String Keyword	Keyword to replace defined empty values in the file	`No Data`
Top Node Keyword	Keyword to use for the parent column property values for the top nodes in the hierarchy. If this is not provided, then a blank value is output in the parent column for top nodes.	`Root`
Pivot Delimiter	For string data type properties only, the character that separates the strings to pivot on. Select a value, or select **Specified** and enter the delimiter character. You can enter more than one character as a delimiter up to a maximum of 10, and you can use whitespace characters (such as Tab).	`Comma`

Table 30-3 (Cont.) File Format Options

Field	Description	Example
Include Column Header Record	Indicator to include the header record at the beginning of the file	`Selected`
Include Custom Header	Select to include a custom header record If selected, enter a custom header to include. You can insert substitution parameters in custom headers. See *Using Substitution Parameters in Custom Headers and Footers*, below. If you are also using the generated column header record, specify if the custom header should be placed before or after the generated column header.	`BOF` (Beginning of file indicator)
Include Custom Footer	Select to include a custom footer record If selected, enter a custom footer to include. You can insert substitution parameters in custom footers. See Using Substitution Parameters in Custom Headers and Footers, below.	`EOF` (End of file indicator)

4. Define the Data Format options:

Table 30-4 Data Format Options

Field	Description	Example
Negative Sign	Character that indicates a negative value	`Parenthesis`
Thousands Separator	Character that separates thousands in numbers	`Comma`

Table 30-4 (Cont.) Data Format Options

Field	Description	Example
Decimal Separator	Character that separates decimals in numbers **Note:** You cannot use the same character for both separators. When you select a character for one separator, that character is removed from the drop down menu for the other separator.	`Dot`
Boolean Values	Specify whether to extract the actual value, the displayed value, or specified values for boolean properties. **Note:** If you select `Specified`, enter the values to represent True and False. You can enter blank values for either option.	`Specified` - Specified Value for True: `IC_INTERCOMPANY_YES` - Specified Value for False:
Allowed Value Properties	Specify whether to extract the actual value or the displayed value for an allowed value property.	`Display`
Time Zone	Time zone for the extract	`UTC`

Table 30-4 (Cont.) Data Format Options

Field	Description	Example
Date Format	Select a date format from the drop down menu, or select `Specified` and enter a date format. **Note:** See Date Formatting Symbols.	`MM/dd/yyyy`
Time Format	Select a time format from the drop down menu, or select `Specified` and enter a time format. **Note:** The **Example Date/Time** field displays the date and time format that you select or specify.	`HH:mm:ss`

5. Click **Save**.

> **Note:**
>
> The **Time Zone**, **Date Format**, and **Time Format** are saved at the extract level. This means that the time zone and formats in the extract will always be the same regardless the locale of the user who runs it.

Using Substitution Parameters in Custom Headers and Footers

You can enter substitution parameters in custom headers and footers. When you run the extract, the value for the parameter is inserted into the extract. For example, if you enter `<%Date%>` in a custom footer, when you run the extract the date that the extract was run is inserted into the footer.

> **Tip:**
>
> Substitution parameters are not case-sensitive.

The following substitution parameters are available:

- `<%Date%>`: Returns the date the extract was run, based on the extracts settings for time zone and format . If the time zone is not set, then UTC is used.
- `<%Time%>`: Returns the time the extract was run, based on the extracts settings for time zone and format . If the time zone is not set, then UTC is used.
- `<%Timestamp%>`: Returns the combination of the date and time the extract was run, based on the extracts settings for time zone and format .
- `<%Username%>`: Returns the username performing the extract.
- `<%UserFirstName%>`: Returns the first name of the user performing the extract.
- `<%UserLastName%>`: Returns the last name of the user performing the extract.
- `<%UserEmail%>`: Returns the email address of the user performing the extract.
- `<%ExtractName%>`: Returns the name of extract.
- `<%FileName%>`: Returns the extract file name.
- `<%RecordCount%>`: Returns the count of records extracted. This count excludes the header and footer rows in the record count, but it does include pivoted rows so that it matches the last record count used.

  > **Note:**
  >
  > This parameter is available in custom footer rows only. You cannot insert the record count in a custom header.

- `<%AppVersion%>`: Returns the Oracle Fusion Cloud Enterprise Data Management release number (for example, "Oracle EDM Cloud Release 22.09").

Reconnecting Parents In Extracts

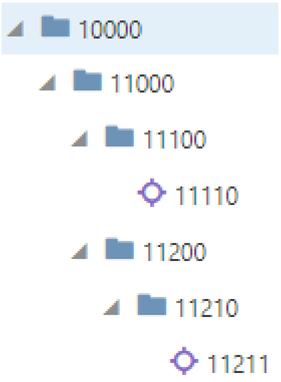
For hierarchy viewpoints only, if you have filtered out parent nodes in an extract, the **Reconnect Parent** option modifies the values for parent properties of child nodes to the nearest unfiltered ancestor.

The **Reconnect Parent** option is available only if the viewpoint for the extract is a hierarchy and an extract filter has been created. See Specifying Extract Options.

Caution:

The **Reconnect Parent** option modifies the extracted parent values but it does not recalculate other positional information, such as relationship-level properties, inherited values, or derived default values that use positional information (parent, ancestors, location, and other relationship-level properties.) Because it modifies the parent property only, other positional properties (for example, Level) may become inconsistent.

Consider the following hierarchy of nodes:



- 10000
  - 11000
    - 11100
      - 11110
    - 11200
      - 11210
        - 11211

The hierarchy has the following parent relationship properties:

Parent	Node
	10000
10000	11000
11000	11100
11100	11110
11000	11200
11200	11210
11210	11211

If you filter out nodes **11000** and **11200** in your extract and you do not enable the **Reconnect Parent** option, the **11000** and **11200** nodes are not extracted but the values for the parents for the extracted nodes are not modified. So, for example, node **11100** still has node **11000** as its parent.

Table 30-5 Reconnect Parent Disabled

Parent	Node
	10000
11000	11100
11100	11110
11200	11210
11210	11211

When you enable **Reconnect Parent**, the values for the parent properties for the extracted nodes are modified to the next-closest ancestor node contained in the extract. For example, node **11100** now has node **10000** as its parent.

Table 30-6 Reconnect Parent Enabled

Parent	Node
	10000
10000	11100
11100	11110
10000	11210
11210	11211

Running Extracts

Depending on your permissions, you can run public and private extracts to a file or a global connection.

Your Goal	Watch This Video
Learn how to run an extract	▶ Running Extracts

Considerations for Running Extracts

- The sort order for nodes and hierarchical relationships in extracts is based on the grouping and sorting options on the hierarchy set. See Creating Hierarchy Sets.
- You can run extracts on a dimension with an Active status only.
- If a viewpoint for an extract has been archived, a warning icon is displayed. You cannot run an extract on an archived viewpoint.

Private Extracts

Considerations

- You must have access to a view and the *Participant (Read)* permission or better on a viewpoint to run a private extract.
- You run private extracts from the Viewpoint Extract report, and they can run to a file only.

To run a private extract:

1. From the Viewpoint Extracts report, click **Run Extract to File** ⤓, enter a file name (or accept the default), and then click **Run**. See Extracts Reports.
2. **Optional** (Incremental extracts only): Modify the default From and To dates for the extract, and then click **Run**.

> **Note:**
>
> This overrides the From and To dates for the current incremental extract only. It does not change the default values on the extract itself. Click **Remove Override** ⏻ to restore the original values from the extract.

Public Extracts

Considerations

- You must have *Data Manager* permission or better on a dimension to run a public extract.
- You can run public extracts from the Applications list, the Viewpoint Extracts report, a REST API, or EPM Automate.
- You can run public extracts to a file or a global connection.

To run a public extract from the Application list:

1. From Applications, in the Actions column, click ⋮, and then select **Extract**.
2. In Extracts, select the extract (indicated by ↗) in the dimension (indicated by ❋) that you want to run.

   Click the ▶ button to hide or display the extracts for each dimension.

   > **Note:**
   >
   > The date and time that the extract was last run is displayed under each extract.

3. In **Extract Target**, select an option:
   - **Connection**: Run the extract to the global connection specified in the extract definition. (Available only if a global connection is defined for the extract).
   - **File**: Run the extract to a text file. You can edit the name of the text file to extract to.
4. **Optional** (Incremental extracts only): Modify the default From and To dates for the extract.

   > **Note:**
   >
   > This overrides the default From and To dates for the current incremental extract only. It does not change the default values on the extract itself. Click **Remove Override** ⏻ to restore the original values from the extract.

5. **Optional**: Click the extract, dimension, or view and viewpoint to inspect or edit those objects.
6. Click **Run Extract.**
   A status message is displayed at the top of the page.

You can also run extracts using EPM Automate (see extractDimension in *Working with EPM Automate*), a REST API (see Extracting Dimension Viewpoints in *REST API for Oracle*

Enterprise Data Management Cloud Service), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

Reviewing Extract History

You can view the history for public extracts only. You cannot view the history for private extracts.

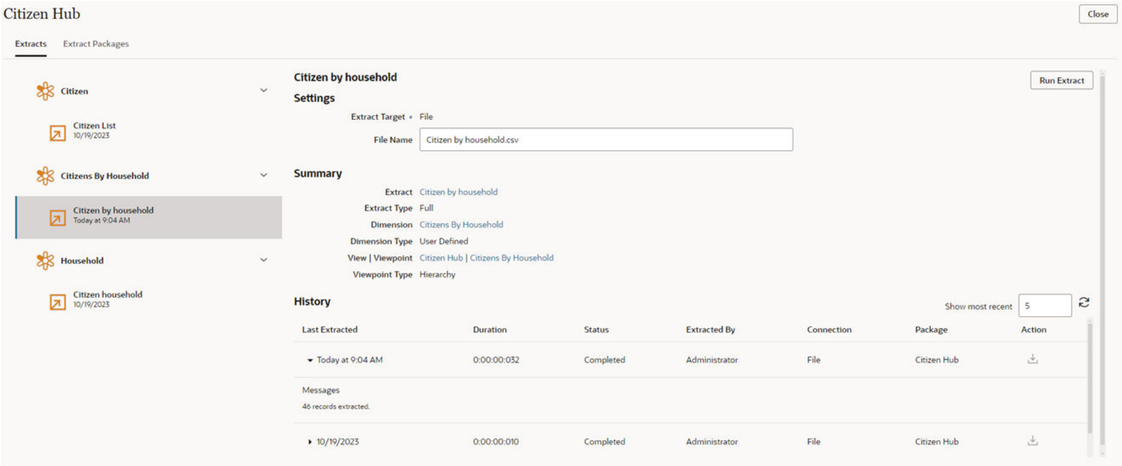
On the extract screen, review the **History** section to see the dates and times of the most recent extracts that were run for this dimension. By default, the last five are displayed. You can adjust the number of extracts to display (up to 50) in the **Show most recent** field. Click ⟳ to update the table.

If an extract was run as part of an extract package, the name of the package is displayed.

In the **Action** column, click **Download to File** ⤓ to download the extract file from the server. This can be useful if your browser times out during long extracts or to compare extract results to previous extract files. You can download extract files that were run to file only. If the extract was run to a global connection, it is not available to download from the Extract screen.

If the file is no longer available on the server, a message is displayed.

> **Note:**
>
> Extracts that were run from a REST API are not available to be downloaded from the extract screen. Use EPMAutomate or a REST API to download those extract files from the staging area. See Download a File from the Staging Area in *REST API for Oracle Fusion Cloud Enterprise Data Management*.



Citizen by household

Settings

Extract Target • File
File Name: Citizen by household.csv

Summary

Extract: Citizen by household
Extract Type: Full
Dimension: Citizens By Household
Dimension Type: User Defined
View | Viewpoint: Citizen Hub | Citizens By Household
Viewpoint Type: Hierarchy

History

Show most recent: 5

Last Extracted	Duration	Status	Extracted By	Connection	Package	Action
▾ Today at 9:04 AM	0:00:00:032	Completed	Administrator	File	Citizen Hub	⤓
Messages 46 records extracted.						
▸ 10/19/2023	0:00:00:010	Completed	Administrator	File	Citizen Hub	⤓

Working with Extract Packages

Extract packages enable you to run multiple extracts from an application with a single operation. When you run the extract package, each extract in the package is run and the results from each of them are added to a single zip file. Optionally, you can combine some or all of the extracts into a single output text file.

Example Use Cases

- The ability to combine multiple extracts into a single file enables you to create sectional files such as those used for the Oracle Hyperion Financial Management .APP files or Oracle Data Relationship Management import files.
- Additionally, because not all extracts in a package have to be combined you can create an extract package that creates a single zip file with the relationships from all of your hierarchy files combined into a single output text file and your list file with all of your node level properties as a separate output text file.

Considerations

- While individual extracts are at the dimension level, extract packages are at the application level. This means that you can create an extract package that contains extracts from multiple dimensions in an application.
- You cannot include extract packages in other extract packages.
- You can run an extract package to a global connection or a file.
- Only extracts of the same file type (text or Excel) can be combined into a single file, and an extract package can only contain one combined file. For example, if your extract package contains two Excel files and two text files, you can combine either the Excel or the text files into a single file, but not both. The other files can still be included in the extract package as separate files, but they cannot be combined.

Best Practices for Combined Extract Files

When combining multiple extracts into a single output text file, consider the following best practices:

- Each individual extract that will be combined into a single output file in the extract package should use the same file and data format options (such as the delimiter character, line feed style, and date and time formats, see Specifying Extract Options).
- If you are creating a single section from multiple extracts in a combined file, you should select the same columns to extract (see Selecting Extract Columns) in each extract.
- Use the extract package header and footer options (see Creating, Editing, Copying, and Deleting Extract Packages) that are appropriate for the intent of your output file. For example, if you are creating a sectional file with one extract with relationship information and another extract with node information, you may retain the individual extract header information on each extract. But if you are including three extracts into a combined relationship file, you may want to retain the header information only on the first extract (excluding the header information on the other two extracts) and the footer on the last extract file.

Creating, Editing, Copying, and Deleting Extract Packages

You create an extract package from the Extract Packages tab of the application inspector.

Creating Extract Packages

You must have *Owner* or *Metadata Manager* permission on at least one dimension in an application in order to create an extract package for that application.

1. Inspect the application that you want to create the extract package for. See Inspecting Applications.
2. On the Extract Packages tab, click **Create**.

3. Enter a name and optionally a description for the extract package, and then click Create. The extract package is displayed in the inspector.

Editing Extract Packages

You must have *Owner* or *Metadata Manager* permission on all of the dimensions in an extract package in order to edit that package.

After you create an extract package, you edit it to change the name or description and to specify extract package details, such as which extracts to include in the package, connection information, and extract combination options.

1. On the General tab, click **Edit** and change any of these Summary settings:
   - Name or description of the extract package
   - **Zip File Name** (must include the .ZIP file extension).
   - **Zip Package**:
     - **Enabled**: The extract package is always zipped, even if it is a single file.
     - **Disabled**: The extract package is zipped as follows:
       - **Extract package run to a file**: If the package creates a single file, the output is not zipped.

         > **Note:**
         >
         > If the package creates multiple files the output is always zipped, regardless of the Zip Package setting.

       - **Extract package run to a connection or to the staging area**: The extract package output is not zipped.
2. In **Connection**, specify a global connection name and any parameters for that connection. The parameters that you specify are specific to the connection type.
   You can test the connection before saving. The **Test Connection** button is enabled only if you have set up a global connection and defined the required parameters. See Creating Global Connections.

   > **Note:**
   >
   > The global connection must be supported by the application that you are creating the extract package for.

3. Select **Combine Extracts** if you want to have one or more extracts in the package combined into a single output file. You select which extracts to combine on the Extracts tab. Only extracts of the same file type (text or Excel) can be combined into a single file, and an extract package can only contain one combined file. The other files can still be included in the extract package as separate files, but they cannot be combined.

> **Note:**
>
> Selecting this option does not automatically combine all of the extracts in the package together. Use the **Combine Extract** check box on the Extracts tab to specify which extracts you want to combine.

After you select Combine Extracts, specify a name and, optionally, a header and footer for the combined file.

4. On the Extracts tab, click **Add** and then select the extracts to add to the package from the drop down menu. You can select any public extract in any dimension in the application that you have *Owner* or *Metadata Manager* permission on. A check mark is displayed next to each extract that you select. Click the extract again to remove your selection, and click outside of the menu to finalize your selections.

   > **Note:**
   >
   > If the viewpoint for an extract has been archived, a warning icon is displayed. You cannot save an extract package that contains an extract that uses an archived viewpoint.

5. For each extract that you want to combine into a single output file, select **Combine Extract**. The extracts are output in the file in the order in which they are listed in the inspector.

   > **Note:**
   >
   > This option is available only if you have selected **Combine Extracts** on the General tab.

   - **To reorder the extracts in the package**: Click **Actions** ⋮ and use the reorder options.
   - **To remove an extract from the package**: Click **Actions** ⋮ and select **Remove**.

6. **Optional**: For extracts with the Excel file type, by default the extract package uses the sheet names that are defined for the extracts. You can enter a different sheet name in **Package Sheet Name**.
7. **Optional**: After you select Combine Extract, enter a **Package Extract Header** to be displayed before the combined file, a **Package Extract Footer** to be displayed after the combined file, and then optionally use the **Exclude Original Extract Header** and **Exclude Original Extract Footer** options to exclude the column or custom header and custom footer of the original extracts.
8. Click **Save**.

Copying Extract Packages

You must have *Owner* or *Metadata Manager* permission on all of the dimensions in an extract package in order to copy that package.

To copy an extract package:

1. From the Extract Packages tab of the application inspector, click **Actions** ⋮ in the Action column of the extract package that you want to copy, and then select **Copy**.
2. Enter a name for the copied extract package, and then click **Copy**.
3. **Optional**: Edit the extract package to specify extract package details, such as which extracts to include in the package, connection information, and extract combination options. See Editing Extract Packages.

Deleting Extract Packages

You must have *Owner* or *Metadata Manager* permission on all of the dimensions in an extract package in order to delete that package.

> **Note:**
>
> Deleting an extract package does not delete the individual extracts that were in that package.

To delete an extract package, from the Extract Packages tab of the application inspector, click **Actions** ⋮ in the Action column of the extract package that you want to delete, and then select **Delete**.

Running Extract Packages

You can run extract packages to a file or a global connection.

Considerations

- You must have *Owner* or *Data Manager* permission on all of the dimensions in all of the extracts in the extract package in order to run it.
- All of the dimensions in all of the extracts in the extract package must have an Active status in order to run the package.
- If one of the extracts in the package fails, the extract for the entire package is failed and none of the extracts in the package are run.
- For incremental extracts that use the **Since Last Extract** option (see Editing Extracts), the context of where the extract was run from (as an individual extract or as part of a specific extract package) is taken into account:
  - When the extract is run as an individual extract (not as part of an extract package), the Since Last Extract parameter will use the last time the extract was run as an individual extract.
  - When the extract is run as part of an extract package, the Since Last Extract parameter will use the last time the extract was run as part of that specific extract package. Each extract package maintains a separate Since Last Extract value. In other words, if you run the extract as part of Package A, and then you run the extract as part of Package B, the extract in Package B will go back to the last time you ran it as part of Package B, not from when you ran it as part of Package A.
  - If no previous extract is found in a particular context (for example, the extract is run for the first time in a specific extract package), then the viewpoint creation date is used.

To run an extract package:

1. From Applications, in the Actions column, click ⋮, and then select **Extract**.
2. On the Extract page, select the **Extract Packages** tab.
3. Select the extract package that you want to run.
4. In **Settings**, select an option:
   - **Connection**: Run the extract package to the global connection specified in the extract package definition. (Available only if a global connection is defined for the extract package).
   - **File**: Run the extract package to a file. For extracts where the output is zipped, you can edit the name of the file in **Zip File Name**.

   > **Note:**
   >
   > When you the Extract Package Target is File, the output is always zipped if the package creates multiple files regardless of the Zip Package setting in the extract package inspector.

   > **Note:**
   >
   > If the extract package combines extracts into a single output file and the Zip Package setting is disabled, no zip file is created regardless of the Zip File Name parameter provided. Instead, only the combined file is output and is named based on the File Name setting in the extract package. See Editing Extract Packages.

5. **Optional**: Click the extract package name to inspect or edit it.
6. Click **Run Package**.
   A status message is displayed at the top of the page.

Reviewing Extract Package History

On the extract packages screen, review the **History** section to see the dates and times of the most recent extract packages that were run for this application. By default, the last five are displayed. You can adjust the number of extract packages to display (up to 50) in the **Show most recent** field. Click ⟳ to update the table.

You can expand an extract package history row to display the extracts that were run as part of that package as well as the duration, status, and number of records that were extracted for each extract.

Working with Viewpoint Loads

Viewpoint loads enable you to load data into viewpoints that are unbound, bound, or partially bound. You can load both bound and unbound properties. You load data into viewpoints using comma separated (CSV) or Excel files, as well as ZIP files containing CSV or Excel files.

Videos

Your Goal	Watch This Video
Learn more about how to use Viewpoint Loads to load data into viewpoints.	▶ Working with Viewpoint Loads

There are two main modes for viewpoint loads:

- **Merge**: Compares the full set of data in the load file to the existing data in the viewpoint and creates a merge request for the differences.
- **Replace**: Used to replace data in the viewpoint with data from the load file.

Both load modes require a full set of data for the viewpoint to be loaded in the load file. If you want to make incremental changes to a viewpoint, use a request file load instead (see Making Changes Using a Load File).

Caution:

When running viewpoint loads in either **Merge** or **Replace** mode, ensure that your load file contains:

- The full set of data for the viewpoint being loaded.
- The same top nodes as the viewpoint being loaded, with the following exceptions:
  - For an initial load into an empty hierarchy set, the top nodes from the load file are added to the viewpoint.
  - For a load into a viewpoint where the hierarchy set is referenced by a single node set, top nodes in the viewpoint are added and removed based on the top nodes in the load file.

You can also use viewpoint loads to import a subset of data by making a alternate viewpoint and loading it. The load will adjust the relationships in the alternate viewpoint and leave the relationships in the hierarchy set that are not part of the alternate viewpoint intact. Common relationships that are in both the alternate and main viewpoints may be affected.

Some possible use cases for using viewpoint loads include:

- Updating only one of multiple top nodes in a hierarchy set by creating a maintenance viewpoint with just that top node and loading into it. For example, in an Employees by Department hierarchy with Analytics, Marketing, and Finance as top nodes you could create a maintenance viewpoint with Finance only as the top node and then load changes just for that department.

- Updating a subset of a hierarchy from a branch down. For example, in a Geographic hierarchy you could create a maintenance viewpoint with Georgia as the top node and load changes from Georgia down.
- Adding and updating nodes with unbound properties (that is, properties other than the ones included in the dimension bindings). See Understanding Bindings and Bound Data Objects.
- Bulk updating of nodes in excess of the request item limit.

> **Note:**
>
> Ensure that the load file contains all of the relationships for all of the top nodes in the viewpoint being loaded to.

Viewpoint Load Example

Hierarchy set **HS1** is used by the following viewpoints:

Viewpoint Name	Top Nodes	View Name
CC_VP	• Legal • Managerial • Geographic	Cost Center (Default application view)
CC_MGR_VP	Managerial	Managerial (maintenance view)
CC_TX_VP	Texas (a node under Geographic)	Texas (maintenance view)

For this scenario:

- You can load into the Managerial viewpoint without losing relationship information for the other top nodes in the Cost Center and Texas viewpoints. Ensure that the load file contains all of the relationship information for the Managerial top node.
- You can load into the Texas viewpoint without losing relationship information for the top nodes in the Managerial and Cost Center viewpoints or the other Geographic branches in the Cost Center viewpoint. Ensure that the load file contains all of the relationship information for the Texas top node

Viewpoint Loads Versus Imports

Viewpoint loads are similar to imports in that both load data into dimensions. However, there are some key differences:

Viewpoint Load	Import
Can be run for any viewpoint or property regardless of binding status.	Limited to bound viewpoints and properties.
Can be used with bound, reference, and unbound dimensions.	Can be used with bound dimensions only.
Not all bound properties are required.	All bound properties are required.
Supports Replace and Merge modes.	Supports Replace, Reset, and Merge modes.
Can load from files only.	Can import from files or from external applications using a connection.

Default and Defined Loads

When running a viewpoint load, you can choose a standard set of default settings, or you can customize the load settings by defining a load.

- **Default**: Runs the load with these predefined options:

Option	Value
Delimiter Character	Comma
Line Feed Style	CRLF
Clear Value Keyword	`<clear>`
Defined Empty String Keyword	`<blank>`
Top Node Keyword	(none)
Ignore Empty Fields:	True
Expand Allowed Values	False
Process Deletes	False
Process Reorders	True
Auto Submit	True
Bypass Approvals and Commits	False

- **Defined**: Runs the load with options that you specify. See Creating, Copying, and Deleting Loads.

Load Modes

When you load data, there are two modes that you can select. The following table describes the load modes.

Load Mode	Hierarchy Viewpoint Updates	List Viewpoint Updates
Merge	• Adds nodes that are in the load source but not in the target viewpoint • Updates node property values that have different values in the load source and the target viewpoint. • For nodes that have different parents in the load source than the target viewpoint: – If shared nodes are enabled, nodes are inserted. – If shared nodes are not enabled, nodes are moved. • Removes nodes that are in the target viewpoint but not in the load source. • If the viewpoint supports a custom order, processes reorder actions. • For relationship property values: – If the relationship property value exists in the load source but not the target viewpoint, the relationship property value is updated. – If the relationship property values are the same in the load source and the target viewpoint, no update takes place. – If the load source does not contain a relationship property value, no update takes place.	• Adds nodes that are in the load source but not in the target viewpoint • Updates node property values that have different values in the load source and the target viewpoint. • Nodes are displayed in descending order by the date that they were created or updated.
Replace	• Replaces hierarchy relationships under the existing top nodes in the target viewpoint with the relationships in the load file • Hierarchy relationships in the load file that are outside of the top nodes in the target viewpoint are ignored • Retains relationship properties in the target viewpoint that are not in the load file. • Retains hierarchy relationships in the same hierarchy set that are not in the target viewpoint	• Adds nodes that are in the load source but not in the target viewpoint • Updates node property values that have different values in the load source and the target viewpoint. • Nodes are displayed in descending order by the date that they were created or updated.

Creating, Copying, and Deleting Loads

Creating a viewpoint load enables you to define the load options beyond the standard set of options that are available in default loads (see Default and Defined Loads).

You must have *Owner* or *Metadata Manager* permission on a dimension in order to create, copy, or delete a viewpoint load for that dimension.

Creating a Load

1. Inspect the dimension that you want to create a load for.
2. On the Loads tab, click **Create**.
3. In the Viewpoint drop down menu, select the viewpoint that you want to load to, and then enter a name and a description and click **Create**.

> **Note:**
>
> You can select from all active viewpoints for the dimension that are not time-labeled.

Editing a Load

After you create a load, you can edit it to specify the load options that you want to use. See Editing Loads.

Copying a Load

1. On the Loads tab in the dimension inspector, click the **Actions** menu ⋮, and then select **Copy**.
2. **Optional**: Edit the name, description, application, dimension, view, or viewpoint for the copied load, and then click **Copy**.
   The copied load is created and displayed in the inspector.

Deleting a Load

To delete a load, on the Loads tab in the dimension inspector, click the **Actions** menu ⋮, select **Delete**, and then click **Yes** to confirm.

> **Note:**
>
> Deleting a load will not delete the history associated with that load (see Reviewing Viewpoint Load History).

Editing Loads

After you have created a load, you can edit it to change the general load settings, map the columns in the load to your existing properties, and modify the formatting options. You must have *Owner* or *Metadata Manager* permission on a dimension in order to be able to edit a load.

1. On the **General** tab of the load inspector, click **Edit** to change any of these general settings:
   - Name and description of the load.
   - The view or viewpoint to run the load to. Only active, non-time-labeled viewpoints within the dimension that you are creating the load for are able to be selected. If the viewpoint has been archived, a warning icon is displayed.

- Default file name for the load file. Ensure that you enter a valid text format such as .csv, .or .zip. See Viewpoint Load File Format.

2. On the **Columns** tab, specify the mapping between the columns in the load and the properties in your dimension. See Configuring Columns in Loads.

> **Tip:**
>
> This enables you to load files that don't contain headers, or that contain column names that are different from the properties in your dimension.

3. On the **Options** tab, specify your **File Format**, **Processing**, and **Merge** options. See Specifying Load Options.
4. Click **Save**.

Configuring Columns in Loads

Use the Columns tab to configure the columns in a load when the headers in the load file do not match your property names or you want to process a subset of columns in the load file.

By default, when you run a load the column headers in the load file are matched against existing property names. For example, if your load file contains a column with a header of "Account Type", that column is matched to the property `PLN.Account Type` in your dimension.

Use the Columns tab to configure the columns in your load file for the following use cases:

Use Case	Column Configuration
The column headers in your load file do not exactly match the names of the properties in your dimension.	Use the Header column to specify the names of the columns in your load file that map to the properties in your dimension. **Note:** If some of the headers match the properties in your dimension and some do not, specify only the headers that don't match the properties. You don't need to specify the columns with headers that exactly match the properties in the dimension, as those columns will be loaded automatically as long as you do not select the **Restrict Load to Specified Columns** option.
You want to load a subset of the columns in your load file.	Specify the columns in your load file that you want to load, and then select the **Restrict Load to Specified Columns** option.
Your load file does not contain headers.	Select **No Headers in File**, and then use the position of your columns (and **Ignore Column**, if necessary) to map the columns in your load file to the properties in your dimension.
You want to override the default True value for Boolean properties.	Specify the columns to load in your load file, and then use the **Boolean True Override** field to designate the value for True for the Boolean properties in your load file.
The column headers in your load file exactly match the names of the properties in your dimension and you want to process all of the columns in your load file.	You do not need to configure columns for the load.

Considerations

- When configuring columns in a list viewpoint, you can select node level properties only. For hierarchy viewpoints, you can select node and relationship level properties.
- For load files without headers, you must add all of the columns that need to be processed, even if some of them will be skipped. For example, if your file has 10 columns and you want to process all but columns 5 and 7, you must still add all 10 columns and then define columns 5 and 7 as skipped columns.

- For load files without headers, the file can contain more columns than the ones that you specified, but the additional columns are ignored during processing. For example, if your load file contains 15 columns but you only want to process the first 10, you can add the 10 columns to be processed and the remaining columns will be ignored.

Configuring Columns

1. From the Columns tab of the load inspector, click **Edit**, and then click **Manage Properties**.
2. From the property list editor, select the properties that you want to configure columns for. Click **OK** when you are finished.
3. In the **Header** column, if the column header in the file is different from your property name, enter the name of the header in your load file that maps to the properties that you selected.

   > **Note:**
   >
   > If you select **No Headers in File**, the Header column in the inspector is informational only.

4. **For load files without headers only**: To skip a column, click **Add Column** and then select **Ignore Column**.
5. **Optional**: To remove all columns, click **Remove All**.
6. **Optional**: For Boolean properties only, use the **Boolean True Override** column to specify the value that represents a True condition in your load file. By default, True conditions are represented by either `1`, `T`, `True`, `Y`, or `Yes` values. If your load file contains True values that are represented by a different character, (for example, `X`), enter that value.
7. **Optional**: In the **Actions** column, use the **Action** ⋮ menu to move or remove a row.
8. Use the **No Headers in File** checkbox to specify that your load file does not contain headers.

   > **Note:**
   >
   > Selecting this option automatically selects the **Restrict Load to Specified Columns** option as well.

9. Use the **Restrict Load to Specified Columns** checkbox to specify the following:
   - **Enabled**: Only the columns listed in the column table are loaded. All other columns are skipped during the load.

     > **Note:**
     >
     > When **Restrict Load to Specified Columns** is enabled, `Name` must always be in the load file, and `Parent` must be in the load file when loading to hierarchy viewpoints.

   - **Disabled**: Columns listed in the column table are loaded with the headers and options set in the column table. All other columns are loaded based on their header only.

Specifying Load Options

Use the Options tab of the load inspector to specify the **File Format**, **Processing**, and **Merge** options for your load.

File Format Options

Field	Description	Example
Delimiter Character	Character that separates text strings. Select a value, or select **Specified** and enter the delimiter characters in the **Delimiter** field. You can enter more than one character as a delimiter up to a maximum of 10, and you can use whitespace characters such as Space.	`Comma`
Line Feed Style	Character that advances to the next line	`CR+LF` (Windows) `CR Only` (Mac) `LF Only` (Linux)
Clear Value Keyword	Keyword to clear the value for a property	`<clear>`
Defined Empty String Keyword	Keyword to replace defined empty values in the file	`No Data`
Top Node Keyword	Keyword that indicates that a node is a top node.	`Root`

Processing Options

Field	Description
Ignore Empty Fields	Specifies how empty fields in the load file are processed: • **Enabled**: No value is loaded for the property. (The value for the property will be null if it does not inherit a value and there is no defined or default value.) • **Disabled**: A defined empty value is stored for the property if the property data type supports defined empty values. If the property data type does not support a defined empty value, the load will display a warning and no value will be loaded for that property.

Field	Description
Expand Allowed Values	(Applies to loads in Replace mode only. Does not apply to dynamic allowed values lists.) Specifies whether or not new distinct allowed values can be added to properties with string and list data types that are set up to use allowed values when you run a load. • **Enabled**: Values in the load file that are not already in the list of allowed values for that property are added to the list of allowed values for that property. **Note:** If there are currently allowed values at the node type override level, the loaded allowed values are added to them. Otherwise, the new allowed values are added at the application override level. • **Disabled**: Values in the load file that are not already in the list of allowed values for that property are not added to the list of allowed values for that property.

Merge Options

Field	Description
Process Deletes	Specifies how delete operations are processed during merge loads: • **Disabled** (default): Removes nodes from a bound viewpoint if they do not exist in the load file. No nodes are deleted from any bound node type. • **Safe Mode**: Removes nodes from a bound viewpoint if they do not exist in the load file. The removed nodes are also deleted from bound node types only if the nodes are not used in any other hierarchy node set or hierarchy set for the dimension. • **Forced Mode**: Deletes nodes from a bound viewpoint if they do not exist in the load file. These nodes are deleted from node types used by any node set and hierarchy set in the dimension.

Field	Description
Process Reordering	If enabled, this sets the order of the nodes in the viewpoint being loaded into to the order in which they appear in the load file. **Caution:** This option may generate a large number of request items if there are significant differences between the existing order and the order in the load file.
Default Data Source	Enables you to optionally select a default data source for the data being loaded. During viewpoint load, each request item created in the load request is populated with the name of the data source that you selected. Only data sources that are unregistered and enabled can be selected. See Understanding Data Sources.
Auto Submit	Select to automatically submit merge load requests if there are no validation issues.
Bypass Approvals and Commits	Select to bypass approvals and commits for auto-submitted merge load requests.

Viewpoint Load File Format

Viewpoint loads accept comma separated (CSV) and Excel files, as well as ZIP files containing CSV and Excel files. This topic describes the requirements for those files.

Considerations

- All viewpoint load files must contain a header record with at least these required fields:
  - Hierarchy viewpoints: Name (or Node), Parent, Node Type, Parent Node Type
  - List viewpoints: Name (or Node), Node Type
- Column headers in the load file are matched against the property name, the full property name (with the namespace), and the viewpoint property label. For example, a column with the heading "Account Type" will match against either `Account Type` or `PLN.Account Type`.

> **Note:**
>
> If a column header matches on more than one property (for example, from different namespaces), an error is displayed instructing you to use fully qualified property names or viewpoint labels.

- Ensure that your load file contains the full set of data for the viewpoint being loaded.
- Only properties that are included in the header are added or updated. Existing properties in the viewpoint that are not included in the load file are ignored.
- For Excel files, the worksheet name must match the viewpoint name that you are loading into, and the first row must be the header record.
- For ZIP files, the files must be in Excel or CSV format.
  - If there is only one file in the ZIP, it will be loaded into the viewpoint regardless of the file name.
  - If there are multiple files in the ZIP:
    - If only one file ends with a viewpoint name, it is loaded into the viewpoint.
    - If multiple files end with the same viewpoint name, none of the files are loaded and an error is displayed.
- Load files should be encoded in UTF-8.

> **Tip:**
>
> To preserve Unicode characters, save your load file with the following option in Excel: `CSV UTF-8 (Comma delimited) (*.csv)`.

Running Viewpoint Loads

You run viewpoint loads from the Applications list.

Best Practice

If you are processing bound properties in a viewpoint load, it is a best practice to validate the viewpoint after the load has completed. See Validating a Viewpoint.

Considerations

- You must have at least *Data Manager* permission on a dimension in order to run a viewpoint load to that dimension.
- You cannot run viewpoint loads on time labeled viewpoints.
- Top node considerations:
  - When performing an initial load into an empty hierarchy viewpoint or when loading into a hierarchy viewpoint that is referenced by a single node set:
    - Top nodes in the load file that are not in the viewpoint are created in the viewpoint
    - Top nodes that are in the viewpoint that are not in the load file are removed.
  - For all other hierarchy viewpoints:
    - The load file must contain all of the top nodes in the viewpoint. If it does not, the load process will display an error.
    - The load file cannot contain top nodes that are not in the viewpoint. If it does, the load process will display an error.
- The load file should not contain nodes that are not under an existing top node in the viewpoint. These nodes are ignored during the load process and a warning is displayed.

- The structure under each top node in the load file must be complete. If a node under a top node in the file does not exist under the same top node in the viewpoint, it is removed from the hierarchy set, but it is not deleted from the node type.
- For a hierarchy format, nodes and hierarchical relationships are loaded and displayed based on the grouping and sorting options on the hierarchy set. You can adjust the sorting of parent and bottom nodes to match the external application by opening the hierarchy set and using the **Use Custom Order**, **Group Parent Nodes First**, and **Sort Bottom Nodes By** settings to determine the sort order for parent and bottom nodes before you run the viewpoint load. See Creating Hierarchy Sets.
- You can run viewpoint loads on active viewpoints only.
- You cannot run viewpoint loads on an application that is currently in a blockout period. See Understanding Blockout Periods.

To run a viewpoint load:

1. From the Applications list, in the Actions column click **Action** ⋮, and then select **Load**.

   > **Note:**
   >
   > If you do not have at least *Data Manager* permission on at least one dimension in the application, the **Load** menu item is not available.

2. In the left panel, select the tab for the type of load that you want to run:
   - **Default Load**: Runs the load with predefined options. See Default and Defined Loads.
   - **Defined Load**: Runs the load with options from the specified load. See Creating, Copying, and Deleting Loads.
3. Select the view and viewpoint that you want to load data into. Expand or collapse the views to display the viewpoints in that view.
4. In **Source**, click or drag to select a viewpoint load file. See Viewpoint Load File Format.
5. In **Purpose**, specify the purpose for the viewpoint load.

   > **Note:**
   >
   > For defined loads, the Purpose defaults to the load name.

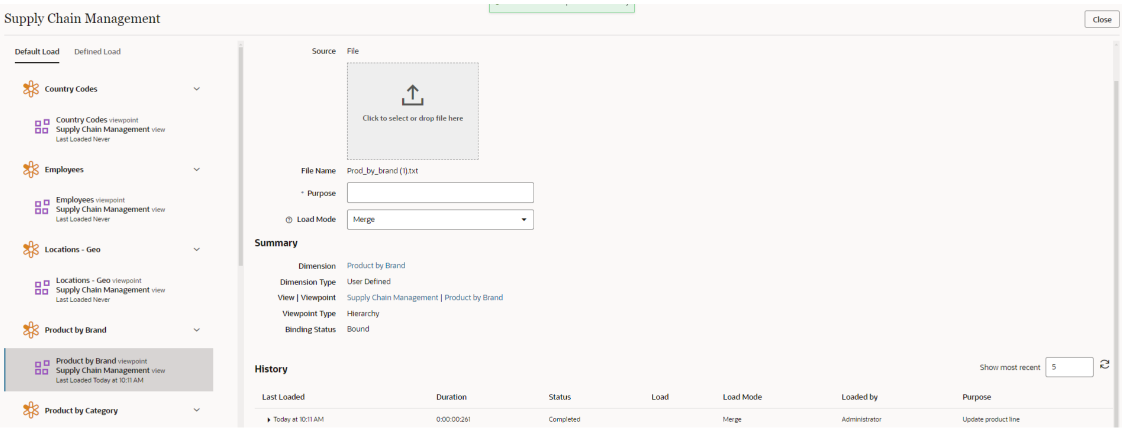
6. Specify the **Load Mode**:
   - **Merge**: Process incremental changes from the load file in a load request
   - **Replace**: Replace existing nodes with nodes from the load file.
7. Click **Run Load**.
   The viewpoint load is run and a status message is displayed at the top of the page. A record is added to the audit log that indicates the application, dimension, and viewpoint that was loaded.

Reviewing Viewpoint Load History

On the viewpoint load screen, review the **History** section to see the dates and times of the most recent loads that were run for this dimension. By default, the last five are displayed. You can adjust the number of loads to display (up to 50) in the **Show most recent** field. Click 🔄 to update the table.

> **Note:**
>
> On the **Default Load** tab, the history of all loads (default and defined) is displayed. On the **Defined Load** tab, only the history of that specific defined load is displayed.

The table displays how long it took to run each load, the status and mode of the load, the load name for defined loads, the user who ran it, and the purpose. Click the expand arrow on a row to view the number of records that were loaded and any status messages.


Supply Chain Management
Close
Default Load
Defined Load
Country Codes
Country Codes viewpoint
Supply Chain Management view
Last Loaded Never
Employees
Employees viewpoint
Supply Chain Management view
Last Loaded Never
Locations - Geo
Locations - Geo viewpoint
Supply Chain Management view
Last Loaded Never
Product by Brand
Product by Brand viewpoint
Supply Chain Management view
Last Loaded Today at 10:11 AM
Product by Category
Click to select or drop file here

Source File

File Name Prod_by_brand (1).txt

* Purpose

Load Mode Merge

Summary

Dimension Product by Brand

Dimension Type User Defined

View | Viewpoint Supply Chain Management | Product by Brand

Viewpoint Type Hierarchy

Binding Status Bound

History

Show most recent 5

Last Loaded	Duration	Status	Load	Load Mode	Loaded by	Purpose
Today at 10:11 AM	0:00:00:261	Completed		Merge	Administrator	Update product line

Using the Staging Area

Oracle Fusion Cloud Enterprise Data Management provides a staging area which is used to store files for automated import, export, and extract processes initiated by the REST API or EPM Automate in situations when direct connections to external systems are not used.

> **Note:**
>
> The staging area is restricted for use by Service Administrators only.

Source files are uploaded to the staging area prior to being imported into a dimension. Export and extract files are written to the staging area and separately downloaded to a remote machine. Files in the staging area can be listed and deleted as needed.

Files in the staging area are deleted if they have not been modified in the past 60 days. See Snapshot and File Retention Policy in *Administering Migration*.

> **Note:**
>
> Non-snapshot files in the staging area are not deleted as part of the process that deletes older snapshot files if the total size of all snapshot files is over 150GB.

The following processes use the Cloud EDM staging area:

- Import Dimension
- Export Dimension
- Export Dimension Mappings
- Export Transaction History
- Extract Dimension Viewpoint
- Import Template
- Export Template
- Load Viewpoint

The following operations are available for staging area files:

REST API

- Upload File
- Download File
- List Files
- Delete File

EPM Automate

- Upload File
- Download File
- List Files
- Delete File

Frequently Asked Questions about Imports, Exports, Extracts, and Loads

- What is the difference between an import and a load?
- When should I use an import versus a load?
- What is the difference between Reset, Replace, and Merge modes for imports and loads, and which one should I use?

What is the difference between an import and a load?

Imports and loads both bring data into Oracle Fusion Cloud Enterprise Data Management, but there are some key differences in how they operate:

Imports

- Controlled by application registration

- Used for bound data at the dimension level
- Require that all bound properties be contained in the import file.
- File format for the import is controlled by the application type and cannot be changed.
- Does not support profiles

Loads

- Focused on viewpoints, rather than dimensions.
- Can be used for bound and unbound data.
- Does not require that all properties are contained in the load file.
- Does not clear out any data chains, and restricts the changes being made to the scope of data as defined by the viewpoint and node set. This means that for hierarchy viewpoints, loads will only affect the structure below the existing top nodes.
- Supports profiles (see Creating, Copying, and Deleting Loads)

For more differences, see Viewpoint Loads Versus Imports.

When should I use an import versus a load?

Loads and imports are similar in that both load data into dimensions. Which one you select depends on several factors, such as whether you are loading data to bound or unbound viewpoints and properties, what type of load option you require (loads don't support Reset mode), and whether you are loading data from a file or an application connection. Here are some general guidelines for when to use imports versus loads:

- Use an **import** when:
  - You are loading bound data from an external system.
  - You are loading data from an application connection.
  - You want to use the Reset mode to clear out existing bound nodes and relationships and then load them from the import file.
  - You want to expand the allowed values for a property.
- Use a **load** when:
  - You want to load data to viewpoints or properties regardless of their binding status.
  - You want to load data to a reference or unbound dimension.
  - You want to load to a subset of a hierarchy viewpoint.

What is the difference between Reset, Replace, and Merge modes for imports and loads, and which one should I use?

Table 30-7 Import and Load Modes

Mode	Description and Key Features	Use Case
Reset (Import Only)	Does a complete reset of bound data including nodes, relationships, and properties. • Applies to imports only. Cannot be used with loads. • Does not update transaction history • Does not trigger subscriptions • Impacts partially bound data (such as mapping viewpoints)	Should be used only when a complete reset of the bound data is desired. **Caution**: This import mode should be used sparingly, as continuous use will increase the size of your database.
Replace	For imports, does a complete reset of the bound hierarchy set replaces it with the incoming relationships. Retains the relationship properties for relationships that existed before the import and are in the import file (even for unbound relationship level properties). For loads, clears out only the relationships in scope for the target viewpoint and node set and replaces them with the incoming relationships. Relationship level property values for relationships that existed before the load and are also in the load are retained (even for unbound relationship level properties). • Applies to imports and loads. • Does not update transaction history • Does not trigger subscriptions • Retains nodes and does not impact partially bound data (such as mapping sets).	Recommended for periodic bulk changes from external systems and large changes that do not depend on subscriptions and for which transaction history is not needed.

Table 30-7 (Cont.) Import and Load Modes

Mode	Description and Key Features	Use Case
Merge	Does not clear data sets. Creates a target set of data and compares it to the current data, and then generates a request with the changes needed to convert the current data to match the incoming import or load data. • Uses the request process, so transaction history is recorded and subscriptions are triggered. • Limited number of changes can be processed per import or load. • Enables you to specify additional options for merging data, such as whether to process Deletes and Reorders, whether to auto-submit merge requests, and whether to bypass approvals and commits. See Merge Import Options for imports and Merge Options for loads.	Recommended for periodic bulk changes from external systems and large changes where subscriptions need to be processed and for which transaction history is required.

Additional details about the specifics of how each mode works for imports and loads are available in these topics:

- Import Modes
- Load Modes

31 Migrating Across Environments

There are two methods available to migrate Oracle Fusion Cloud Enterprise Data Management applications across environments:

- *Migration* enables you to create a snapshot of your entire environment and then import that snapshot in a new environment. See Using Migration.
- *Templates* enable you to transfer the metadata for specific applications or dimensions across environments. See Working with Templates.

Migration Snapshot Considerations

Migration snapshots transfer application metadata, data (that is, nodes, relationships, and property values), and audit history across environments, including:

- Transaction history
- Requests and request history
- Users, groups, policies, and permissions
- System events in the system event log
- History of changes made to data chain objects

Because of this, migration snapshots are ideal for scenarios where you want to fully replicate a source environment in a target environment. Some examples of snapshot usage include:

- Refreshing a test environment with all of the applications, data, and history from your production environment.
- Backing up an environment for disaster recovery purposes

Templates Considerations

Templates transfer just the application metadata across environments. It does not transfer any data or audit history. This makes templates ideal for scenarios where you want to move applications but not necessarily the data in them across environments. For example:

- Migrating from a test to a production environment, where you want to transfer the application structure but not the test data
- Incrementally updating production applications with changes from a test environment

> **Note:**
>
> Templates are designed to migrate changes from one environment to another. For that reason, it is important to designate a source system where all changes are made and a target system where changes are migrated to. Do not make changes in both environments. You can use migration snapshots to resynchronize your environments if needed. See **Best Practices** in Migration Scenario Examples.

See Migration Scenario Examples for some additional ideas on when to use migration snapshots and when to use templates to migrate data across environments.

Migration Scenario Examples

This topic provides information on using migration snapshots and templates to migrate applications in different scenarios.

Best Practices

- Before you transfer applications from a source to a target environment, ensure that both environments are on the same version of Oracle Fusion Cloud Enterprise Data Management.
- When you have two environments that you are transferring applications between, ensure that you are making changes (for example, application registration changes or property creation) in only one of the environments and then transferring them to the other. Do not make changes in both the source and target environments and then try to merge them. For example, do not create a property in both environments and then try to migrate changes to that property from one environment to the other using a template. This will cause the system to create a second version of that property, as the source and target objects will not have the same ID. See Updating an Existing Application or Dimension using a Template.
- To ensure that your objects have the same IDs in both environments, you can use a migration snapshot to resynchronize them. See Scenario 4: Refreshing a Test Environment with Production Data.

Scenario 1: Moving Applications from a Test Environment to Production in Preparation to Go Live

In this scenario, you have been developing and testing in your test environment and you are ready to transfer your applications to your production environment in preparation to go live.

For this scenario, you can use a **template** to migration the application metadata. This ensures that you bring your applications over, but not any audit history or test data from your test environment that may not be relevant to your production environment.

Then, if you also want to transfer the users and groups from your source environment you could use a **migration snapshot for users and groups only** to transfer them, and if you do want to bring over the data from the source system you can use exports and imports (or extracts and loads) from your source to your target environment. For example, here is a common approach for this scenario:

1. Ensure that the target environment doesn't have any existing applications, data, or audit history in it already.

   > **Tip:**
   >
   > You can accomplish this by recreating the service in your target environment. See Recreating the Service.

2. Ensure that the users and groups are set up in the target environment.

> **Tip:**
>
> If the users and groups in your target environment are the same or very similar to the users in your source environment, you can use migration to export the **Groups and Membership** component only and import it to your target environment. This brings over your users and groups, as well as any assigned application roles. See Export Artifacts in *Administering Migration*

3. Use a template to transfer the applications in your source environment that you want to bring to your production environment. You can transfer multiple applications using a single template file. See Working with Templates.
4. If you want to bring the data from your test environment over as well, use the following to transfer data from your source to your target environment:
    - Exports and imports (for bound data only).
    - Extracts and viewpoint loads (for bound and unbound data, as well as unbound dimensions and hierarchies).
5. Perform any additional manual post-transfer tasks for objects and settings that do not get transferred with templates. These can include:
    - Application and global connection parameters such as locations, identity names, user names, and passwords.
    - Top nodes in node sets
    - Top node filters in subscriptions

    See Template Objects and Settings for more information.

Scenario 2: Updating a Production Environment with a New Application from a Test Environment

In this scenario, you have created a new application in your test environment, performed all of your acceptance testing to your satisfaction, and you are ready to load it into your production environment to roll it out to users.

For this scenario, you would most likely use a **template**. Templates enable you to move a single application and merge it into an environment that contains other applications. Keep in mind, however, that only the application metadata is transferred. If the application in your source environment also contains data that you want to transfer to your target environment, you will have to export and import (bound data only) or extract and load (bound and unbound data).

Scenario 3: Incrementally Updating a Production Application with Changes from a Test Environment

In this scenario, you have an application in production that you want to make changes to. You have made the changes in a test environment (for example, let's say you've modified some custom validations, created a new extract, and changed the formula for a derived property) and you want to migrate those changes to your production application.

For this scenario, you would use a **template** to incrementally update your production application with changes from your test environment. See Updating an Existing Application or Dimension using a Template for considerations.

Scenario 4: Refreshing a Test Environment with Production Data

When you have a production environment for your day-to-day activities and a test environment where you test new applications or new features, it can be useful to refresh your test environment periodically with information from your production environment in order to keep the test environment in sync.

For this scenario, it is most often helpful to use **migration snapshots**. This enables you to do a complete refresh of your test environment with all applications, settings, data, users, and groups from your production environment.

> **Note:**
>
> The migration snapshot of your production environment can also be used on a user acceptance training (UAT) site or as a backup on a disaster recovery site.

Using Migration

You export artifacts to create backups that can be used to restore the service or to migrate applications across environments; for example, from a test environment to a production environment or to help set up a new service instance. You can export or import all the artifacts in the service or a combination of the artifacts.

Videos

Your Goal	Watch This Video
Learn about migration.	Migrating Artifacts

Artifacts are organized into two Oracle Fusion Cloud Enterprise Data Management categories:

- The Groups and Membership category contains the Native Directory artifact which contains information about users, groups, and predefined roles.
- The Enterprise Data Management category contains the artifacts:
  - Repository Data which contains all data within the system such as applications, data chains, enterprise data, requests, and transaction history.
  - Data Grants which contains permissions that associate users and groups with the data they can access.

These are considerations for different environments when you import artifacts.

- If importing artifacts into a *new* environment with the *same* users and groups, import the artifacts in this order: Native Directory, Repository Data, and Data Grants. For example, if you are importing artifacts from a test environment to a production environment import the Native Directory first and then the Repository Data, and Data Grants.
- If importing into an *existing* environment with the *same* users and groups, import the artifacts in this order: Repository Data and Data Grants.

> **Note:**
>
> The Native Directory is not required.

- If importing into an *existing* environment with *different* users and groups, import the Native Directory first and then the Repository Data.

> **Note:**
>
> Do not import the Data Grants. After import you can define user and group permissions to access data objects.

To migrate artifacts, see Using Migration for EPM Cloud Services.

> **Note:**
>
> If you import one artifact at a time, import the artifacts in this order, Native Directory, Repository Data and Data Grants.

> **Note:**
>
> To ensure data integrity, when you perform a migration task other users are prevented from performing any operations in the service while the migration is running. A message is displayed that a service administration task is in progress, and users are returned to the home screen.

Working with Templates

Templates enable you to store application or dimension configurations in an offline file for use in other Oracle Fusion Cloud Enterprise Data Management environments. Use templates to transfer applications or dimensions and their metadata objects across environments (for example, from a test environment to a production environment) or to get a quick start in new implementations.

To use a template, first you export an application or a dimension and all of its related metadata objects in the source environment to a `.json` file, and then you import that template in the target environment.

Templates contain metadata only and do not include any data, requests, or transaction history.

Best Practice

If you are creating an application in a test environment that you plan to transfer to an existing production environment, it is a best practice to refresh your test environment with a current snapshot of your production environment first, and then create the application to be transferred. This ensures that global objects such as properties and views have the same IDs in both environments.

> **Caution:**
>
> Templates are designed to migrate changes from one environment to another. For that reason, it is important to designate a source system where all changes are made and a target system where changes are migrated to. Do not make changes in both environments. You can use migration snapshots to resynchronize your environments if needed. See **Best Practices** in Migration Scenario Examples.

Considerations

- You must be a Service Administrator or have the *Migrations - Administer* application role (see Understanding Application Roles and Permissions) to import and export applications and dimensions to templates.
- When an application is transferred to the target environment:
  - If the application owner from the source environment also exists in the target environment, they are preserved as the application owner in the target environment.
  - If the application owner from the source environment does not exist in the target environment, a Service Administrator must assign an owner to the application in the target environment.
- You can transfer new applications or dimensions from the source to the target environment, or you can transfer applications or dimensions that already exist in the target environment. The template import process updates existing applications or dimensions in the target environment by creating objects that do not already exist and modifying existing objects for the target application or dimension. See Updating an Existing Application or Dimension using a Template.
- You can transfer active applications only.
- You can exclude archived data chain objects from template imports if the archived objects in the source environment do not already exist in the target environment.

  > **Note:**
  >
  > Objects that are dependent on an archived object are also excluded from the template import if they do not exist in the target environment. If a dependent object would be partially excluded, the object will appear in the template preview or import with a warning message indicating it could not be imported.
  > For example, if you exclude an archived dimension during the import and the dimension does not already exist in the target environment, then objects such as dimension bindings, node types, policies, and extracts are also excluded from the template import if they don't already exist in the target environment.

- You can transfer multiple applications in a single template file. If you are transferring at the dimension level, you cannot transfer multiple dimensions in a single template file.
- When transferring dimensions, the application that contains the dimension must exist in the target environment.
- To transfer an application or a dimension between environments, both environments must be on the same release version.

- When importing a template, if a user or group exists in both the source and target environment, the following user and group references for the associated objects are imported:
  - Permissions
  - Policy users and groups
  - Subscription Assignees
  - View filters for users and groups

  If a user or group in the source environment does not exist in the target, the associated user or group reference are not transferred. A message is displayed indicating the missing users and groups in the target environment and the associated objects that were not imported.
- Both the source and target applications or dimensions must exist in the target environment in order to import node type converters and subscriptions. Transfer both the source and target applications in the same template file, or else ensure that you import the source application or dimension for converters and subscriptions first, so that when you import the target application or dimension the converters and subscriptions get created.
- If you are transferring a dimension with properties that use a node data type to reference other dimensions, those source dimensions must already exist in the target system. If they do not, either transfer those source dimensions first or use an application template to transfer all of the dimensions in the same template file.
- When you import the following global objects, if the object already exists in the target environment with the same name but a different ID, the source object is renamed in the target environment to make it unique:
  - Properties
  - Views
  - Lookup Sets
  - Global Connections
- After you export an application or a dimension to a template, you can review the contents of the template file in a `.json` viewer.

> **Caution:**
>
> Manually editing the template file is not supported.

To export applications or a dimension to a template:

1. From the **Tools** tile, select **Templates**.

   > **Note:**
   >
   > The **Tools** tile is visible to Service Administrators or users with the *Migrations - Administer* application role only.

2. On the **Export** tab, perform an action:
   - To export applications, select an application from the left panel. Use Ctrl/Cmd+Click to select multiple applications. The applications that you select are listed in the Selected Artifacts section.

- To export a dimension, expand the application in the left panel and select the dimension to export. You cannot select multiple dimensions in a single template file.

  > **Note:**
  >
  > To export an attribute dimension, select the base dimension for the attribute dimension.

3. In **Template File**, enter a file name with a `.json` extension, and then click **Export**. The template is created and downloaded to your local file system. The status of your export is displayed.

To import an application or a dimension from a template:

1. From the **Tools** tile, select **Templates**.

   > **Note:**
   >
   > The **Tools** and **Templates** tiles are visible to Service Administrators or users with the *Migrations - Administer* application role only.

2. On the Import tab, click **Choose File**.
3. From your local file system, select a file with a `.json` extension, and then click **Open**.
4. **Optional**: Choose one or more **Import Options**:
   - **Replace Permissions**: Replace existing permissions in the target environment with permissions from the source template.
   - **Replace Policy Groups**: Replace existing policy groups in the target environment with policy groups from the source template.
   - **Replace Property Node Type Overrides**:
     - **Enabled**: Existing property node type overrides in the target environment are replaced by property node type overrides from the source template.

       > **Tip:**
       >
       > This enables you to remove node type overrides in the target environment if they have been removed in the source.

     - **Disabled**: Existing property node type overrides in the target environment are merged with property node type overrides from the source template.
   - **Exclude Archived Objects**: Excludes archived objects in a template from a source environment from being imported into the target environment if the source objects do not exist in the target environment.

     > **Note:**
     >
     > Archived objects in a source template which already exist in the target environment cannot be excluded from template preview or import.

See Template Objects and Settings.

5. Click **Preview** to identify the objects in the template file and view the status of each. See Previewing Template Imports and Reviewing Import Results.
6. Click **Import**, and then click **Yes** on the confirmation message.
   The template is imported, and the status of each object in the template is displayed. See Previewing Template Imports and Reviewing Import Results.

Updating an Existing Application or Dimension using a Template

If the application or dimension that you are importing already exists in the target environment, that application or dimension is updated during the template import process. New objects are added, and existing objects are modified by the objects and settings in the import file.

For example, if you add a node type in an application in a Test environment, you can use a template to add that node type to the application in your Production environment.

Considerations

- An application or dimension is updated only if it was created in the same environment as the source application or dimension and migrated to the target either from a template or a migration snapshot. If you have applications or dimensions with the same name in your source and target environments that were not created in the same environment, they are not considered the same application or dimension.
- Source and target objects are matched by ID, not by name.
  - If a source object has the same name but not the same ID as an existing object in the target, the source object is renamed using an incrementing suffix during the import process and a message is displayed.
  - If a source object has the same ID but a different name than an existing object in the target, the target object is renamed with the source name as long as it does not conflict with an existing name in the target environment. If it does, the current target name is retained and a message is displayed.
- Objects that are archived or deleted in the source environment will not be removed from the target environment.

Template Objects and Settings

The following table lists the expected behavior for all of the objects and settings in a template when applications or dimensions are exported from the source environment, and then imported and either added or updated in the target environment (depending on if the application or dimension already exists in the target environment).

When the application or dimension being imported already exists in the target environment, the application or dimension is updated with the information in the template file. (See Updating an Existing Application or Dimension using a Template). During the update, the following actions are possible depending on the object type:

- **Overwrite**: The settings or parameters in the target application or dimension are overwritten with the settings or parameters from the template file. For example, if the template file contains a different name for an application than the target application, the name is overwritten with the name from the template.
- **Merge**: New objects in the template file are added to an existing set of those objects in the target application or dimension, and the existing set of objects are retained. For example, if

the template file contains a new dimension and node type, those objects are added to the existing dimensions and node types in the target application.

- **Retain**: The object, setting, or parameter in the target application or dimension is retained even if a new one exists in the template. This is primarily used when the object is not included in the template file. For example, top node filters for subscriptions are retained in the target application because they are not included in the template file.
- **Replace**: The object, setting, or parameter in the target application or dimension is replaced with those from the template file.

Table 31-1 Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Applications	Active applications are exported. **Note**: The **Disable Workflow** setting at the application level is not exported.	New applications are added.	General application parameters such as name, description, and external system type and label (Universal applications only) as well as application registration information and application blockout periods for existing applications are overwritten.
Data sources	Unregistered data sources are exported. If you are exporting an application, registered data sources for that application are exported.	Registered and unregistered data sources are added.	New data sources are merged. General data source parameters such as name, description, and code are overwritten.
Properties	For sequence properties, the last sequence value is not exported from the source environment. Sequence values in the target system will be generated based on the sequence in the target environment (or the sequence Starting Value if that sequence has not been incremented). See Working with Sequence Properties.	Property is created. Property overrides are set for the application. If a target property already exists with the same name but a different ID, the source property is renamed in the target environment to make it unique.	New properties are merged and ordered at the end of the property list. The following property attributes are overwritten: • Labels • Descriptions • Base parameters • Application and node type overrides • Sort order • Required flag
Property Groups	Property groups in views and viewpoints for the selected application are exported.	Property groups are created at the view level and properties are assigned to groups at the viewpoint level.	New property groups are merged. Existing property groups are updated at the view and viewpoint level.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Lookup Sets	Only lookup sets that are used by properties or node type converters for the application are exported. When you export a dimension, all lookup sets that are used by the application that contains that dimension are exported to the template file.	Lookup sets are created. Lookup values are set for the application. If a target lookup set already exists with the same name but a different ID, the source lookup set is renamed in the target environment to make it unique.	New lookup sets are merged. General lookup set parameters, such as name and description, and lookup values are overwritten.
Global Connections	• Only global connections that are used by extracts for the application are exported. • For security reasons, the location, identity name, user name, and password of a global connection are not exported.	Source global connections in the template file are matched to global connections in the target environment. If a target connection already exists with the same name but a different ID, the source connection is renamed in the target environment to make it unique. After the global connection is created in the target environment, you must edit it to add the location, identity name, user name, and password. See Creating Global Connections.	New global connections are merged. General global connection parameters such as name and description are overwritten. The location and credentials of the global connections in the target applications are retained, because that information is not included in the template.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Application Connections	For security reasons, the location, identity name, user name, and password of a global connection are not exported.	You must modify the application registration in the target environment to add the location, identity name, user name, and password for the connection.	New application connections are merged. General application connection parameters such as name and description are overwritten. Primary flag is also overwritten if the template has it set on a different connection than the target, or retained if the template does not have it set on a connection and the target does. The location and credentials of the application connections in the target applications are retained, because that information is not included in the template.
Views	Only views that contain viewpoints for the selected application are exported, and only the viewpoints for the selected application are exported. Active and archived views are exported.	If a target view already exists with the same name but a different ID, the source view is renamed in the target environment to make it unique. For view filters, if a source user or group does not exist in the target environment, the view filter is not imported and a message is displayed indicating the missing user or group.	New views, viewpoints, and permissions are merged. General view parameters such as name and description are overwritten. Show on Views page setting: If the filter type is the same, new users and groups are merged. If it is different, the target is overwritten with the filter from the template.
Time labels	When you export an application, all of the time labels in the views for that application are exported.		New time labels are merged. General time label parameters such as name are overwritten.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Dimensions	Active and archived dimensions are exported. Attribute dimensions for the base dimension are exported.	Active and archived dimensions are created. Attribute dimensions are created.	New dimensions are merged. New attribute dimensions are merged. General dimension or attribute parameters such as name and description as well as import and export options are overwritten.

Table 31-1   (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Node Types	Active and archived node types are exported. The following objects are exported for each node type: • Properties and property overrides • Node type converters • Custom validations • Permissions and policies	Node types are created for each dimension, along with the following objects: • Properties are added to each node type, and property overrides are set for each node type. • Node type converters are created if both the source and target applications exist in the target environment. **Tip**: When migrating applications with node type converters, migrate the source application for the converter first so that when you migrate the target application for the converter, the node type converter is created. • Custom validations are created for each node type. • Permissions and policies are created for each node type if the source user or group exists in the target environment.	New node types, node type converters, and custom validations are merged. General node type parameters such as name and description are overwritten.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Hierarchy Sets	Active and archived hierarchy sets are exported. The following objects are exported for each hierarchy set: • Custom validations • Permissions and policies	Hierarchy sets are created for each dimension, along with the following objects: • Custom validations are created for each hierarchy set. • Permissions and policies are created for each hierarchy set if the source user or group exists in the target environment.	New hierarchy sets, node types, and custom validations are merged. Matching child node types are updated. General hierarchy set parameters such as name and description, as well as the Shared Nodes and Custom Order settings are overwritten.
Node Sets	Active and archived node sets are exported. **Caution:** Top nodes are not imported.	Top nodes are not imported.	New node sets and node types for lists are merged. General node set parameters such as name and description are overwritten. Top nodes are retained because they are not included in the template file.
Viewpoints	Active and archived viewpoints are exported. Normal and time labeled viewpoints are exported. Only viewpoints for the selected application are exported.		New viewpoints are merged. General viewpoint parameters such as name and description, as well as allowed actions, time labels, and editable properties are overwritten.
Compare Profiles	Compare profiles for applications or dimensions containing the From or To viewpoint in the profile are exported. **Note:** The From and To node parameters in a compare profile are not exported.	Compare profiles are created when both the From and To viewpoints exist in the target environment.	New compare profiles are merged. Compare parameters such as name and description, as well as the compare scope and node filters are overwritten.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Queries	Saved public and private queries are exported when you export the application or dimension that contains the queries. The **Selected Node** parameter is not exported.	Public queries are imported. Private queries are imported only if the user exists (by username) in the target system.	New public queries are merged. New private queries are merged only if the user exists (by username) in the target system. General query parameters such as name, description, and code are overwritten. Private queries that were promoted to public in the source will be promoted to public in the target. If the viewpoint for a query does not exist in the target, the query is not imported. **Selected Node** parameters for existing queries are retained in the target environment.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Subscriptions	Subscriptions for target viewpoints are exported. **Note:** If you are exporting an application that contains viewpoints that are the source of a subscription, the subscription is not exported. **Caution:** Top node filters for subscriptions are not exported.	Subscriptions are created only if both the subscription source and target applications exist in the target environment. **Tip:** When migrating applications with subscriptions, migrate the source application for the subscription first so that when you migrate the target application, the subscription is created. Top node filters for subscriptions are not imported.	Source and target viewpoints are matched to identify existing subscriptions in target. - New subscriptions for viewpoint combinations that do not exist in the target are merged. - Existing subscriptions in the target are updated. General subscription parameters such as name and description, as well as options, action and condition filters, and inclusion properties are overwritten. Top node filters are retained. Default assignees are retained. Alternate assignees are merged.
Bindings			New non-mapping bindings are merged. Existing mapping bindings in the target application are replaced, because only one mapping binding is supported for each dimension. General binding parameters such as name and description, as well as settings, keys, and mapping keys are overwritten.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Extracts	Public and private extracts are exported.	Public extracts are imported. Private extracts are imported only if the user exists (by username) in the target system.	New public extracts are merged. New private extracts are merged only if the user exists (by username) in the target system. General extract parameters such as name and description, as well as columns, filters, and options are overwritten. Private extracts that were promoted to public in the source will be promoted to public in the target.
Loads	Loads for all viewpoints are exported	Loads for all viewpoints are imported	New loads are merged. General load parameters such as name, description, and options are overwritten.
Custom Validations			Overwritten
Constraints	When exporting applications, application, dimension, and node type constraints are exported. When exporting dimensions, only dimension constraints are exported.		New constraints are merged. Existing constraints are overwritten.

Table 31-1 (Cont.) Expected Behavior when Applications or Dimensions are Exported, Added, or Updated Using a Template

Object or Setting	When an Application or Dimension is Exported:	When an Application or Dimension is Added:	When an Application or Dimension is Updated:
Policies **Note:** When you import a template, select **Replace Policy Groups** to replace existing policy groups in your target environment with the policy groups from the source template. See Working with Templates.	Policy users and groups are exported for the following objects: • Selected application • Dimensions in the selected application • Node types in the selected application • Hierarchy sets in the selected application	If a source user or group exists in the target environment, policy users and groups are imported. If a source user or group does not exist in the target environment, the policy user or group is not imported and a message is displayed indicating the missing user or group and the associated object.	New policies and policy groups are merged. General policy parameters such as name and description, as well as policy filters are overwritten. Policy groups for existing policies are retained by default. When the **Replace Policy Groups** option is used, existing policy groups are replaced instead of retained.
Permissions **Note:** When you import a template, select **Replace Permissions** to replace existing permissions in your target environment with the permissions from the source template. See Working with Templates.	Permissions and data access are exported for the following objects: • Selected application • Dimensions in the selected application • Node types in the selected application • Hierarchy sets in the selected application • Views with viewpoints in the selected application	If a source user or group exists in the target environment, permissions and data access are imported. If a source user or group does not exist in the target environment, the permissions and data access are not imported and a message is displayed indicating the missing user or group and the associated object.	New permissions are merged. Existing permissions and data access in the target are retained by default. When the **Replace Permissions** option is used, existing permissions and data access are replaced instead of retained.

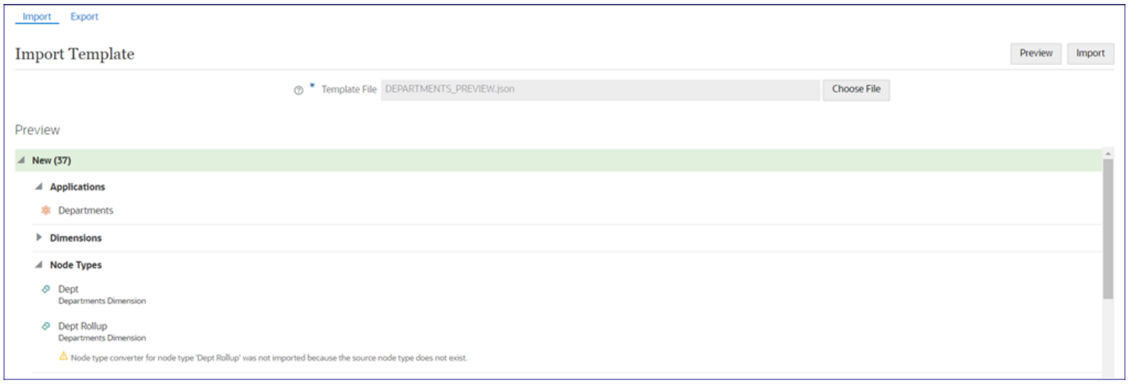
Previewing Template Imports and Reviewing Import Results

Before you import a template, you can preview it to visualize the changes that the template import will create in the target environment. After you run the import, you can review the results for each object.

Previewing Template Imports

Previewing a template import enables you to identify the objects in the template file and to determine which ones will be added, modified, unchanged, and skipped by the import process. If there are issues with some of the objects in the template file, previewing the import enables you to identify those issues and take corrective action on them before the template is imported.

In the following example, the preview indicates that a node type converter for the "Dept Rollup" node type will not be imported because the source node type does not exist in the target environment. In order to import the node type converter, you should add the "Dept Rollup" node type in the target environment before you run the import.



Import Template

* Template File: DEPARTMENTS_PREVIEW.json

Preview

- New (37)
  - Applications
    - Departments
  - Dimensions
  - Node Types
    - Dept
      Departments Dimension
    - Dept Rollup
      Departments Dimension
      ⚠ Node type converter for node type 'Dept Rollup' was not imported because the source node type does not exist.

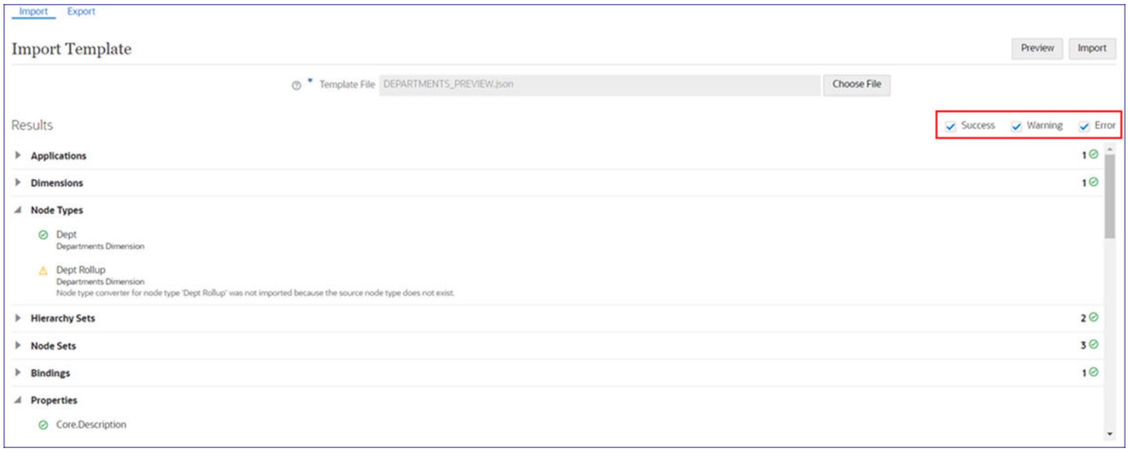
Reviewing Import Results

After you run the template import, review the results to identify the status for each of the objects in the template file:

- **Success**: The object was imported from the import file without any changes.
- **Warning**: The object was imported, but it was changed from the import file in order to import it.
- **Error**: The object was not imported.

You can enable and disable each of these filters to see just the objects that have a particular status. For example, you can view only the objects with warnings or errors.

In the following example, the node type converter for the "Dept Rollup" node type in the template was not imported because the source node type does not exist in the target environment.



Import Template

* Template File: DEPARTMENTS_PREVIEW.json

Results

- Applications 1 ✓
- Dimensions 1 ✓
- Node Types
  - ✓ Dept
    Departments Dimension
  - ⚠ Dept Rollup
    Departments Dimension
    Node type converter for node type 'Dept Rollup' was not imported because the source node type does not exist.
- Hierarchy Sets 2 ✓
- Node Sets 3 ✓
- Bindings 1 ✓
- Properties
  - ✓ Core.Description

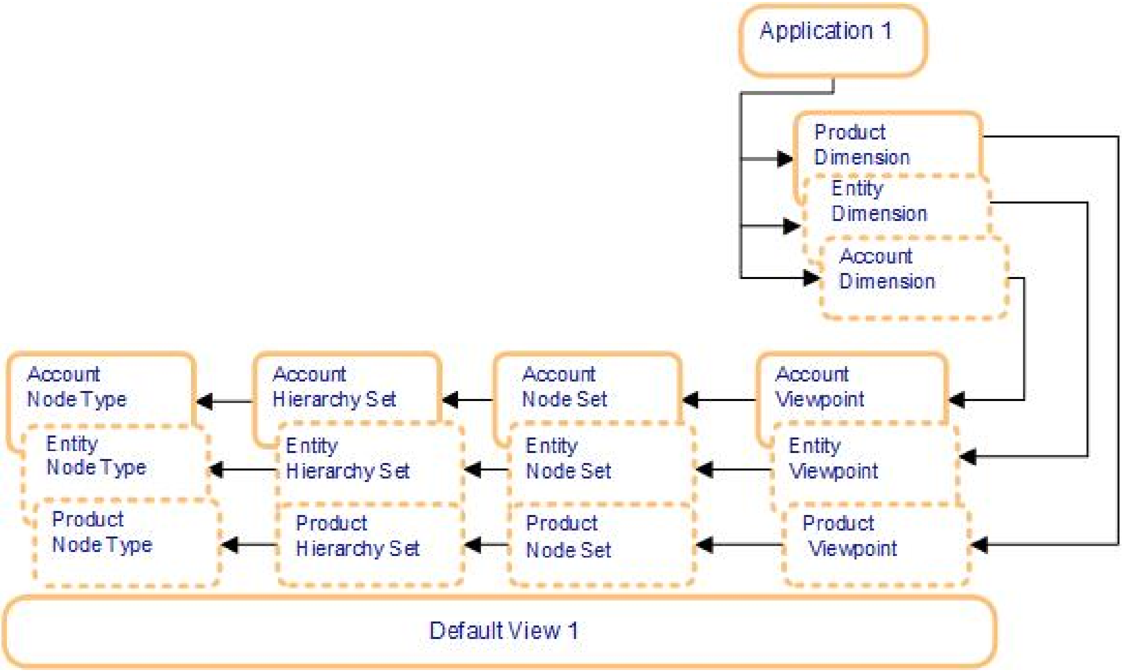
32 Working with Planning and FreeForm Applications

You register and modify Planning, Planning Modules, and FreeForm applications using a wizard that prompts you for information, such as the external application type, connection URL, and dimensions.

- **Planning** provides a flexible planning application that supports enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model.
- **Planning Modules** provides pre-built business applications that support enterprise-wide planning, budgeting, and forecasting which includes planning and budgeting solutions for Financials, Workforce, Capital, Projects, and Strategic Modeling in a cloud-based deployment model.
  Except where noted, when the topics in this chapter refer to Planning applications, they refer to Planning Modules applications as well.
- **FreeForm** provides a customizable application with full dimension and cube flexibility for multi-dimensional analysis, reporting, and planning in a managed cloud-based deployment model.

Managing data for Planning and FreeForm applications in Oracle Fusion Cloud Enterprise Data Management involves:

- Registering Planning and FreeForm Applications.
- Importing Planning and FreeForm Dimensions.
- Modifying Registered Planning and FreeForm Applications.
- Getting Started with Data Management.
- Exporting Planning and FreeForm Dimensions.



When you create a Planning or FreeForm application, the following objects get created:

- An application is created with settings, such as alias tables, multi-currency codes, and optional connection settings.
- One default view is created for each application.
- One dimension is created for each registered dimension.
- For each dimension, these objects are created and bound to the dimension:
  - One node type with all of the application specific properties.
  - One hierarchy set using the node type for both parent and child.
  - One node set that points to the hierarchy set.
  - One viewpoint that points to the node set.

> **Note:**
>
> Binding rules ensure that the registered dimension conforms to the external application dimension requirements, see Understanding Binding Rules.

In addition to registering dimensions, you can also register attributes. See Understanding Attributes.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working, you can export the dimension back to the external application.

Planning Modules Predefined Members

Planning Modules predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM.

Any changes to the predefined members must be performed in the external application. After the changes are made, you must perform a merge import (see Working with Merge Imports) to update the predefined members in Cloud EDM.

Registering Planning and FreeForm Applications

When you register a Planning or a FreeForm external application into Oracle Fusion Cloud Enterprise Data Management, an application and default view are created which represent the external application.

Videos

Your Goal	Watch This Video
Learn about registering Planning and FreeForm applications.	▶ Registering Planning and FreeForm Applications

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

  > **Note:**
  >
  > The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which dimensions you want to register. You must register all the cubes associated with each dimension. You can register three dimension types: Account, Entity, and User Defined.
- When you add a cube to a new or existing application, the data storage property for cube (for example, `PLN.Data Storage (Plan 1)`) is automatically set to be derived from the application default data storage property (`PLN.Data Storage`). See Derived Properties.

  > **Note:**
  >
  > Editing an existing cube by modifying an application's registration will not set the data storage property for the cube to be derived from the application default. If you want the value for the cube data storage property to be derived from the application default, you must set up an application override. See Editing Property Parameters.

For more information, see Understanding Registering Applications.

To register a Planning or FreeForm application:

1. From **Applications**, click **Register**.
2. Select one of these application types:
   - **Planning**
   - **Planning Modules**

- **FreeForm**

3. Enter an application name and description.

> **Note:**
>
> The application type, name and description are displayed in the Application list.

4. If you are using a local csv (comma-delimited) file for your dimension, skip the next step.

5. If you are connecting to an external application, click **Add**, and define the connection settings. You can add multiple application connections.

Field	**Valid Values**	**Example: connecting to an external application**
Connection Name	Enter a required connection name, unique to this application.	US Planning Cloud Connection
Connection Description	Enter an optional description.	Use this connection to connect to a Planning application.
Instance Location	Enter a connection URL to the external application in the format: `https://`*`InstanceName-Cloud_Account_Name.EPM_Cloud_Family.Data_Center_Region`*`.ocs.oraclecloud.com/`*`Context`*	See Sample EPM Cloud URLs in *Getting Started Guide for Administrators*
Primary Connection	Select this check box to make this the default for importing from and exporting to the external application.	
Authentication Type	Select an option: - Basic: Use the provided user credentials to authenticate. - OAuth: Use OAuth2 IDs and tokens to authenticate. (Available for environments on Oracle Cloud Infrastructure (OCI) / Gen 2 architecture only.) See Using OAuth2 in Cloud EPM Applications.	`Basic`
Basic:		

Field	Valid Values	Example: connecting to an external application
Identity Domain	Enter the identity domain of the external application (not required for OCI (Gen 2) URLs). **Note:** This is the same as the Cloud Account Name used in the **Instance Location**.	`epmid`
Username	Enter an authorized user name **Note:** The user must be a Service Administrator of the external application, and they cannot be authenticated using corporate SSO (identity provider) credentials.	`tom.smith`
Password	Enter a password.	xxxxxxxx
OAuth:		
Client ID	Client ID for the client that you created for application connections.	
Access Token	Access token for the client that you created for application connections.	
Refresh Token	Refresh token for the client that you created for application connections.	

6. Review the application summary information, and then click **Create**.

   > **Note:**
   >
   > The application and default view are created and set to draft mode. The default view name is the same as the application name you specified above.

7. Continue the registration, see Registering Cubes, Application Settings, and Dimensions.

Registering Cubes, Application Settings, and Dimensions

Use these steps to register cubes, application settings, and dimensions from Planning and FreeForm applications.

To register cubes and define application settings:

1. On the **Cubes** screen, click **Add**, then enter the cube name and then specify the cube type:
   - ASO (all application types)
   - BSO (Planning and Planning Modules applications only)
   - BSO-Hybrid (FreeForm applications only)

   > **Note:**
   >
   > The cube name is put into the `Source Plan Type` property and other related properties, such as `Aggregation, Formula, and Data Storage`.

   Add all the cubes used in the dimensions you are registering.

2. Define the application settings.

Field	Valid Values	Example
Alias Tables	Enter dimension alias tables.	Default English Spanish
Weekly Distribution Type (Planning and Planning Modules applications only)	Specify the weekly distribution type. This option is not displayed for FreeForm applications because the weekly distribution for those applications is always **Even**.	Even
Multi Currency (Planning and Planning Modules applications only)	Set this check box, if the Planning application supports multi-currency. This option is not displayed for FreeForm applications because Multi Currency is always enabled for those applications.	Select the check box

Field	Valid Values	Example
Hybrid Mode (Planning and Planning Modules applications only)	Set this check box if the Planning application supports hybrid mode. This option is not displayed for FreeForm applications because Hybrid mode is always enabled for those applications. **Note:** Select this check box to add the plan type-specific `Solve Order` properties to all BSO and ASO cubes. Clear the check box to add the plan-type specific `Solve Order` properties only to ASO cubes.	Select the check box
Currencies	You must enter at least one currency code.	`USD, CAD, INR`
Smart Lists	Enter the name of one or more smart lists. The smart list names you enter will be a selectable item in the `Smart List` property. Smart list values can be managed in the external application only.	Channel

3. Register the dimensions.

   To register one or more dimensions, click **Add**, and then enter the following information:

Field	Valid Values	Example
Dimension Name	Enter a dimension name. If this is different than the actual dimension in your external application, specify the external dimension name below.	Financial Account The value you enter becomes the name of the viewpoint, node set, hierarchy set, node type and dimension in Oracle Fusion Cloud Enterprise Data Management.
Dimension Description	Optionally enter a dimension description.	

Field	Valid Values	Example
Dimension Type	Select the dimension type: Account, Entity or Custom. **Note:** The default Dimension Type is "Account" for Planning and Planning Modules applications, and "Custom" for FreeForm applications). For FreeForm applications, if you select the Account or Entity dimension type you must have enabled the dimension and associated it with the Account or Entity dimension type in the external application. See Creating a FreeForm App Using the Application Creation Wizard in *Administering Planning*.	Account
Allow Shared Nodes	Select if the dimension you are importing has shared nodes.	

Field	Valid Values	Example
Select the valid cubes for this dimension **Note:** For each cube that you select, a set of cube-specific properties is created (for example, Plan Type).	Select all the cubes for the dimension. Use the back button to enter additional cubes if needed. **Note:** You must select at least one valid cube for all dimension types except for Custom. Selecting cubes for Custom dimensions is optional.	Plan1 Plan2
Enter the External Dimension name	Optionally enter the external dimension name. This lets you enter a different value for the dimension name above.	acc1

4. Optionally, add one or more attributes:
   - For simple attributes:
     – Select **Simple Attribute**.
     – Enter an attribute name.
     – Enter all values you want to use for the attributes.

     > **Note:**
     >
     > The attribute name is put into the `Attribute - name` property and the values are put into a pull-down list.

   - For attribute dimensions:
     – Select **Attribute Dimension**.
     – Enter an attribute dimension name.

     > **Note:**
     >
     > The attribute dimension name is put into the `Attribute Dim - name` node data type property and the viewpoint for the attribute dimension is added to the default view for the application.

     – For **Attribute Type**, select Text, Numeric, or Date.

     See Understanding Attributes.
   - For registered dimensions:

- Select **Registered Dimension**.
- Enter an attribute name.

> **Note:**
>
> A node data type property is created with the name that you enter in order to reference the registered dimension.

- Select the dimension that contains the attribute values that you want to reference from the **Dimension** drop down menu.

5. Review the summary, and then click **Apply**.
The dimension and its data chain are created. The data objects are bound to the dimension and set to active status. You can now import dimension data, see Importing Planning and FreeForm Dimensions.

Predefined Properties for Planning and FreeForm Applications

> **Note:**
>
> Predefined members are used in Planning Modules only. See Customizing Predefined Artifacts.

The following properties are predefined for Planning and FreeForm applications. All Planning and FreeForm properties use the PLN namespace.

For more information, see:

- Working with Properties
- Inheriting Properties

Property	Data Type	Inherited	Level	Description
Access Granted to Member Creator	String	No	Node	Determines the access that member creators have to dynamic members that they create
Account Type	String	Yes	Node	Defines account time balance (how values flow over time) and determines accounts sign behavior for variance reporting with member formulas
Aggregation	String	Yes	Relationship	Determines how child member values aggregate to parent members
Alias	String	No	Node	Alternate unique description for the dimension member
Attribute Dimension Name	String	No	Node	Specifies the name of the attribute dimension

Property	Data Type	Inherited	Level	Description
Base Currency	String	Yes	Node	Specifies the base currency for an entity dimension member in a Standard multiple currency application
Data Storage	String	No **Note:** When you add a cube, the data storage property for the cube (for example, `PLN.Data Storage (Plan 1)`) is derived from the application default data storage property (`PLN.Data Storage`).	Relationship	Specifies whether data is stored or calculated for a dimension member
Data Type	String	Yes	Node	Specifies the type of data values for a dimension member
Enable for Dynamic Children	Boolean	No	Node	Enables users to create children for this member by entering a member name in the runtime prompt for a business rule that has been configured with a dynamic parent member
Enterprise Predefined Member	Boolean	No	Node	Designates a dimension member as predefined by the application. Predefined members are restricted to only specific types of changes such as moves, reorders, or adding a child under them. **Note**: This property is set by the import process when importing predefined members from a Planning Modules application and it cannot be manually edited in Oracle Fusion Cloud Enterprise Data Management.
Exchange Rate Type	String	Yes	Node	Determines how exchanges rates are used to calculate values
Formula	Memo	No	Node	Member formula used to calculate a value for a dimension member.
Formula Description	String	No	Node	Description for a member formula

Property	Data Type	Inherited	Level	Description
Hierarchy Type	String	No	Node	Available for dimension bound to an aggregate storage cube. Aggregate storage dimension are automatically enable to support multiple hierarchies. The first hierarchy in a multiple hierarchy dimension must be stored.
Number of Possible Dynamic Children	Integer	No	Node	Maximum number of dynamic children that can be created for a dimension member. This option is available if Enable for Dynamic Children is selected.
Plan Type	Boolean	Yes	Relationship	Specifies whether the dimension members if valid for the plan type.
Process Management Enabled	Boolean		Node	Determines whether process management is enabled
Skip Value	String	Yes	Node	Determines how database calculations treat zeros and missing values when Time Balance is First, Balance, or Average.
Smart List	String	No	Node	Select a smart list to associate to a dimension member.
Source Plan Type	String	Yes	Node	Source plan type for a dimension member
Time Balance	String	Yes	Node	Specifies how the application calculates the value of summary time periods
Two Pass Calculation	Boolean	No	Node	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Data Storage of Dynamic Calc or Dynamic Calc and Store.
UDA	List	No	Node	User defined attribute used for calculation or reporting purposes
Variance Reporting	String	No	Node	Determines whether an account is treated as an expense during variance reporting
Weekly Distribution	Boolean	No	Node	Sets the weekly distribution. Available for bottom level account members if the option was selected when creating the application and the base time period is 12 months.

Importing Planning and FreeForm Dimensions

For Planning and FreeForm dimensions, Oracle Fusion Cloud Enterprise Data Management supports importing from comma-delimited (CSV) files and from an external application outbox using a `.zip` file.

Planning applications, FreeForm applications, and Planning Modules applications are identical in terms of being managed in Cloud EDM, except for this difference for Planning Modules nodes:

On import, the property `Enterprise Predefined Member` is set to `True` for predefined members and cannot be edited, with the exception of adding a child under it.

> **Note:**
>
> Predefined Members can be updated in Planning Modules only, see Customizing Predefined Artifacts in *Administering Planning Modules*. If the predefined members in the external application are changed, you must re-import the dimensions to update the predefined members in Cloud EDM.

Videos

Your Goal	Watch This Video
Learn about importing and exporting dimensions.	▶ Importing and Exporting Application Data

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- In the external application, create a data file to import using one of these formats:
  - If you are using a local file, a `.csv` (comma-delimited) file for each dimension. The local `.csv` filename does not have any file naming restrictions.
  - If you are connecting to the external application outbox, a `.zip` file that contains `.csv` files for one or more dimensions. The `.csv` filename must contain the dimension name or external dimension name, if specified in the registration, at the end with preceding text. For example, `acc1.csv` is invalid, but `tom_smith_acc1.csv` is valid, where `acc1` is the dimension name.

> **Note:**
>
> From Planning, create the files to import into Cloud EDM. See Exporting Metadata in *Administering Planning*.

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. Click the dimension that you want to import data into.
3. In **Input Source**, select **File** or **Connection**, and then do one of the following:
   - If you selected **File**, specify an import file name.
   - If you selected **Connection**, select a connection to import from, and specify an import file name.
4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.

5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*), a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

Import and Export File Format

When you import and export data for Planning and FreeForm dimensions, you must use a comma-delimited (`.csv`) file.

The following list describes the comma-delimited file's format and other considerations:

- The file must have the following columns:
  - **Name** The dimension name specified in the registration or the external dimension name, if you entered it.
  - **Parent** The name of the parent node. If there is no parent node, this field must be empty.
- **Alias:** *name*, where *name* is the name of a the alias table you registered. You must have at least one alias table.
- These are common columns:
  - **Plan Type** (*cube name*), **Aggregation** (*cube name*), **Data Storage** (*cube name*), **Formula** (*cube name*), **Formula Description** (*cube name*), these are typically defined for each cube you registered.
  - **Solve Order** *(cube name)* this is defined for each ASO cube you registered.

    > **Tip:**
    >
    > *cube name* is the name of the cube you registered.

  - **Smart List** The smart list names are optionally specified for nodes.
  - **Attribute Dimension Name** The name of the attribute dimension entered during registration.

    > **Note:**
    >
    > Additional columns can be imported and exported from the external application, see Dimension Properties and Member Properties.

- The import file's data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a child node's row precedes its parent node's row, an error occurs when the file is imported.
- The import ignores duplicate rows.
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)

> **Note:**
>
> The import contains the logic to process shared nodes (Data Storage = shared) based on the requirements of Planning.

Modifying Registered Planning and FreeForm Applications

After you register a Planning or FreeForm application, you can modify it. For example, you can register additional dimensions, remove dimensions, change the multi-currency setting (Planning only) or add a cube. The application, dimension, and data objects are updated based on your changes. The dimension binding rules are updated to reflect the modifications you make.

> **Note:**
>
> If you remove a cube, attribute dimension, or alias, its corresponding property is unbound from the dimension and is no longer imported or exported. For example, if you remove the Default alias dimension, the corresponding property, `Alias: Default`, is no longer imported or exported for the node type. The property remains on the node type until you remove it.
>
> If you remove a dimension, the corresponding data chain objects are unbound from that dimension but they are not deleted. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for a Planning or FreeForm application:

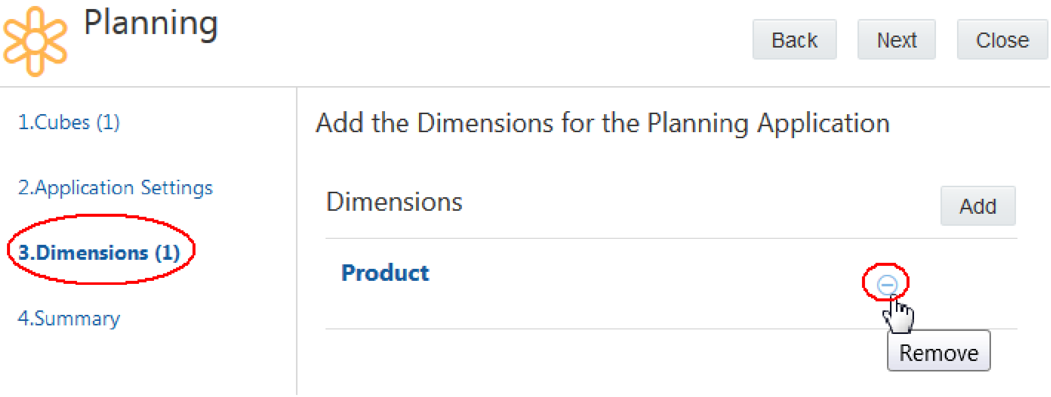
1. From **Applications**, find your application, click ⋮, and then select **Modify Registration**.
2. To change or add cubes, application settings, and dimensions, see Registering Cubes, Application Settings, and Dimensions.
3. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

   > **Note:**
   >
   > If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
   >
   > If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Removing Dimensions

To remove a dimension from the application:

1. From **Applications**, find the application, click ⋮, and then select **Modify Registration**.
2. Select **Dimensions**, and then click ⊖ .



Upgrading Planning Applications to Planning Modules

You can upgrade your Planning application instances in Oracle Fusion Cloud Enterprise Data Management to Planning Modules without having to create a new application instance. You must have the *Service Administrator* role or the *Owner* or *Metadata Manager* permission on the application to upgrade it.

Considerations

- Upgrade your Planning application instance in Cloud EDM only after you have upgraded the external target Planning system.
- When you upgrade the external target Planning application, new predefined members get added. Re-import the dimensions from your target application using the **Reset** option in order to load the new predefined members into the upgraded application in Cloud EDM. See Importing Dimensions.
- You can upgrade only active application instances. You cannot upgrade an application that has been archived.

To upgrade a Planning application to Planning Modules, from Applications click ⋮, select **Upgrade to Planning Modules**, and click the confirmation button.

> **Caution:**
>
> After you upgrade a Planning application to Planning Modules. the upgrade action cannot be undone.

Converting Planning Applications to FreeForm Applications

You can convert your Planning application instances in Oracle Fusion Cloud Enterprise Data Management to FreeForm applications without having to create a new application instance.

You must have the *Service Administrator* role or the *Owner* or *Metadata Manager* permission on the application to convert it.

> **Note:**
>
> Implementations which previously used the Planning application type to manage their FreeForm application data should convert their applications from Planning to FreeForm.

Considerations

- The Planning application must have the following values in these application settings in order to be able to be converted to a FreeForm application:
  - Hybrid Mode must be enabled
  - Weekly Distribution must be set to Even
  - Multi Currency must be enabled

  If any of these application settings have a different value than the ones listed above, the application cannot be converted to FreeForm as is. You must modify the application registration and change the application settings to have the values listed above before you convert the application.
- You can convert only active application instances. You cannot convert an application that has been archived.
- Converting Planning application instances in Cloud EDM to FreeForm applications is a one-way operation that cannot be undone. You cannot convert FreeForm applications back to Planning applications.

To convert a Planning application to FreeForm, from Applications click ⋮, select **Convert to FreeForm Application**, and click the confirmation button.

Binding Rules for Planning and FreeForm Applications

Binding rules are essential because they automatically make the Oracle Fusion Cloud Enterprise Data Management application conform to the requirements of the external application.

The following table describes the binding rules for Planning and FreeForm applications.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.
Hierarchy Set	For bound hierarchy sets, you cannot select a node type that is missing required dimension properties unless you confirm that the system may add the properties when you save.
Node Set	For bound node sets, you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. The hierarchy set is checked and then the node types are checked.

Object	Rule
Dimension	When you edit a viewpoint bound to a dimension: • You can select a different viewpoint for the binding to a dimension. The new viewpoint is checked for binding rules as are all data objects up the data chain. **Note:** You can also remove a dimension by deleting it from the application registration. The dimension still displays in the Application Inspector, and data chain objects may be created using the dimension, but it cannot be used for importing or exporting data. • In the application import and export sections, you must select a dimension that passes its own binding rules or any binding rules up the data chain.

For more information, see:

- Understanding Binding Rules
- Binding Rules for All Application Types

Binding a Viewpoint to a Dimension in Planning and FreeForm Applications

Planning and FreeForm applications support binding a viewpoint to a single external dimension. If you create an alternate viewpoint that you want to bind to the external dimension instead, you can edit the binding to point to the alternate viewpoint by taking the following steps.

For more information, see:

- Understanding Bindings and Bound Data Objects
- Understanding Viewpoints

1. From **Applications**, find the application containing the viewpoint.
2. In the application's **Actions** column, click ⋮, then select **Inspect**.
3. On the Dimensions tab, click the name of the dimension to which you are binding the viewpoint.

   The dimension's inspector displays.
4. Select Bindings, and then select the binding to associate with the alternate viewpoint.
5. Click **Edit**.

   The **View** and **Viewpoint** drop-down lists become active.
6. From **View**, select the view containing the viewpoint you want to bind to the dimension.
7. Select the viewpoint.
8. Click **Save**.

Planning and FreeForm Validations

Validations ensure data integrity. The following validations run for Planning, FreeForm, and Planning Modules applications. They all have the Planning source validation type.

In addition to these Planning and FreeForm validations, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

> **Note:**
>
> For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

The following validations have the Planning source validation type.

Name	Description	Scope
Alias Uniqueness	Alias are not unique among a group of sibling nodes	Siblings
Cube Data Storage	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only	Node
Data Storage	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only **Note:** This is not checked if the member is not in the cube.	Viewpoint
Date Attribute Name	Bottom level node names in a date attribute dimension are not in `mm-dd-yyyy` format	Node
Exchange Rate Type	Exchange Rate Type is not None and Data Type is not Currency	Node
Formula Allowed	Data Storage is not Dynamic Calc or Dynamic Calc and Store	Node
Member Name Conflicts	Member name is the same as a dimension name, a cube, or an attribute dimension defined for the current application	Node and data chain
Name Uniqueness	Member name is not unique across all nodes used in an application	Application-bound node types
Numeric Attribute Name	Bottom level node names in a numeric attribute dimension are not in numeric format	Node

Name	Description	Scope
Plan Types	Plan Type is set to True but the parent does not have a True value **Note:** This validation is skipped if the parent is a predefined member.	Viewpoint
Shared Member before Base Member	Shared members appear above base members in a hierarchy	Viewpoint
Shared Members	Shared members are not in the same dimension	Application-bound node types
Smart List Required	Smart List is missing for member if Data Type is Smart List	Node
Source Plan Type	Source Plan Type is not a valid Plan Type for the member	Node
Variance Reporting	Variance Reporting is not Expense when Account Type is Expense or Variance Reporting is Non-Expense if Account Type is anything other than Expense or Saved Assumption.	Node

The following validation runs for Planning Modules applications only. It has a source validation type of Planning Modules.

Name	Description	Scope
Predefined Member	If the Enterprise Predefined Member property equals `True` for a node, it can only be moved or reordered. You cannot edit the node or modify the Enterprise Predefined Member property on the node.	Node

Note:

In Planning Modules applications, most predefined validations are not run against predefined members (that is, nodes with `PLN.Enterprise Predefined Member` set to `True`), because these members are imported from the external application and are not able to be edited in Oracle Fusion Cloud Enterprise Data Management.

The only predefined validations that are run against predefined members in Planning Modules applications are the Predefined Member and Alias Uniqueness validations.

Note:

The Source Plan Type, Formula Allowed, and Plan Type validations are not checked on shared nodes (Data Storage = shared). The imports and exports at the application level transform these values as required by Planning and Planning Modules.

Invalid Characters, First Characters, and Values for Planning and FreeForm

When you register a Planning or FreeForm application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` and `Alias` properties. For a list of these characters and values, see Naming Restrictions for Dimensions, Members, and Aliases in *Administering Planning* .

You can edit these values by editing the property parameters. See Editing Property Parameters

Exporting Planning and FreeForm Dimensions

You can export from a viewpoint that is bound to a dimension back to a Planning or FreeForm application.

Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States, and all states. If the node set's top node is Texas, only Texas and the nodes under it are exported.

Videos

Your Goal	Watch This Video
Learn about exporting dimensions.	Importing and Exporting Application Data

Tutorials

Tutorials provide instructions with sequenced videos and documentation to help you learn a topic.

Your Goal	Learn How
Manually export dimension metadata from Oracle Fusion Cloud Enterprise Data Management to a connected Planning environment, and then import the metadata into Planning	Integrating dimension metadata from Enterprise Data Management Cloud with Planning using manual export and import
Use EPM Automate to export dimension metadata from Cloud EDM, copy it to a Planning environment, and then import the metadata into Planning	Integrating dimension metadata from Enterprise Data Management Cloud with Planning using EPM Automate
Use a Groovy business rule in Planning to export dimension metadata from Cloud EDM, and then import the metadata into Planning	Integrating dimension metadata from Enterprise Data Management Cloud with Planning using Groovy rules

Best Practice

It's a best practice to validate the viewpoint bound to the dimension prior to export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension that you are exporting from.
- When you register a Planning or FreeForm application, nodes and hierarchical relationships are imported and displayed in the order they were in the application. Nodes and hierarchical relationships are exported in the order they are in Cloud EDM. If you want nodes and hierarchical relationships to be exported in alpha numeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.
- You can edit the binding keys for a dimension to change the order of the export columns or to specify the direction of each column in the export file. See Editing Binding Keys.
- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- You export a dimension to a CSV (comma-delimited) file. You can export to a local file, or if you set up a connection to an external application during registration you can export to the Planning application inbox.

> **Tip:**
>
> You can view and test application connection settings by inspecting the application. See Inspecting Applications.

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. Select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:
   - If you selected **Connection**, select a connection to export to, specify an export file name, and then click **Export**.
   - If you selected **File**, an export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example Planning_Account_20200123.csv, and can be edited before running the export. Your browser settings define the download location. Accept or edit the default filename, and then click **Export**.
5. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*), a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

After you export the dimension, you can import the data into your external application dimension, see Importing Dimension Metadata in *Administering Planning*.

Note:

The application export manages the node-level properties for shared nodes to comply with the requirements of Planning.

33 Working with Financial Consolidation and Close Applications

Financial Consolidation and Close provides an end to end solution for both effectively and efficiently managing the consolidation and close process. You use Oracle Fusion Cloud Enterprise Data Management to manage Financial Consolidation and Close dimensions.

Managing Financial Consolidation and Close data in Cloud EDM involves:

- Registering Financial Consolidation and Close Applications
- Importing Financial Consolidation and Close Dimensions
- Modifying Registered Financial Consolidation and Close Applications
- Exporting Financial Consolidation and Close Dimensions

When you create a Financial Consolidation and Close application, the following objects get created:

- An application is created with settings, such as alias tables, multi-currency codes, and optional connection settings.
- One default view is created for each application.
- One dimension is created for each registered dimension.
- For each dimension, these objects are created and bound to the dimension:
  - One node type with all of the application specific properties.
  - One hierarchy set using the node type for both parent and child.
  - One node set that points to the hierarchy set.
  - One viewpoint that points to the node set.

> **Note:**
>
> Binding rules ensure that the registered dimension conforms to the external application dimension requirements, see Understanding Binding Rules.

In addition to registering dimensions, you can also register attributes. See Understanding Attributes.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working, you can export the dimension back to the external application.

Financial Consolidation and Close Predefined Members

Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under them, but they cannot otherwise be edited in Cloud EDM.

Any changes to the predefined members must be performed in the external application. After the changes are made, you must perform a merge import (see Working with Merge Imports) to update the predefined members in Cloud EDM.

Registering Financial Consolidation and Close Applications

You register and modify Financial Consolidation and Close applications using a wizard that prompts you for information, such as the external application settings, connection URL, and dimensions.

Videos

Your Goal	Watch This Video
Learn about registering Financial Consolidation and Close applications.	▶ Registering Financial Consolidation and Close Applications

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

  > **Note:**
  >
  > The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which dimensions you want to register. You can register four dimension types: Account, Entity, Movement, and Custom. Each dimension type is included in specific cubes (see table below).
  When you register an application, properties for each dimension get created automatically. In addition to the standard properties, some additional properties unique to Financial Consolidation and Close get created depending on your application settings. See Predefined Properties for Financial Consolidation and Close Applications

Dimension Type	Included In These Cubes	Additional Properties Created
Account	– `Consolidation (Consol)` – `Rates` – Additional `Consol` cubes that you have enabled in the application settings (see Registering Financial Consolidation and Close Application Settings and Dimensions)	– `Is CICTA Account` – `CICTA Redirection Account` If intercompany is enabled: – `Is Plug Account` – `Plug Account` – `Intercompany Account`

Dimension Type	Included In These Cubes	Additional Properties Created
Entity	– Consol – Rates – Additional Consol cubes that you have enabled in the application settings (see Registering Financial Consolidation and Close Application Settings and Dimensions)	If intercompany is enabled: Intercompany Entity
Movement	– Consol – Additional Consol cubes that you have enabled in the application settings (see Registering Financial Consolidation and Close Application Settings and Dimensions)	
Custom	By default, the dimension is included in the Consol cube. You can clear that selection or select additional Consol cubes or the Rates cube in the application settings. See Registering Financial Consolidation and Close Application Settings and Dimensions	

- For new applications, the data storage properties for the Consol and Rates cubes (for example, `PLN.Data Storage (Consol)`) are automatically set to be derived from the application default data storage property (`PLN.Data Storage`). See Derived Properties.

> **Note:**
>
> Modifying an existing application's registration and editing an existing cube will not automatically set the data storage property for the cube to be derived from the application default. If you want the value for the cube data storage property to be derived from the application default, you must set up an application override. See Editing Property Parameters.

- The number of custom dimensions that you can have in your external Financial Consolidation and Close application depends on your application settings:
  - If you have not enabled Extended Dimensionality, the external application supports a maximum of two custom dimensions (one, if multi-GAAP is enabled).
  - If you have enabled Extended Dimensionality, the external application supports a maximum of four custom dimensions (three, if multi-GAAP is enabled).

> **Note:**
>
> Oracle Fusion Cloud Enterprise Data Management does not limit the number of custom dimensions that you can create for a Financial Consolidation and Close application.

For more information, see Understanding Registering Applications.

To register a Financial Consolidation and Close application:

1. From **Applications**, click **Register**.
2. Enter an application name and, optionally, a description.

> **Note:**
>
> The application type, name and description are displayed in the Application list.

3. If you are connecting to an external application, click **Add**, and define the connection settings. You can add multiple application connections.

Field	Valid Values	Example: connecting to an external application
Connection Name	Enter a required connection name, unique to this application.	US FCCS Cloud Connection
Connection Description	Enter an optional description.	Use this connection to connect to a Financial Consolidation and Close application.
Instance Location	Enter a connection URL to the external application in the format: `https://InstanceName-Cloud_Account_Name.EPM_Cloud_Family.Data_Center_Region.ocs.oraclecloud.com/Context`	See Sample EPM Cloud URLs in *Getting Started Guide for Administrators*
Primary Connection	Select this check box to make this the default for importing from and exporting to the external application.	

Field	Valid Values	Example: connecting to an external application
Authentication Type	Select an option: • Basic: Use the provided user credentials to authenticate. • OAuth: Use OAuth2 IDs and tokens to authenticate. (Available for environments on Oracle Cloud Infrastructure (OCI) / Gen 2 architecture only.) See Using OAuth2 in Cloud EPM Applications.	`Basic`
Basic:		
Identity Domain	Enter the identity domain of the external application (not required for OCI (Gen 2) URLs). **Note:** This is the same as the Cloud Account Name used in the **Instance Location**.	`epmid`
Username	Enter an authorized user name **Note:** The user must be a Service Administrator of the external application, and they cannot be authenticated using corporate SSO (identity provider) credentials.	`tom.smith`
Password	Enter a password.	xxxxxxxx
OAuth:		
Client ID	Client ID for the client that you created for application connections.	
Access Token	Access token for the client that you created for application connections.	
Refresh Token	Refresh token for the client that you created for application connections.	

4. Review the application summary information, and then click **Create**.

> **Note:**
>
> The application and default view are created and set to draft mode. The default view name is the same as the application name you specified above.

5. Continue the registration, see Registering Financial Consolidation and Close Application Settings and Dimensions.

Registering Financial Consolidation and Close Application Settings and Dimensions

Use these steps to register application settings and dimensions for Financial Consolidation and Close applications.

To define application settings and register dimensions:

1. Define the Financial Consolidation and Close application settings:

Field	Valid Values	Example
Multi-Currency	Set this check box if the application supports multi-currency.	Select the check box
Currencies	You must enter at least one currency code. The first currency that you enter will be set as the main currency by default.	`USD,CAD,INR`
Main Currency	Select one of the currencies that you entered in the currencies field. The default value is the first currency that you entered.	`USD`
Intercompany Enabled	Set this check box if the application supports intercompany transactions between entities	Select the check box
Multi-GAAP Enabled	Set this check box if the application requires reporting using multiple GAAP standards	Select the check box

Field	Valid Values	Example
Hybrid Mode	Set this check box if the application supports hybrid mode. **Note:** Select this check box to add the plan type-specific `Solve Order` properties to all BSO and ASO cubes. Clear the check box to add the plan-type specific `Solve Order` properties only to ASO cubes.	Select the check box
Extended Dimensionality Application	Set this check box if this is an extended dimensionality application. Extended dimensionality applications can have up to four custom dimensions in the external application instead of two.	Select the check box
Alias Tables	Enter dimension alias tables.	Default English Spanish

Field	Valid Values	Example
Additional Smart Lists	Enter the name of one or more smart lists. The smart list names you enter will be a selectable item in the `Smart List` property. Smart list values can be managed in the external application only. **Note:** The smart list property will always have `Calculation Status` in the list of allowed values for Financial Consolidation and Close. Enter any additional smart lists in this field.	Channel
Additional Cubes Enabled	Select the additional Consol cubes that you want to enable. You can enable Consol2 through Consol5 in any order.	Consol3, Consol5

2. Register the dimensions.

   To register one or more dimensions, click **Add**, and then enter the following information:

Field	Valid Values	Example
Dimension Name	Enter a dimension name. Note: **Dimension Name** is a display name in Oracle Fusion Cloud Enterprise Data Management only. The **External Dimension Name** gets exported to Financial Consolidation and Close.	Financial Account The value you enter becomes the name of the viewpoint, node set, hierarchy set, node type and dimension in Cloud EDM.
Dimension Description	Optionally enter a dimension description.	
Dimension Type	Select the dimension type: Account, Entity, Movement, or Custom.	Account
Allow Shared Nodes	Select if the dimension you are importing has shared nodes.	Select the check box
Select the valid cubes for this dimension (Custom dimension type only) Note: For each cube that you select, a set of cube-specific properties is created (for example, Plan Type).	For Custom dimension types, select the cubes for the dimension. By default, the Consol cube is selected. You can clear that selection to create the dimension without cubes.	Consol

Field	Valid Values	Example
External Dimension Name (Custom dimension type only)	If the dimension type is Custom, enter the external dimension name. **Note:** For Account, Entity, and Movement dimensions, the external name is always "Account", "Entity", or "Movement", and it cannot be changed. For Custom dimensions, you must enter an external name.	Product

3. Click **Next**.
4. If the dimension type is Entity, Movement, or Custom, optionally add one or more attributes:
   - For simple attributes:
     – Select **Simple Attribute**.
     – Enter an attribute name.
     – Enter all values you want to use for the attributes.

     > **Note:**
     >
     > The attribute name is put into the `Attribute - name` property and the values are put into a pull-down list.

   - For attribute dimensions:
     – Select **Attribute Dimension**.
     – Enter an attribute dimension name.

     > **Note:**
     >
     > The attribute dimension name is put into the `Attribute Dim - name` node data type property and the viewpoint for the attribute dimension is added to the default view for the application.

     – For **Attribute Type**, select Text, Numeric, or Date.

   See Understanding Attributes.

- For registered dimensions:
  - Select **Registered Dimension**.
  - Enter an attribute name.

    > **Note:**
    >
    > A node data type property is created with the name that you enter in order to reference the registered dimension.

  - Select the dimension that contains the attribute values that you want to reference from the **Dimension** drop down menu.

> **Note:**
>
> For simple attributes,attribute dimensions, and registered dimensions, the attribute name cannot be the same as any of the predefined account names (Account, Entity, Movement) or any of the custom dimension names that you entered. Additionally, the name cannot be one of the predefined account attribute dimension names:
>
> - Is CICTA Account
> - CICTA Redirection Account
> - Intercompany Account
> - Is Plug Account
> - Plug Account
> - Intercompany Entity

5. Review the summary, and then click **Apply**.
The dimension and its data chain are created. The data objects are bound to the dimension and set to active status. You can now import dimension data, see Importing Financial Consolidation and Close Dimensions.

Predefined Properties for Financial Consolidation and Close Applications

The following properties are predefined for all Financial Consolidation and Close applications.

Property	Data Type	Inherited	Level	Description
Access Granted to Member Creator	String	No	Node	Determines the access that member creators have to dynamic members that they create
Account Type	String	Yes	Node	Defines account time balance (how values flow over time) and determines accounts sign behavior for variance reporting with member formulas
Aggregation	String	Yes	Relationship	Determines how child member values aggregate to parent members

Property	Data Type	Inherited	Level	Description
Alias	String	No	Node	Alternate unique description for the dimension member
Allow Upper Level Entity Input	Boolean	No	Node	Specify whether parent entity input is allowed for this member.
Base Currency	String	Yes	Node	Specifies the base currency for an entity dimension member in a Standard multiple currency application **Note**: Base Currency is added to the Entity dimension only if Multi Currency is enabled during application registration.
Data Storage	String	No **Note**: For the Consol and Rates cubes in new applications, the data storage property for the cube (for example, `PLN.Data Storage (Consol)`) is derived from the application default data storage property (`PLN.Data Storage`)	Relationship	Specifies whether data is stored or calculated for a dimension member
Data Type	String	Yes	Node	Specifies the type of data values for a dimension member
Enable for Dynamic Children	Boolean	No	Node	Enables users to create children for this member by entering a member name in the runtime prompt for a business rule that has been configured with a dynamic parent member

Property	Data Type	Inherited	Level	Description
Enterprise Predefined Member	Boolean	No	Node	Designates a dimension member as predefined by the application. Predefined members are restricted to only specific types of changes such as moves, reorders, or adding a child under them. **Note**: This property is set by the import process when importing predefined members from a Financial Consolidation and Close application and it cannot be manually edited in Oracle Fusion Cloud Enterprise Data Management.
Exchange Rate Type	String	Yes	Node	Determines how exchanges rates are used to calculate values
Formula	Memo	No	Node	Member formula used to calculate a value for a dimension member.
Formula Description	String	No	Node	Description for a member formula
Hierarchy Type	String	No	Node	Available for dimension bound to an aggregate storage cube. Aggregate storage dimension are automatically enable to support multiple hierarchies. The first hierarchy in a multiple hierarchy dimension must be stored.
Number of Possible Dynamic Children	Integer	No	Node	Maximum number of dynamic children that can be created for a dimension member. This option is available if Enable for Dynamic Children is selected.
Plan Type	Boolean	Yes	Relationship	Specifies whether the dimension members if valid for the plan type.
Process Management Enabled	Boolean		Node	Determines whether process management is enabled
Skip Value	String	Yes	Node	Determines how database calculations treat zeros and missing values when Time Balance is First, Balance, or Average.
Smart List	String	No	Node	Select a smart list to associate to a dimension member.
Source Plan Type	String	Yes	Node	Source plan type for a dimension member
Time Balance	String	Yes	Node	Specifies how the application calculates the value of summary time periods

Property	Data Type	Inherited	Level	Description
Two Pass Calculation	Boolean	No	Node	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Data Storage of Dynamic Calc or Dynamic Calc and Store.
UDA	List	No	Node	User defined attribute used for calculation or reporting purposes
Variance Reporting	String	No	Node	Determines whether an account is treated as an expense during variance reporting
Weekly Distribution	Boolean	No	Node	Sets the weekly distribution. Available for bottom level account members if the option was selected when creating the application and the base time period is 12 months.

Depending on your dimensions and application settings, the following properties may also be predefined for Financial Consolidation and Close applications. These properties do not get inherited.

Property	Created If	Data Type	Level	Description
Is CICTA Account	Account type dimension is defined	String	Node	Specifies if an account is a Comprehensive Income Cumulative Translation Adjustment (CICTA) account (`CICTA_Acc_Yes`) or not (`Blank`).
CICTA Redirection Account	Account type dimension is defined	Node	Node	Specifies the CICTA redirection account. This property is restricted to nodes where the **Is CICTA Account** property is set to `CICTA_Acc_Yes`.
Intercompany Account	Account type dimension is defined and Intercompany is enabled	String	Node	Specifies if an account is used for intercompany eliminations.
Is Plug Account	Account type dimension is defined and Intercompany is enabled	String	Node	Specifies if an account is a plug account (`Plug_Acc_Yes`) or not (`Blank`).

Property	Created If	Data Type	Level	Description
Plug Account	Account type dimension is defined and Intercompany is enabled	Node	Node	Specifies the intercompany plug account. This property is restricted to nodes where the **Is Plug Account** property is set to `Plug_ACC_YES`.
Intercompany Entity	Entity type dimension is defined and Intercompany is enabled	String	Node	Specifies if an entity is used for intercompany eliminations.
Default Movement	Account type dimension is defined	String	Node **Note:** This property can be set on bottom nodes only.	Specifies members in the Movement dimension to calculate movements for a member of the Account dimension. **Note:** The Default Movement property is set up to use an Allowed Values list, with no values identified initially. Populate the allowed values list with the nodes from the Movement dimension where you specified that **Is Default Movement** is set to `YES`.

Property	Created If	Data Type	Level	Description
Is Default Movement	Movement type dimension is defined	String	Node **Note:** This property can be set on bottom nodes that are under the `FCCS_Mvmts_Subtotal` ancestor only.	Designates a Movement dimension member for selection as a Default Movement for an Account dimension member. **Note:** Set this field to `YES` for nodes that will be used in the Account dimension **Default Movement** property. After you set this field to Yes, you must add the node to the allowed values for the **Default Movement** property in the Account dimension.

Importing Financial Consolidation and Close Dimensions

For Financial Consolidation and Close dimensions, Oracle Fusion Cloud Enterprise Data Management supports importing from comma-delimited (CSV) files and from an external application outbox using a `.zip` file.

On import, the property `Enterprise Predefined Member` is set to `True` for predefined members and cannot be edited, with the exception of adding a child under it. See Seeded Dimension Members in *Administering Financial Consolidation and Close*. If the predefined members in the external application are changed, you must re-import the dimensions to update the predefined members in Cloud EDM.

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- In the external application, create a data file to import using one of these formats:
  - If you are using a local file, a `.csv` (comma-delimited) file for each dimension. The local `.csv` filename does not have any file naming restrictions.
  - If you are connecting to the external application outbox, a `.zip` file that contains `.csv` files for one or more dimensions. The `.csv` filename must contain the dimension name or external dimension name, if specified in the registration, at the end with preceding

text. For example, `acc1.csv` is invalid, but `tom_smith_acc1.csv` is valid, where `acc1` is the dimension name.

> **Note:**
>
> From Financial Consolidation and Close, create the files to import into Cloud EDM. See Exporting Metadata in *Administering Financial Consolidation and Close*.

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. Click the dimension that you want to import data into.
3. In **Input Source**, select **File** or **Connection**, and then do one of the following:
   - If you selected **File**, specify an import file name.
   - If you selected **Connection**, select a connection to import from, and specify an import file name.
4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.
5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*), a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

Financial Consolidation and Close Import and Export File Format

When you import and export data for Financial Consolidation and Close dimensions, you must use a comma-delimited (`.csv`) file.

The following list describes the comma-delimited file's format and other considerations:

- The file must have the following columns:
  - **Name** The dimension name specified in the registration or the external dimension name, if you entered it.
  - **Parent** The name of the parent node. If there is no parent node, this field must be empty.
- **Alias:** *name*, where *name* is the name of a the alias table you registered. You must have at least one alias table.
- These are common columns:
  - **Plan Type** (*cube name*), **Aggregation** (*cube name*), **Data Storage** (*cube name*), **Formula** (*cube name*), **Formula Description** (*cube name*), these are typically defined for each cube you registered.
  - **Solve Order** *(cube name)* this is defined for each ASO cube you registered.

> **Tip:**
>
> *cube name* is the name of the cube you registered.

  - **Smart List** The smart list names are optionally specified for nodes.
  - **Attribute Dimension Name** The name of the attribute dimension entered during registration.

> **Note:**
>
> Additional columns can be imported and exported from the external application, see Editing Dimension Properties in the Simplified Dimension Editor and Editing Member Properties in the Simplified Dimension Editor.

- The import file's data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a child node's row precedes its parent node's row, an error occurs when the file is imported.
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)
- The import ignores duplicate rows.

> **Note:**
>
> The import contains the logic to process shared nodes (Data Storage = shared) based on the requirements of Financial Consolidation and Close.

Modifying Registered Financial Consolidation and Close Applications

After you register a Financial Consolidation and Close application, you can modify it. For example, you can register additional dimensions, remove dimensions, or change the multi-currency setting. The application, dimension, and data objects are updated based on your changes. The dimension binding rules are updated to reflect the modifications you make.

> **Note:**
>
> If you remove an attribute dimension or alias, its corresponding property is unbound from the dimension and is no longer imported or exported. For example, if you remove the Default alias dimension, the corresponding property, `Alias: Default`, is no longer imported or exported for the node type. The property remains on the node type until you remove it.
>
> If you remove a dimension, the corresponding data chain objects are unbound from that dimension but they are not deleted. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for a Financial Consolidation and Close application:

1. From **Applications**, find your application, click ⋮, and then select **Modify Registration**.
2. To change or add application settings or dimensions, see Registering Financial Consolidation and Close Application Settings and Dimensions.
3. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

> **Note:**
>
> If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
>
> If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Binding Rules for Financial Consolidation and Close Applications

Binding rules are essential because they automatically make the Oracle Fusion Cloud Enterprise Data Management application conform to the requirements of the external application.

The following table describes the binding rules for Financial Consolidation and Close.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.
Hierarchy Set	For bound hierarchy sets, you cannot select a node type that is missing required dimension properties unless you confirm that the system may add the properties when you save.
Node Set	For bound node sets, you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. The hierarchy set is checked and then the node types are checked.

Object	Rule
Dimension	When you edit a viewpoint bound to a dimension: • You can select a different viewpoint for the binding to a dimension. The new viewpoint is checked for binding rules as are all data objects up the data chain. **Note:** You can also remove a dimension by deleting it from the application registration. The dimension still displays in the Application Inspector, and data chain objects may be created using the dimension, but it cannot be used for importing or exporting data. • In the application import and export sections, you must select a dimension that passes its own binding rules or any binding rules up the data chain.

For more information, see:

- Understanding Binding Rules
- Binding Rules for All Application Types

Binding a Viewpoint to a Dimension in Financial Consolidation and Close Applications

Financial Consolidation and Close applications support binding a viewpoint to a single external dimension. If you create an alternate viewpoint that you want to bind to the external dimension instead, you can edit the binding to point to the alternate viewpoint by taking the following steps.

For more information, see:

- Understanding Bindings and Bound Data Objects
- Understanding Viewpoints

1. From **Applications**, find the application containing the viewpoint.
2. In the application's **Actions** column, click ⋮, then select **Inspect**
3. Select Dimensions, and then click the name of the dimension to which you are binding the viewpoint.
   The dimension inspector is displayed.
4. Select Bindings, and click the binding to associate with the alternate viewpoint.
5. Click **Edit**.
   The **View** and **Viewpoint** drop-down lists become active.
6. From **View**, select the view containing the viewpoint you want to bind to the dimension.
7. Select the viewpoint.
8. Click **Save**.

Financial Consolidation and Close Validations

Validations ensure data integrity. The following validations run for Financial Consolidation and Close applications.

In addition to these Financial Consolidation and Close, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

> **Note:**
>
> For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

Name	Source Type	Description	Scope
Alias Uniqueness	Planning	Alias are not unique among a group of sibling nodes	Siblings
Cube Data Storage	Planning	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only	Node
Data Storage	Planning	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only **Note**: This is not checked if the member is not in the cube.	Viewpoint
Date Attribute Name	Planning	Bottom level node names in a date attribute dimension are not in `mm-dd-yyyy` format	Node
Default Movement Level	Financial Consolidation and Close	The Default Movement property is defined on a node other than a bottom level node	Node
Exchange Rate Type	Planning	Exchange Rate Type is not None and Data Type is not Currency	Node
Formula Allowed	Planning	Data Storage is not Dynamic Calc or Dynamic Calc and Store	Node
Is Default Movement Ancestor	Financial Consolidation and Close	The Is Default Movement property is defined on a node that is not a descendant of `FCCS_Mvmts_Subtotal`	Viewpoint
Is Default Movement Level	Financial Consolidation and Close	The Is Default Movement property is defined on a node other than a bottom level node	Node
Member Name Conflicts	Planning	Member name is the same as a dimension name, a cube, or an attribute dimension defined for the current application	Node and data chain
Name Uniqueness	Planning	Member name is not unique across all nodes used in an application	Application-bound node types

Name	Source Type	Description	Scope
Numeric Attribute Name	Planning	Bottom level node names in a numeric attribute dimension are not in numeric format	Node
Plan Types	Planning	Plan Type is set to True but the parent does not have a True value **Note**: This validation is skipped if the parent is a predefined member.	Viewpoint
Predefined Member	Planning Modules	If the Enterprise Predefined Member property equals `True` for a node, it can only be moved or reordered. You cannot edit the node or modify the Enterprise Predefined Member property on the node.	Node
Shared Member before Base Member	Planning	Shared members appear above base members in a hierarchy	Viewpoint
Shared Members	Planning	Shared members are not in the same dimension	Application-bound node types
Smart List Required	Planning	Smart List is missing for member if Data Type is Smart List	Node
Source Plan Type	Planning	Source Plan Type is not a valid Plan Type for the member	Node
Variance Reporting	Planning	Variance Reporting is not Expense when Account Type is Expense or Variance Reporting is Non-Expense if Account Type is anything other than Expense or Saved Assumption.	Node

Note:

Most predefined validations are not run against predefined members (that is, nodes with `PLN.Enterprise Predefined Member` set to `True`), because these members are imported from the external application and are not able to be edited in Oracle Fusion Cloud Enterprise Data Management.

The only predefined validations that are run against predefined members are the Predefined Member and Alias Uniqueness validations.

Invalid Characters, First Characters, and Values for Financial Consolidation and Close

When you register a Financial Consolidation and Close application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` and `Alias` properties. For a list of these characters and values, see Naming Restrictions for Dimensions, Members, and Aliases in *Administering Financial Consolidation and Close* .

You can edit these values by editing the property parameters. See Editing Property Parameters

Exporting Financial Consolidation and Close Dimensions

You can export from a viewpoint that is bound to a dimension back to a Financial Consolidation and Close application.

Best Practice

It's a best practice to validate the viewpoint bound to the dimension prior to export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States, and all states. If the node set's top node is Texas, only Texas and the nodes under it are exported.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension that you are exporting from.
- When you register a Financial Consolidation and Close application, nodes and hierarchical relationships are imported and displayed in the order they were in the application. Nodes and hierarchical relationships are exported in the order they are in Oracle Fusion Cloud Enterprise Data Management. If you want nodes and hierarchical relationships to be exported in alpha numeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.
- You can edit the binding keys for a dimension to change the order of the export columns or to specify the direction of each column in the export file. See Editing Binding Keys.
- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- You export a dimension to a CSV (comma-delimited) file. You can export to a local file, or if you set up a connection to an external application during registration you can export to the Financial Consolidation and Close application inbox.

> **Tip:**
>
> You can view and test application connection settings by inspecting the application. See Inspecting Applications.

After you export, you can import the data into your external application dimension, see Importing Metadata in *Administering Financial Consolidation and Close*.

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. Select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:

- If you selected **Connection**, select a connection to export to, specify an export file name, and then click **Export**.
- If you selected **File**, an export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example FCCS_Account_20200123.csv, and can be edited before running the export. Your browser settings define the download location. Accept or edit the default file name, and then click **Export**.

5. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*), a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

34 Working with Enterprise Profitability and Cost Management Applications

You use Enterprise Profitability and Cost Management to manage the cost and revenue allocations that are necessary to compute profitability for a business segment, such as a product, customer, region, or branch.

> **Note:**
>
> The Enterprise Profitability and Cost Management application adapter should be used for Enterprise Profitability and Cost Management external applications only. If you are integrating with a Profitability and Cost Management external application you must use the Universal application adapter. See Working with Universal Applications.

Managing data for Enterprise Profitability and Cost Management involves:

- Registering Enterprise Profitability and Cost Management Applications
- Importing Enterprise Profitability and Cost Management Dimensions
- Modifying Registered Enterprise Profitability and Cost Management Applications
- Exporting Enterprise Profitability and Cost Management Dimensions

When you create an Enterprise Profitability and Cost Management application, the following objects get created:

- An application is created with settings, such as alias tables, multi-currency codes, and optional connection settings.
- One default view is created for each application.
- One dimension is created for each registered dimension.
- For each dimension, these objects are created and bound to the dimension:
  - One node type with all of the application specific properties.
  - One hierarchy set using the node type for both parent and child.
  - One node set that points to the hierarchy set.
  - One viewpoint that points to the node set.

  > **Note:**
  >
  > Binding rules ensure that the registered dimension conforms to the external application dimension requirements, see Understanding Binding Rules.

In addition to registering dimensions, you can also register attributes. See Understanding Attributes.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working, you can export the dimension back to the external application.

Enterprise Profitability and Cost Management Predefined Members

Predefined members must be imported from the external application, and they cannot be edited in Oracle Fusion Cloud Enterprise Data Management. Any changes to the predefined members must be performed in the external application. After the changes are made, you must perform a merge import (see Working with Merge Imports) to update the predefined members in Cloud EDM.

Registering Enterprise Profitability and Cost Management Applications

When you register an Enterprise Profitability and Cost Management external application into Oracle Fusion Cloud Enterprise Data Management, an application and default view are created which represent the external application.

Videos

Your Goal	Watch This Video
Learn about registering Enterprise Profitability and Cost Management applications.	▶ Registering Enterprise Profitability and Cost Management Applications

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

> **Note:**
>
> The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which dimensions you want to register. You must register all the cubes associated with each dimension. You can register two dimension types: Entity and Custom.
- When you add a cube to a new or existing application, the data storage property for the cube (for example, `PLN.Data Storage (PCM_CLC)`) is automatically set to be derived from the application default data storage property (`PLN.Data Storage`). See Derived Properties.

> **Note:**
>
> Editing an existing cube by modifying an application's registration will not set the data storage property for the cube to be derived from the application default. If you want the value for the cube data storage property to be derived from the application default, you must set up an application override. See Editing Property Parameters.

For more information, see Understanding Registering Applications.

To register an Enterprise Profitability and Cost Management application:

1. From **Applications**, click **Register**, and then select **Enterprise Profitability and Cost Management**.
2. Enter an application name and description.

   > **Note:**
   >
   > The application type, name and description are displayed in the Application list.

3. If you are using a local csv (comma-delimited) file for your dimension, skip the next step.
4. If you are connecting to an external application, click **Add**, and define the connection settings. You can add multiple application connections.

Field	**Valid Values**	**Example: connecting to an external application**
Connection Name	Enter a required connection name, unique to this application.	US Enterprise Profitability Cloud Connection
Connection Description	Enter an optional description.	Use this connection to connect to an Enterprise Profitability and Cost Management application.
Instance Location	Enter a connection URL to the external application in the format: `https://`*`InstanceName-Cloud_Account_Name.EPM_Cloud_Family.Data_Center_Region`*`.ocs.oraclecloud.com/`*`Context`*	See Sample EPM Cloud URLs in *Getting Started Guide for Administrators*
Primary Connection	Select this check box to make this the default for importing from and exporting to the external application.	

Field	Valid Values	Example: connecting to an external application
Authentication Type	Select an option: • Basic: Use the provided user credentials to authenticate. • OAuth: Use OAuth2 IDs and tokens to authenticate. (Available for environments on Oracle Cloud Infrastructure (OCI) / Gen 2 architecture only.) See Using OAuth2 in Cloud EPM Applications.	`Basic`
Basic:		
Identity Domain	Enter the identity domain of the external application (not required for OCI (Gen 2) URLs). **Note:** This is the same as the Cloud Account Name used in the **Instance Location**.	`epmid`
Username	Enter an authorized user name **Note:** The user must be a Service Administrator of the external application, and they cannot be authenticated using corporate SSO (identity provider) credentials.	`tom.smith`
Password	Enter a password.	xxxxxxxx
OAuth:		
Client ID	Client ID for the client that you created for application connections.	
Access Token	Access token for the client that you created for application connections.	
Refresh Token	Refresh token for the client that you created for application connections.	

5. Review the application summary information, and then click **Create**.

> **Note:**
>
> The application and default view are created and set to draft mode. The default view name is the same as the application name you specified above.

6. Continue the registration, see Registering Enterprise Profitability and Cost Management Cubes, Application Settings, and Dimensions.

Registering Enterprise Profitability and Cost Management Cubes, Application Settings, and Dimensions

Use these steps to register cubes, application settings, and dimensions for Enterprise Profitability and Cost Management applications.

Predefined Cubes and Dimensions

When you register an Enterprise Profitability and Cost Management application, the following cubes and dimensions are created automatically:

Cubes:

- `PCM_CLC`: In Enterprise Profitability and Cost Management, use this cube to run calculations.
- `PCM_REP`: In Enterprise Profitability and Cost Management, use this cube to view data and run reports.

Dimensions:

Dimension Name	Dimension Type	Valid Cubes
Account	Custom	`PCM_CLC`, `PCM_REP`
Entity	Entity	`PCM_CLC`, `PCM_REP`

You can delete these predefined cubes and dimensions, and you can define additional cubes and dimensions as needed.

To define cubes, application settings and register dimensions:

1. On the **Cubes** screen, click **Add**, and then enter the cube name.

   > **Note:**
   >
   > The cube type is automatically set to ASO and cannot be changed.

   Add all the cubes used in the dimensions you are registering.

2. Define the Enterprise Profitability and Cost Management application settings:

Field	Valid Values	Example
Alias Tables	Enter dimension alias tables.	Default English Spanish

Field	Valid Values	Example
Multi-Currency	Set this check box if the application supports multi-currency.	Select the check box
Currencies	You must enter at least one currency code.	`USD,CAD,INR`

3. Register the dimensions.

   To register one or more dimensions, click **Add**, and then enter the following information:

Field	Valid Values	Example
Dimension Name	Enter a dimension name. **Note:** **Dimension Name** is a display name in Oracle Fusion Cloud Enterprise Data Management only. The **External Dimension Name** gets exported to Enterprise Profitability and Cost Management.	Product The value you enter becomes the name of the viewpoint, node set, hierarchy set, node type and dimension in Cloud EDM. **Note:** You cannot use PCM_Balance or PCM_Rule as a dimension name.
Dimension Description	Optionally enter a dimension description.	
Dimension Type	Select the dimension type: Custom (default) or Entity. **Tip:** Enterprise Profitability and Cost Management uses the Custom dimension type for the Account dimension.	Custom
Allow Shared Nodes	Select if the dimension you are importing has shared nodes.	Select the check box

Field	Valid Values	Example
Select the valid cubes for this dimension **Note:** For each cube that you select, a set of cube-specific properties is created (for example, Plan Type).	Select all the cubes for the dimension. Use the back button to enter additional cubes if needed. **Note:** You must select at least one valid cube for the Entity dimension type. Selecting cubes for Custom dimensions is optional.	PCM_CLC, PCM_REP
External Dimension Name	Enter the external dimension name.	Product

4. Click **Next**.
5. Optionally, add one or more attributes:
   - For simple attributes:
     – Select **Simple Attribute**.
     – Enter an attribute name.
     – Enter all values you want to use for the attributes.

     > **Note:**
     >
     > The attribute name is put into the `Attribute - name` property and the values are put into a pull-down list.

   - For attribute dimensions:
     – Select **Attribute Dimension**.
     – Enter an attribute dimension name.

     > **Note:**
     >
     > The attribute dimension name is put into the `Attribute Dim - name` node data type property and the viewpoint for the attribute dimension is added to the default view for the application.

     – For **Attribute Type**, select Text, Numeric, or Date.

     See Understanding Attributes.
   - For registered dimensions:
     – Select **Registered Dimension**.

- Enter an attribute name.

> **Note:**
>
> A node data type property is created with the name that you enter in order to reference the registered dimension.

- Select the dimension that contains the attribute values that you want to reference from the **Dimension** drop down menu.

6. Review the summary, and then click **Apply**.
The dimension and its data chain are created. The data objects are bound to the dimension and set to active status. You can now import dimension data, see Importing Enterprise Profitability and Cost Management Dimensions.

Predefined Properties for Enterprise Profitability and Cost Management Applications

The following properties are predefined for Enterprise Profitability and Cost Management applications. All Enterprise Profitability and Cost Management properties use the PLN namespace.

For more information, see:

- Working with Properties
- Inheriting Properties

Property	Data Type	Inherited	Level	Description
Access Granted to Member Creator	String	No	Node	Determines the access that member creators have to dynamic members that they create
Aggregation	String	Yes	Relationship	Determines how child member values aggregate to parent members
Alias	String	No	Node	Alternate unique description for the dimension member
Attribute Dimension Name	String	No	Node	Specifies the name of the attribute dimension
Base Currency	String	Yes	Node	Specifies the base currency for an Entity dimension member in a standard multiple currency application. **Note**: This property is displayed only in the Entity dimension in applications where multi-currency is enabled.

Property	Data Type	Inherited	Level	Description
Data Storage	String	No **Note:** When you add a cube, the data storage property for the cube (for example, `PLN.Data Storage (PCM_CLC)`) is derived from the application default data storage property (`PLN.Data Storage`)	Relationship	Specifies whether data is stored or calculated for a dimension member
Data Type	String	Yes	Node	Specifies the type of data values for a dimension member
Enable for Dynamic Children	Boolean	No	Node	Enables users to create children for this member by entering a member name in the runtime prompt for a business rule that has been configured with a dynamic parent member
Enterprise Predefined Member	Boolean	No	Node	Designates a dimension member as predefined by the application. Predefined members are restricted to only specific types of changes such as moves, reorders, or adding a child under them. **Note**: This property is set by the import process when importing predefined members from a Enterprise Profitability and Cost Management application and it cannot be manually edited in Oracle Fusion Cloud Enterprise Data Management.
Formula	Memo	No	Node	Member formula used to calculate a value for a dimension member.
Formula Description	String	No	Node	Description for a member formula
Hierarchy Type	String	No	Node	Available for dimension bound to an aggregate storage cube. Aggregate storage dimension are automatically enable to support multiple hierarchies. The first hierarchy in a multiple hierarchy dimension must be stored.

Property	Data Type	Inherited	Level	Description
Number of Possible Dynamic Children	Integer	No	Node	Maximum number of dynamic children that can be created for a dimension member. This option is available if Enable for Dynamic Children is selected.
Plan Type	Boolean	Yes	Relationship	Specifies whether the dimension members if valid for the plan type.
Process Management Enabled	Boolean		Node	Determines whether process management is enabled
Smart List	String	No	Node	Select a smart list to associate to a dimension member.
Solve Order	Integer	No	Node	Specifies the order in which formulas are evaluated.
Two Pass Calculation	Boolean	No	Node	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Data Storage of Dynamic Calc or Dynamic Calc and Store.
UDA	List	No	Node	User defined attribute used for calculation or reporting purposes

Importing Enterprise Profitability and Cost Management Dimensions

For Enterprise Profitability and Cost Management dimensions, Oracle Fusion Cloud Enterprise Data Management supports importing from comma-delimited (CSV) files and from an external application outbox using a `.zip` file.

On import, the property `Enterprise Predefined Member` is set to `True` for predefined members and cannot be edited, with the exception of adding a child under it.

Note:

Predefined Members can be updated in the external Enterprise Profitability and Cost Management application only. If the predefined members in the external application are changed, you must re-import the dimensions to update the predefined members in Cloud EDM.

Videos

Your Goal	Watch This Video
Learn about importing and exporting dimensions.	▶ Importing and Exporting Application Data

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- In the external application, create a data file to import using one of these formats:
  - If you are using a local file, a `.csv` (comma-delimited) file for each dimension. The local `.csv` filename does not have any file naming restrictions.
  - If you are connecting to the external application outbox, a `.zip` file that contains `.csv` files for one or more dimensions. The `.csv` filename must contain the dimension name or external dimension name, if specified in the registration, at the end with preceding text. For example, `acc1.csv` is invalid, but `tom_smith_acc1.csv` is valid, where `acc1` is the dimension name.

> **Note:**
>
> From Enterprise Profitability and Cost Management, create the files to import into Cloud EDM. See Exporting Metadata in *Administering and Working with Enterprise Profitability and Cost Management*.

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. Click the dimension that you want to import data into.
3. In **Input Source**, select **File** or **Connection**, and then do one of the following:
   - If you selected **File**, specify an import file name.
   - If you selected **Connection**, select a connection to import from, and specify an import file name.
4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.
5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*) or a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

Enterprise Profitability and Cost Management Import and Export File Format

When you import and export data for Enterprise Profitability and Cost Management dimensions, you must use a comma-delimited (`.csv`) file.

The following list describes the comma-delimited file's format and other considerations:

- The file must have the following columns:
  - **Name** The dimension name specified in the registration or the external dimension name, if you entered it.

  - **Parent** The name of the parent node. If there is no parent node, this field must be empty.
- **Alias:** *name*, where *name* is the name of a the alias table you registered. You must have at least one alias table.
- These are common columns:
  - **Plan Type** (*cube name*), **Aggregation** (*cube name*), **Data Storage** (*cube name*), **Formula** (*cube name*), **Formula Description** (*cube name*), these are typically defined for each cube you registered.
  - **Solve Order** *(cube name)* this is defined for each ASO cube you registered.

    > **Tip:**
    >
    > *cube name* is the name of the cube you registered.

  - **Attribute Dimension Name** The name of the attribute dimension entered during registration.

    > **Note:**
    >
    > Additional columns can be imported and exported from the external application, see Dimension Properties and Member Properties.

- The import file's data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a child node's row precedes its parent node's row, an error occurs when the file is imported.
- The import ignores duplicate rows.
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)

> **Note:**
>
> The import contains the logic to process shared nodes (Data Storage = shared) based on the requirements of Enterprise Profitability and Cost Management.

Modifying Registered Enterprise Profitability and Cost Management Applications

After you register an Enterprise Profitability and Cost Management application, you can modify it. For example, you can register additional dimensions, remove dimensions, change the multi-currency setting, or add a cube. The application, dimension, and data objects are updated based on your changes. The dimension binding rules are updated to reflect the modifications you make.

> **Note:**
>
> If you remove a cube, attribute dimension, or alias, its corresponding property is unbound from the dimension and is no longer imported or exported. For example, if you remove the Default alias dimension, the corresponding property, `Alias: Default`, is no longer imported or exported for the node type. The property remains on the node type until you remove it.
>
> If you remove a dimension, the corresponding data chain objects are unbound from that dimension but they are not deleted. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for an Enterprise Profitability and Cost Management application:

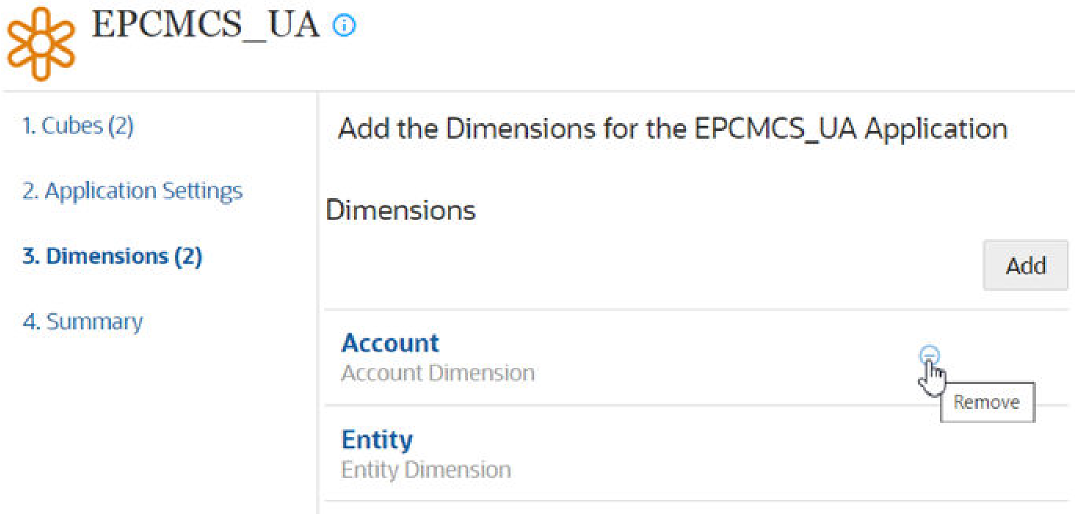
1. From **Applications**, find your application, click ⋮, and then select **Modify Registration**.
2. To change or add cubes, application settings, and dimensions, see Registering Enterprise Profitability and Cost Management Cubes, Application Settings, and Dimensions.
3. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

   > **Note:**
   >
   > If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
   >
   > If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Removing Dimensions

To remove a dimension from the application:

1. From **Applications**, find the application, click ⋮, and then select **Modify Registration**.
2. Select **Dimensions**, and then click ⊖ .



EPCMCS_UA

1. Cubes (2)

2. Application Settings

3. Dimensions (2)

4. Summary

Add the Dimensions for the EPCMCS_UA Application

Dimensions

Add

Account
Account Dimension

Remove

Entity
Entity Dimension

Binding Rules for Enterprise Profitability and Cost Management Applications

Binding rules are essential because they automatically make the Oracle Fusion Cloud Enterprise Data Management application conform to the requirements of the external application.

The following table describes the binding rules for Enterprise Profitability and Cost Management applications.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.
Hierarchy Set	For bound hierarchy sets, you cannot select a node type that is missing required dimension properties unless you confirm that the system may add the properties when you save.
Node Set	For bound node sets, you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. The hierarchy set is checked and then the node types are checked.

Object	Rule
Dimension	When you edit a viewpoint bound to a dimension: • You can select a different viewpoint for the binding to a dimension. The new viewpoint is checked for binding rules as are all data objects up the data chain. **Note:** You can also remove a dimension by deleting it from the application registration. The dimension still displays in the Application Inspector, and data chain objects may be created using the dimension, but it cannot be used for importing or exporting data. • In the application import and export sections, you must select a dimension that passes its own binding rules or any binding rules up the data chain.

For more information, see:

- Understanding Binding Rules
- Binding Rules for All Application Types

Enterprise Profitability and Cost Management Validations

Validations ensure data integrity. The following validations run for Enterprise Profitability and Cost Management applications.

In addition to these Enterprise Profitability and Cost Management, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

> **Note:**
>
> For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

Name	Source Type	Description	Scope
Alias Uniqueness	Planning	Alias are not unique among a group of sibling nodes	Siblings
Cube Data Storage	Planning	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only	Node
Data Storage	Planning	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only **Note**: This is not checked if the member is not in the cube.	Viewpoint

Name	Source Type	Description	Scope
Date Attribute Name	Planning	Bottom level node names in a date attribute dimension are not in mm-dd-yyyy format	Node
Formula Allowed	Planning	Data Storage is not Dynamic Calc or Dynamic Calc and Store	Node
Member Name Conflicts	Planning	Member name is the same as a dimension name, a cube, or an attribute dimension defined for the current application	Node and data chain
Name Uniqueness	Planning	Member name is not unique across all nodes used in an application	Application-bound node types
Numeric Attribute Name	Planning	Bottom level node names in a numeric attribute dimension are not in numeric format	Node
Plan Types	Planning	Plan Type is set to True but the parent does not have a True value **Note**: This validation is skipped if the parent is a predefined member.	Viewpoint
Predefined Member	Planning Modules	If the Enterprise Predefined Member property equals True for a node, it can only be moved or reordered. You cannot edit the node or modify the Enterprise Predefined Member property on the node.	Node
Shared Member before Base Member	Planning	Shared members appear above base members in a hierarchy	Viewpoint
Shared Members	Planning	Shared members are not in the same dimension	Application-bound node types
Smart List Required	Planning	Smart List is missing for member if Data Type is Smart List	Node

Note:

Most predefined validations are not run against predefined members (that is, nodes with PLN.Enterprise Predefined Member set to True), because these members are imported from the external application and are not able to be edited in Oracle Fusion Cloud Enterprise Data Management.

The only predefined validations that are run against predefined members are the Predefined Member and Alias Uniqueness validations.

> **Note:**
>
> The Formula Allowed and Plan Type validations are not checked on shared nodes (Data Storage = shared). The imports and exports at the application level transform these values as required by Enterprise Profitability and Cost Management.

Invalid Characters, First Characters, and Values for Enterprise Profitability and Cost Management

When you register a Enterprise Profitability and Cost Management application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` and `Alias` properties. For a list of these characters and values, see Naming Restrictions for Dimensions, Members, and Aliases in *Administering and Working with Enterprise Profitability and Cost Management* .

You can edit these values by editing the property parameters. See Editing Property Parameters

Exporting Enterprise Profitability and Cost Management Dimensions

You can export from a viewpoint that is bound to a dimension back to a Enterprise Profitability and Cost Management application.

Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States, and all states. If the node set's top node is Texas, only Texas and the nodes under it are exported.

Videos

Your Goal	Watch This Video
Learn about exporting dimensions.	▶ Importing and Exporting Application Data

Best Practice

It's a best practice to validate the viewpoint bound to the dimension prior to export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension that you are exporting from.
- When you register an Enterprise Profitability and Cost Management application, nodes and hierarchical relationships are imported and displayed in the order they were in the application. Nodes and hierarchical relationships are exported in the order they are in Oracle Fusion Cloud Enterprise Data Management. If you want nodes and hierarchical relationships to be exported in alpha numeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.

- You can edit the binding keys for a dimension to change the order of the export columns or to specify the direction of each column in the export file. See Editing Binding Keys.
- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- You export a dimension to a CSV (comma-delimited) file. You can export to a local file, or if you set up a connection to an external application during registration you can export to the Enterprise Profitability and Cost Management application inbox.

> **Tip:**
>
> You can view and test application connection settings by inspecting the application. See Inspecting Applications.

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. Select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:
   - If you selected **Connection**, select a connection to export to, specify an export file name, and then click **Export**.
   - If you selected **File**, an export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example Profitability_Account_20221223.csv, and can be edited before running the export. Your browser settings define the download location. Accept or edit the default filename, and then click **Export**.
5. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*) or a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

After you export the dimension, you can import the data into your external application dimension, see Importing Metadata in *Administering and Working with Enterprise Profitability and Cost Management*.

> **Note:**
>
> The application export manages the node-level properties for shared nodes to comply with the requirements of Enterprise Profitability and Cost Management.

35 Working with Tax Reporting Applications

Oracle EPM Tax Reporting provides comprehensive global tax reporting for medium to large multinational companies and includes Tax Provisioning, Country by Country Reporting, Workflow Management, Supplemental Data Management, and Dashboards and Key Performance Indicator reporting.

Managing Tax Reporting data in Oracle Fusion Cloud Enterprise Data Management involves:

- Registering Tax Reporting Applications
- Importing Tax Reporting Dimensions
- Modifying Registered Tax Reporting Applications
- Exporting Tax Reporting Dimensions

When you create a Tax Reporting application, the following objects get created:

- An application is created with settings, such as alias tables, multi-currency codes, and optional connection settings.
- One default view is created for each application.
- One dimension is created for each registered dimension.
- For each dimension, these objects are created and bound to the dimension:
  - One node type with all of the application specific properties.
  - One hierarchy set using the node type for both parent and child.
  - One node set that points to the hierarchy set.
  - One viewpoint that points to the node set.

> **Note:**
>
> Binding rules ensure that the registered dimension conforms to the external application dimension requirements, see Understanding Binding Rules.

In addition to registering dimensions, you can also register attributes. See Understanding Attributes. When you register an application, a number of attributes are predefined for you depending on your registration settings. See Predefined Attributes for Tax Reporting Applications. You can optionally add one or more additional attributes during registration.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working, you can export the dimension back to the external application.

Tax Reporting Predefined Members

Predefined members must be imported from the external application. They can be moved and reordered and you can add a child under or a top node above them, but they cannot otherwise be edited in Cloud EDM.

Any changes to the predefined members must be performed in the external application. After the changes are made, you must perform a merge import (see Working with Merge Imports) to update the predefined members in Cloud EDM.

Registering Tax Reporting Applications

You register and modify Tax Reporting applications using a wizard that prompts you for information, such as the external application settings, connection URL, and dimensions.

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

> **Note:**
>
> The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which dimensions you want to register. You can register six dimension types to be managed in Oracle Fusion Cloud Enterprise Data Management: Account, Custom, Entity, Jurisdiction, Movement, and From Currency. Each dimension type is valid for specific cubes.

> **Note:**
>
> For Custom dimensions, you select the valid cubes during registration. See Registering Tax Reporting Application Settings and Dimensions.

Dimension Type	Valid Cubes for the Dimension
Account	`Consolidation (Consol)`, `Rates`, `Country by Country Reporting (CbCR)`, and `Config`
Entity	`Consol`, `Rates`, `CbCR`, and `Config`
Jurisdiction	`Consol`, `CbCR`, and `Config`
Movement	`Consol`, `CbCR`, and `Config`
From Currency	`Rates`

> **Note:**
>
> The **Consol** cube is always created. The other cubes get created depending on the application settings that you select during registration. See Registering Tax Reporting Application Settings and Dimensions
> You cannot create additional cubes.

When you register an application, properties for each dimension get created automatically. In addition to the standard properties, some additional properties unique to Tax Reporting get created depending on your application settings. See Predefined Properties for Tax Reporting Applications.

- For new applications, the data storage properties for the cubes (for example, `PLN.Data Storage (Consol)`) are automatically set to be derived from the application default data storage property (`PLN.Data Storage`). See Derived Properties.

> **Note:**
>
> Modifying an existing application's registration and editing an existing cube will not automatically set the data storage property for the cube to be derived from the application default. If you want the value for the cube data storage property to be derived from the application default, you must set up an application override. See Editing Property Parameters.

To register a Tax Reporting application:

1. From **Applications**, click **Register**.
2. Select **Tax Reporting**.
3. Enter an application name and description.

> **Note:**
>
> The application type, name and description are displayed in the Application list.

4. If you are using a local csv (comma-delimited) file for your dimension, skip the next step.
5. If you are connecting to an external application, click **Add**, and define the connection settings. You can add multiple application connections.

Field	**Valid Values**	**Example: connecting to an external application**
Connection Name	Enter a required connection name, unique to this application.	TRCS Cloud Connection
Connection Description	Enter an optional description.	Use this connection to connect to a Tax Reporting application.
Instance Location	Enter a connection URL to the external application in the format: `https://`*`InstanceName-Cloud_Account_Name.EPM_Cloud_Family.Data_Center_Region.ocs.oraclecloud.com/Context`*	See Sample EPM Cloud URLs in *Getting Started Guide for Administrators*
Primary Connection	Select this check box to make this the default for importing from and exporting to the external application.	

Field	Valid Values	Example: connecting to an external application
Authentication Type	Select an option: • Basic: Use the provided user credentials to authenticate. • OAuth: Use OAuth2 IDs and tokens to authenticate. (Available for environments on Oracle Cloud Infrastructure (OCI) / Gen 2 architecture only.) See Using OAuth2 in Cloud EPM Applications.	`Basic`
Basic:		
Identity Domain	Enter the identity domain of the external application (not required for OCI (Gen 2) URLs). **Note:** This is the same as the Cloud Account Name used in the **Instance Location**.	`epmid`
Username	Enter an authorized user name **Note:** The user must be a Service Administrator of the external application, and they cannot be authenticated using corporate SSO (identity provider) credentials.	`tom.smith`
Password	Enter a password.	xxxxxxxx
OAuth:		
Client ID	Client ID for the client that you created for application connections.	
Access Token	Access token for the client that you created for application connections.	
Refresh Token	Refresh token for the client that you created for application connections.	

6. Review the application summary information, and then click **Create**.

> **Note:**
>
> The application and default view are created and set to draft mode. The default view name is the same as the application name you specified above.

7. Continue the registration, see Registering Tax Reporting Application Settings and Dimensions.

Registering Tax Reporting Application Settings and Dimensions

Use these steps to register application settings and dimensions for Tax Reporting applications.

To define application settings and register dimensions:

1. Define the Tax Reporting application settings:

Field	Description	Effects of Application Setting
Is this application Multi-Currency?	Select this checkbox if the application supports multi-currency. For more information about this setting, see Working with Multiple Currencies in *Administering Tax Reporting* .	Enables the `Rates` cube for Custom dimensions Creates these properties: - `Override FX Translation Method` - `FX Rates - Global Account Rate Override` - `NIBT FX Override Account`
Currencies	Enter the currencies that the application supports. You must enter at least one currency in order to proceed to the next registration step.	
Is CbCR Enabled?	Select this checkbox if the application supports Country by Country Reporting. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Enables the `CbCR` cube for Custom dimensions Creates the `Alias:CbCR Filing` alias.
Is Current Tax Payable Automation Enabled?	Select this checkbox if the application supports Current Tax Payable Automation. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Enables the `Config` cube for Custom dimensions Creates the `Current Tax Payable Automation` property.

Field	Description	Effects of Application Setting
Is Deferred Tax Expense to Equity Enabled?	Select this checkbox if the application supports Deferred Tax Expense to Equity. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Creates the `DefExpToEquity` property.
Is Interim Provision Enabled?	Select this checkbox if the application supports Interim Tax Provision. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Adds the `InterimForecast` scenario in the **Scenarios** field. Creates the `AEETRGlobalEntity` property.
Is Net Operating Loss Automation Enabled?	Select this checkbox if the application supports Net Operating Loss (NOL) Automation. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Enables the `Config` cube for Custom dimensions
Is Ownership Management Enabled?	Select this checkbox if Ownership Management is enabled in the application. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Creates the `NCI Account` property. Updates smart list values.
Is Pillar Two Enabled?	Select this checkbox if Pillar Two is enabled in the application. For more information about this setting, see Working with Pillar Two in *Administering Tax Reporting*	Enables the `Config` cube for Custom dimensions Creates these properties: • `Deferred Tax Movements Pillar Two` • `PillarTwoEntity` • `Automation Source Movement Pillar Two` • `PillarTwoCurrency` Updates smart list values.

Field	Description	Effects of Application Setting
Is Safe Harbor Enabled? **Note:** This option is available only when **Is Pillar Two Enabled?** is selected.	Select this checkbox if the application supports Safe Harbor rules. For more information about this setting, see Working with Transitional Safe Harbor Rules in *Administering Tax Reporting*	Updates smart list values.
Is Weighted Average Enabled?	Select this checkbox if the application supports Net Operating Loss (NOL) Automation. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Creates these properties: - `WARSourceAccount` - `Weighted Ownership Account`
Is Scenario - Entity Tax Rate Change to Equity Enabled?	Select this checkbox if the application supports the Scenario - Entity Tax Rate Change to Equity feature. For more information about this setting, see Enabling Application Features in *Administering Tax Reporting*	Creates `{Scenario}_RC` properties, where {Scenario} is the name of each scenario. For example, `Actual_RC`.
Scenarios **Note:** This option is available only when **Is Scenario - Entity Tax Rate Change to Equity Enabled?** is selected.	Enter the scenarios for your application.	The `Actual` scenario is added by default. If you selected the **Is Interim Provision Enabled?** checkbox, the `InterimForecast` scenario is added.
Enter the Additional Alias Tables	Enter any additional alias tables for the application.	
Additional Smart Lists	Enter any additional smart lists for the application.	

2. Register the dimensions.

   To register one or more dimensions, click **Add**, and then enter the following information:

Field	Description
Name	Enter a dimension name. The value you enter becomes the name of the viewpoint, node set, hierarchy set, node type and dimension in Oracle Fusion Cloud Enterprise Data Management. **Note:** **Name** is a display name in Cloud EDM only. The **External Dimension Name** gets exported to Tax Reporting. The external dimension names for dimension types other than Custom are fixed to the dimension type (for example, **Account**). For Custom dimensions, you enter the external dimension name. See **External Dimension Name** , below.
Description	Optionally enter a dimension description.
Dimension Type	Select the dimension type: - Account - Entity - Movement - Jurisdiction - From Currency - Custom
Allow Shared Nodes	Select if the dimension you are managing has shared nodes.
Cubes (Custom dimension type only) **Note:** For each cube that you select, a set of cube-specific properties is created (for example, Plan Type).	(Optional): Select the valid cubes for the custom dimension. The cubes that you can select from depend on the application settings that you selected, as follows: - **Consol**: Always available - **Rates**: Available if you selected **Is this application Multi-Currency?** - **CbCR**: Available if you selected **Is CbCR Enabled?** - **Config**: Available if you selected any of these settings: – **Is Current Tax Payable Automation Enabled?** – **Is Net Operating Loss Automation Enabled?** – **Is Pillar Two Enabled?**
External Dimension Name (Custom dimension type only)	If the dimension type is Custom, enter the external dimension name.

3. Click **Next**.
4. Optionally add one or more attributes to the dimension:

   > **Note:**
   >
   > When you register an application, a number of attributes are predefined for you depending on your registration settings. See Predefined Attributes for Tax Reporting Applications. You can optionally add one or more additional attributes during this registration step.

   - For simple attributes:
     - Select **Simple Attribute**.
     - Enter an attribute name.
     - Enter all values you want to use for the attributes.

       > **Note:**
       >
       > The attribute name is put into the `Attribute - name` property and the values are put into a pull-down list.
   - For attribute dimensions:
     - Select **Attribute Dimension**.
     - Enter an attribute dimension name.

       > **Note:**
       >
       > The attribute dimension name is put into the `Attribute Dim - name` node data type property and the viewpoint for the attribute dimension is added to the default view for the application.
     - For **Attribute Type**, select Text, Numeric, or Date.

     See Understanding Attributes.
   - For registered dimensions:
     - Select **Registered Dimension**.
     - Enter an attribute name.

       > **Note:**
       >
       > A node data type property is created with the name that you enter in order to reference the registered dimension.
     - Select the dimension that contains the attribute values that you want to reference from the **Dimension** drop down menu.

> **Note:**
>
> For simple attributes,attribute dimensions, and registered dimensions, the attribute name cannot be the same as any of the predefined account names (Account, Entity, Movement, Jurisdiction, From Currency) or any of the external custom dimension names that you entered. Additionally, the name cannot be one of the predefined attribute dimension names: Predefined Attributes for Tax Reporting Applications.

5. Review the summary, and then click **Apply**.
   The dimension and its data chain are created. The data objects are bound to the dimension and set to active status. You can now import dimension data, see Importing Tax Reporting Dimensions.

Predefined Properties for Tax Reporting Applications

The following properties are predefined for all Tax Reporting applications.

Property	Data Type	Inherited	Level	Description
Access Granted to Member Creator	String	No	Node	Determines the access that member creators have to dynamic members that they create
Account Type	String	Yes	Node	Defines account time balance (how values flow over time) and determines accounts sign behavior for variance reporting with member formulas
Alias: {Alias Table}	String	No	Node	Alternate unique description for the dimension member
Allow Upper Level Entity Input	Boolean	No	Node	Specify whether parent entity input is allowed for this member.
Base Currency	String	Yes	Node	Specifies the base currency for an entity dimension member in a Standard multiple currency application
Data Storage	String	No	Relationshi p	Specifies whether data is stored or calculated for a dimension member
Data Type	String	Yes	Node	Specifies the type of data values for a dimension member
Enable for Dynamic Children	Boolean	No	Node	Enables users to create children for this member by entering a member name in the runtime prompt for a business rule that has been configured with a dynamic parent member

Property	Data Type	Inherited	Level	Description
Enterprise Predefined Member	Boolean	No	Node	Designates a dimension member as predefined by the application. Predefined members are restricted to only specific types of changes such as moves, reorders, or adding a child under them. **Note**: This property is set by the import process when importing predefined members from a Tax Reporting application and it cannot be manually edited in Oracle Fusion Cloud Enterprise Data Management.
Exchange Rate Type	String	Yes	Node	Determines how exchanges rates are used to calculate values
Formula	Memo	No	Node	Member formula used to calculate a value for a dimension member.
Formula Description	String	No	Node	Description for a member formula
Hierarchy Type	String	No	Node	Available for dimension bound to an aggregate storage cube. Aggregate storage dimension are automatically enable to support multiple hierarchies. The first hierarchy in a multiple hierarchy dimension must be stored.
Number of Possible Dynamic Children	Integer	No	Node	Maximum number of dynamic children that can be created for a dimension member. This option is available if Enable for Dynamic Children is selected.
Process Management Enabled	Boolean		Node	Determines whether process management is enabled
Skip Value	String	Yes	Node	Determines how database calculations treat zeros and missing values when Time Balance is First, Balance, or Average.
Smart List	String	No	Node	Select a smart list to associate to a dimension member.
Source Plan Type	String	Yes	Node	Source plan type for a dimension member
Time Balance	String	Yes	Node	Specifies how the application calculates the value of summary time periods
Two Pass Calculation	Boolean	No	Node	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Data Storage of Dynamic Calc or Dynamic Calc and Store.

Property	Data Type	Inherited	Level	Description
UDA	List	No	Node	User defined attribute used for calculation or reporting purposes
Variance Reporting	String	No	Node	Determines whether an account is treated as an expense during variance reporting
Weekly Distribution	Boolean	No	Node	Sets the weekly distribution. Available for bottom level account members if the option was selected when creating the application and the base time period is 12 months.

The following properties are predefined for each cube in your application:

Table 35-1 Tax Reporting Properties by Cube

Property	Data Type	Inherited	Level	Description
Aggregation {Cube}	String	Yes	Relationship	Determines how child member values aggregate to parent members
Data Storage {Cube}	String	No	Relationship	Specifies whether data is stored or calculated for a dimension member
Formula {Cube}	Memo	No	Node	Member formula used to calculate a value for a dimension member.
Formula Description {Cube}	String	No	Node	Description for a member formula
Plan Type {Cube}	Boolean	Yes	Relationship	Specifies whether the dimension member is valid for the plan type.
Solve Order {Cube}	Integer	No	Node	Specifies the order in which formulas are evaluated. The formulas for members that have a specified solve order are calculated in order from the lowest solve order to the highest.

Predefined Attribute Properties

See Predefined Attributes for Tax Reporting Applications for a list of attribute properties that are created when you register a Tax Reporting application. Note the following about attribute properties:

- They use the `TRCS` namespace.
- They are all at the Node level.
- They are all string data types, except for `FX Rates - Global Account Rate Override` and `Plug Account`, which are node data types.

Predefined Attributes for Tax Reporting Applications

The following predefined attributes are created as properties when you register a Tax Reporting application.

Considerations

- Some attributes are enabled by default, and others are enabled based on your registration settings. The registration setting that enables the attribute is listed in the table below.
- All attributes allow blank values.
- You can add additional attributes by modifying the application registration. See Modifying Registered Tax Reporting Applications.

Table 35-2 Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
AEETRGlobalEntity	**Is Ownership Management Enabled?**	This attribute is used to identify the entity for which you will use the AEETR within a hierarchy.	Entity
ApplyWARRATE	**Is Weighted Average Enabled?**	This attribute is used to apply the specific Weighted Average Rate type for any book Account, so that on translation the specified book Account will have the same Weighted average rate applied.	Account
Automation Source Movement Pillar Two	**Is Interim Provision Enabled?**	This attribute is used to identify a movement as source movement in pillar two automation.	Entity
CbCR	Enabled by default	This is deprecated in Tax Reporting. See Defining Account Tax Attributes in *Administering Tax Reporting*	Account

Table 35-2 (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
Classifications	Enabled by default	This attribute is used to set level of classification for deferred tax purposes.	Account
Consolidated ETR	Enabled by default	This attribute defines that this account and the data will be included in the Consolidated Effective Tax Rate Reconciliation (CETR).	Account
Current Tax Payable Automation	**Is Current Tax Payable Automation Enabled?**	This attribute is used to assign a Current Tax payable movement for automation from Current Provision.	Entity
Deferred Tax Movements Pillar Two	**Is Pillar Two Enabled?**	This attribute identifies the total tax effect of a Deferred Tax movement less valuation allowance which will be automated to the Pillar Two Account for which it is being assigned (i.e. Other Adjustments).	Account
DefExpToEquity	**Is Deferred Tax Expense to Equity Enabled?**	This attribute is used to identify the Regional jurisdictions that the Deferred Tax Expense will be reclassified as an equity adjustment. For example, when set on US_Blended, a new attribute becomes available for all entities that allows the setting of DefExpToEquityUS_Blended_Yes to reclassify the Deferred Tax Expense for the US_Blended Regional Deferred Tax calculation to Equity.	Jurisdiction
Domicile	Enabled by default	This attribute is used to identify the principal place of business of the Entity, and to identify which Jurisdiction member is valid for the Entity.	Account

Table 35-2   (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
Equity Reversal Target Account	Enabled by default	This attribute is used to identify the additional provision account where the tax effect of the permanent or temporary difference account will be reversed for equity accounted tax.	Account
Equity Reversal Target Account Regional	Enabled by default	This attribute is used to identify the regional additional provision account where the tax effect of the regional permanent or temporary difference account will be reversed for equity accounted tax.	Account
FBOS	Enabled by default	This attribute defines that "No federal benefit of state" would be applied on this account.	Account
FX Rates - Global Account Rate Override	**Is this application Multi-Currency?**	This attribute identifies an account that an FX override rate is required.	Account
Intercompany Account	Enabled by default	This attribute is used to identify the Intercompany account that is that is set as "Plug account".	Account
Intercompany Entity	Enabled by default	For Entity members, you specify if the member stores Intercompany detail. If you set the Intercompany property (ICP_Entity_Yes) for an Entity member, a member with the same name is created in the Intercompany dimension with the name ICP_<Entity Name>.	Entity
Is Plug Account	Enabled by default	Specifies if this account is a Plug account.	Account

Table 35-2 (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
Is Switch Sign	Enabled by default	This is deprecated in Tax Reporting. See Defining Account Tax Attributes in *Administering Tax Reporting*	Account
NCI Account	**Is Ownership Management Enabled?**	This attribute is used to identify the NCI Account that is a permanent difference where the non-controlling interest of pre-tax income will be posted	Account
NIBT FX Override Account	**Is this application Multi-Currency?**	This attribute is used to identify the NIBT Account is translated using and FX Override Rate.	Account
NIBT FX Override Difference Account	Enabled by default	This attribute is used to identify the NIBT Override Account, a permanent difference account where the difference between the NIBT FX Override Rate and the Average Rate * NIBT is reflected.	Account
Override FX Translation Method	**Is this application Multi-Currency?**	Tax Reporting provides translation methodology overriding on an account-by-account basis.	Account
Parent Tax Calculations	Enabled by default	Parent Tax Calculations is set up when you require tax calculations to be run at the parent level.	Entity
PillarTwoCurrency	**Is Pillar Two Enabled?**	This attribute is used to identify the input currency for the Pillar Two jurisdiction	Jurisdiction
PillerTwoEntity	**Is Pillar Two Enabled?**	This attribute is used to identify an entity that is in-scope for Pillar Two Global Minimum Tax purposes	Entity

Table 35-2 (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
Plug Account	Enabled by default	For the accounts that have Intercompany set to Yes, and IsPlugAccount is not set, you can specify a Plug account.	Account
Rate change to equity	Enabled by default	This attribute defines that impact from change in Tax rates data will be reversed and applied to the Impact from Change in Tax Rates in Equity column for all the temporary difference accounts with this attribute.	Account
Regional Deduction	Enabled by default	Regional Deduction is set when you require the Regional tax expense to be deductible on the National Tax Provision.	Entity
RTA Prior Year Rate	Enabled by default	This attribute when assigned to the RTA-Deferred Only movement allows you to tax effect data using Prior year Tax rate, Prior year Closing Rate, or Prior year difference between P12 and P13 tax rates	Movement
{Scenario}_RC, where Scenario is a scenario name. For example, Actual_RC.	**Is Scenario - Entity Tax Rate Change to Equity Enabled?**	Creates an attribute property for each of the scenarios in the application. See Defining Entity Tax Attributes in *Administering Tax Reporting*	Entity
TAR Account Classification	Enabled by default	This attribute sets newly created TAR accounts to be calculated as Domestic or Foreign.	Account
TAR Book Account National	Enabled by default	This is deprecated in Tax Reporting. See Defining Account Tax Attributes in *Administering Tax Reporting*	Account

Table 35-2 (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
TAR Book Account Regional	Enabled by default	This is deprecated in Tax Reporting. See Defining Account Tax Attributes in *Administering Tax Reporting*	Account
TAR Difference Reclassification	Enabled by default	This attribute is used to identify the TAR journal difference posting account under the Tax Automation RollForward (TAR) account hierarchy. The difference between the ending balance and the ending balance as per book account will get stored into the mapped account.	Account
TAR Jurisdiction Classification	Enabled by default	This attribute is used to identity the jurisdictions that are domestic. Any jurisdiction without the attribute set will automatically be classified as foreign	Jurisdiction
Tax Automation Source Movement	Enabled by default	Reserved for future use in Tax Reporting. See Defining Movement Tax Attributes in *Administering Tax Reporting* .	Movement
Tax Automation Target Movement	Enabled by default	This attribute when assigned to a deferred Tax movement , allows a user to target a source data with Tax Automation and Tax Automation Regional.	Movement
Tax Data Type	Enabled by default	This attribute defines how data is entered in the application for the accounts - PreTax, Tax and Apportioned.	Account
Tax Type	Enabled by default	Classifies whether the data present in the account is of type National data or Regional data or of both National and Regional.	Account

Table 35-2 (Cont.) Tax Reporting Predefined Attributes

Property	Registration Setting to Enable	Description	Dimension
Translation Type	Enabled by default	This attribute when assigned to a movement allows a user to translate data using end of period rate.	Movement
WARSourceAccount	**Is Weighted Average Enabled?**	This attribute associates the tax expense accounts that will be used to calculate the Weighted Average rate for other Tax and Book accounts.	Account
Weighted Ownership Account	**Is Weighted Average Enabled?**	This attribute is used to identify the account used for weighting ownership as ownership changes over time. If no attribute is set the TRCS_NIBTAdjusted account is used.	Account

Importing Tax Reporting Dimensions

For Tax Reporting dimensions, Oracle Fusion Cloud Enterprise Data Management supports importing from comma-delimited (CSV) files and from an external application outbox using a `.zip` file.

On import, the property `Enterprise Predefined Member` is set to `True` for predefined members and cannot be edited, with the exception moves, removes or adding a child under it. See Seeded Dimension Members in *Administering Tax Reporting* . If the predefined members in the external application are changed, you must re-import the dimensions to update the predefined members in Cloud EDM.

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- In the external application, create a data file to import using one of these formats:
  - If you are using a local file, a `.csv` (comma-delimited) file for each dimension. The local `.csv` filename does not have any file naming restrictions.
  - If you are connecting to the external application outbox, a `.zip` file that contains `.csv` files for one or more dimensions. The `.csv` filename must contain the dimension name or external dimension name, if specified in the registration, at the end with preceding text. For example, `acc1.csv` is invalid, but `tom_smith_acc1.csv` is valid, where `acc1` is the dimension name.

> **Note:**
>
> From Tax Reporting, create the files to import into Cloud EDM. See Exporting Metadata in *Administering Tax Reporting*

.

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. Click the dimension that you want to import data into.
3. In **Input Source**, select **File** or **Connection**, and then do one of the following:
   - If you selected **File**, specify an import file name.
   - If you selected **Connection**, select a connection to import from, and specify an import file name.
4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.
5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*), a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

Tax Reporting Import and Export File Format

When you import and export data for Tax Reporting dimensions, you must use a comma-delimited (`.csv`) file.

The following list describes the comma-delimited file's format and other considerations:

- The file must have the following columns:
  - **Name** The dimension name specified in the registration or the external dimension name, if you entered it.
  - **Parent** The name of the parent node. If there is no parent node, this field must be empty.
- **Alias:** *name*, where *name* is the name of a the alias table you registered. You must have at least one alias table.
- These are common columns:
  - **Plan Type** (*cube name*), **Aggregation** (*cube name*), **Data Storage** (*cube name*), **Formula** (*cube name*), **Formula Description** (*cube name*), these are typically defined for each cube you registered.
  - **Solve Order** *(cube name)* this is defined for each ASO cube you registered.

> **Tip:**
>
> *cube name* is the name of the cube you registered.

- **Smart List** The smart list names are optionally specified for nodes.
- **Attribute Dimension Name** The name of the attribute dimension entered during registration.

> **Note:**
>
> Additional columns can be imported and exported from the external application, see Editing Dimensions in the Simplified Dimension Editor and Editing Member Properties using the Simplified Dimension Editor in *Administering Tax Reporting* .

- The import file's data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a child node's row precedes its parent node's row, an error occurs when the file is imported.
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)
- The import ignores duplicate rows.

> **Note:**
>
> The import contains the logic to process shared nodes (Data Storage = Shared) based on the requirements of Tax Reporting.

Modifying Registered Tax Reporting Applications

After you register a Tax Reporting application, you can modify it. For example, you can register additional dimensions, remove dimensions, or change the multi-currency setting. The application, dimension, and data objects are updated based on your changes. The dimension binding rules are updated to reflect the modifications you make.

> **Note:**
>
> If you remove an attribute dimension or alias, its corresponding property is unbound from the dimension and is no longer imported or exported. For example, if you remove the Default alias dimension, the corresponding property, `Alias: Default`, is no longer imported or exported for the node type. The property remains on the node type until you remove it.
>
> If you remove a dimension, the corresponding data chain objects are unbound from that dimension but they are not deleted. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for a Tax Reporting application:

1. From **Applications**, find your application, click ⋮, and then select **Modify Registration**.
2. To change or add application settings or dimensions, see Registering Tax Reporting Application Settings and Dimensions.
3. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

> **Note:**
>
> If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
>
> If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Binding Rules for Tax Reporting Applications

Binding rules are essential because they automatically make the Oracle Fusion Cloud Enterprise Data Management application conform to the requirements of the external application.

The following table describes the binding rules for Tax Reporting.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.
Hierarchy Set	For bound hierarchy sets, you cannot select a node type that is missing required dimension properties unless you confirm that the system may add the properties when you save.
Node Set	For bound node sets, you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. The hierarchy set is checked and then the node types are checked.

Object	Rule
Dimension	When you edit a viewpoint bound to a dimension: • You can select a different viewpoint for the binding to a dimension. The new viewpoint is checked for binding rules as are all data objects up the data chain. **Note:** You can also remove a dimension by deleting it from the application registration. The dimension still displays in the Application Inspector, and data chain objects may be created using the dimension, but it cannot be used for importing or exporting data. • In the application import and export sections, you must select a dimension that passes its own binding rules or any binding rules up the data chain.

Binding a Viewpoint to a Dimension in Tax Reporting Applications

Tax Reporting applications support binding a viewpoint to a single external dimension. If you create an alternate viewpoint that you want to bind to the external dimension instead, you can edit the binding to point to the alternate viewpoint by taking the following steps.

For more information, see:

- Understanding Bindings and Bound Data Objects
- Understanding Viewpoints

1. From **Applications**, find the application containing the viewpoint.
2. In the application's **Actions** column, click ⋮, then select **Inspect**
3. Select Dimensions, and then click the name of the dimension to which you are binding the viewpoint.
   The dimension inspector is displayed.
4. Select Bindings, and click the binding to associate with the alternate viewpoint.
5. Click **Edit**.
   The **View** and **Viewpoint** drop-down lists become active.
6. From **View**, select the view containing the viewpoint you want to bind to the dimension.
7. Select the viewpoint.
8. Click **Save**.

Tax Reporting Validations

Validations ensure data integrity. The following validations run for Tax Reporting applications.

Note:

Validations in the external Tax Reporting application are implemented as Recommendations and are not enforced on the simple or classic dimension editors. By default, Tax Reporting validations in Oracle Fusion Cloud Enterprise Data Management are implemented with the Error severity. You can change the severity to Warning. See Configuring Validation Enforcement and Severity.

In addition to these validations, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

Note:

For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

Name	Source Type	Description	Scope
Alias Application Uniqueness	Tax Reporting	Alias is not unique across the dimensions in the application	Bound node types across the application
Alias Uniqueness	Planning	Alias are not unique among a group of sibling nodes	Siblings
Consolidated ETR (Account dimension only)	Tax Reporting	Descendants of any of the following nodes have the CETR value assigned to the Consolidated ETR attribute: • TRCS_TempGSTotal • TRCS_TempSTTotal • TRCS_TaxLossesCFSTotal • TRCS_TaxCreditTotal • TRCS_VATotal • TRCS_RTempTotal • TRCS_RTaxLossesPreCFSTotal • TRCS_RTaxCreditTotal • TRCS_RVATotal • TRCS_RTaxAttribTotal	Viewpoint
Cube Data Storage	Planning	Default Data Storage is Shared or Label Only but Data Storage for cube is not Shared or Label Only	Node
Current Additional Total Accounts Shared (Account dimension only)	Tax Reporting	Bottom node descendants of TRCS_CurrentAddtlTotal must also be shared descendants of TRCS_CurrentAddtlETRTotal	Viewpoint
Data Storage	Planning	Only one instance of a node can be set to non-shared data storage. All other instances of the node must be Shared.	Viewpoint

Name	Source Type	Description	Scope
Date Attribute Name	Planning	Bottom level node names in a date attribute dimension are not in `mm-dd-yyyy` format	Property
Domicile (Entity dimension only)	Tax Reporting	Domicile has to be set for all bottom node entities and also for the parent entities which have Parent Tax calculation enabled.	Viewpoint
Exchange Rate Type	Planning	Exchange Rate Type is not None and Data Type is not Currency	Node
Formula Allowed	Planning	Data Storage is not Dynamic Calc or Dynamic Calc and Store	Node
Intercompany Entity Validation (Entity dimension only)	Tax Reporting	Intercompany should be set on bottom entities only	Node
Member Name Conflicts	Planning	Member name is the same as a dimension name, a cube, or an attribute dimension defined for the current application	Node and data chain
Name Uniqueness	Planning	Member name is not unique across all nodes used in an application	Bound node types across the application
NCI Account (Account dimension only)	Tax Reporting	NCI Account should only be set on a Permanent Difference Stat to Tax or GAAP to Stat Account	Node
Numeric Attribute Name	Planning	Bottom level node names in a numeric attribute dimension are not in numeric format	Property
Plan Types	Planning	Plan Type is set to True but the parent does not have a True value **Note**: This validation is skipped if the parent is a predefined member.	Viewpoint
Plug Accounts (Account dimension only)	Tax Reporting	Intercompany Account must be Yes and Is Plug Account must not be set to define a Plug Account	Node
Predefined Member	Planning Modules	If the Enterprise Predefined Member property equals `True` for a node, it can only be moved or reordered. You cannot edit the node or modify the Enterprise Predefined Member property on the node.	Node
Shared Member before Base Member	Tax Reporting	Shared members appear above base members in a hierarchy	Viewpoint

Name	Source Type	Description	Scope
Single NCI Account (Account dimension only)	Tax Reporting	At most one tax account should have the NCI Account set	Viewpoint
Smart List Required	Planning	Smart List is missing for member if Data Type is Smart List	Node
Source Plan Type	Planning	Source Plan Type is not a valid Plan Type for the member	Node
Tax Data and Tax Data Type (Account dimension only)	Tax Reporting	Bottom node descendants of TRCS_TaxAccounts must have both the Tax Type and Tax Data Type attributes assigned.	Viewpoint
Variance Reporting	Planning	Variance Reporting is not Expense when Account Type is Expense or Variance Reporting is Non-Expense if Account Type is anything other than Expense or Saved Assumption.	Node
Weighted Ownership Account (Account dimension only)	Tax Reporting	At most one tax account should have the Weighted Ownership Account set.	Viewpoint

Invalid Characters, First Characters, and Values for Tax Reporting

When you register a Tax Reporting application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` and `Alias` properties. For a list of these characters and values, see Restrictions for Dimensions, Members, Aliases, and Forms in *Administering Tax Reporting*

You can edit these values by editing the property parameters. See Editing Property Parameters

Exporting Tax Reporting Dimensions

You can export from a viewpoint that is bound to a dimension back to a Tax Reporting application.

Best Practice

It's a best practice to validate the viewpoint bound to the dimension prior to export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States, and all states. If the node set's top node is Texas, only Texas and the nodes under it are exported.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension that you are exporting from.
- When you register a Tax Reporting application, nodes and hierarchical relationships are imported and displayed in the order they were in the application. Nodes and hierarchical relationships are exported in the order they are in Oracle Fusion Cloud Enterprise Data Management. If you want nodes and hierarchical relationships to be exported in alpha numeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.
- You can edit the binding keys for a dimension to change the order of the export columns or to specify the direction of each column in the export file. See Editing Binding Keys.
- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- You export a dimension to a CSV (comma-delimited) file. You can export to a local file, or if you set up a connection to an external application during registration you can export to the Tax Reporting application inbox.

> **Tip:**
>
> You can view and test application connection settings by inspecting the application. See Inspecting Applications.

After you export, you can import the data into your external application dimension, see Importing Metadata in *Administering Tax Reporting* .

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. Select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:
    - If you selected **Connection**, select a connection to export to, specify an export file name, and then click **Export**.
    - If you selected **File**, an export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example TRCS_Account_20200123.csv, and can be edited before running the export. Your browser settings define the download location. Accept or edit the default file name, and then click **Export**.
5. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*), a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*), or Task Manager (see Automation Integrations for EPM Cloud in *Administering Financial Consolidation and Close*).

36

Working with Oracle Financials Cloud General Ledger Applications

You use Oracle Fusion Cloud Enterprise Data Management to manage chart of accounts segment values for value sets in Oracle Financials Cloud General Ledger. Users create new segment values, edit their properties, and maintain trees in Cloud EDM. This information can be exported to Oracle Financials Cloud General Ledger and imported using a scheduled process.

Videos

Your Goal	Watch This Video
Learn about working with Oracle Financials Cloud General Ledger applications.	Integrating Oracle Financials Cloud General Ledger.

The following types of General Ledger data can be managed in Cloud EDM:

- Chart of accounts segment values
- Segment value properties

> **Caution:**
>
> If you modify the predefined properties that get created by the registration system, do not use property inheritance or positional logic for deriving property values. See Predefined Properties for Oracle Financials Cloud General Ledger Applications.

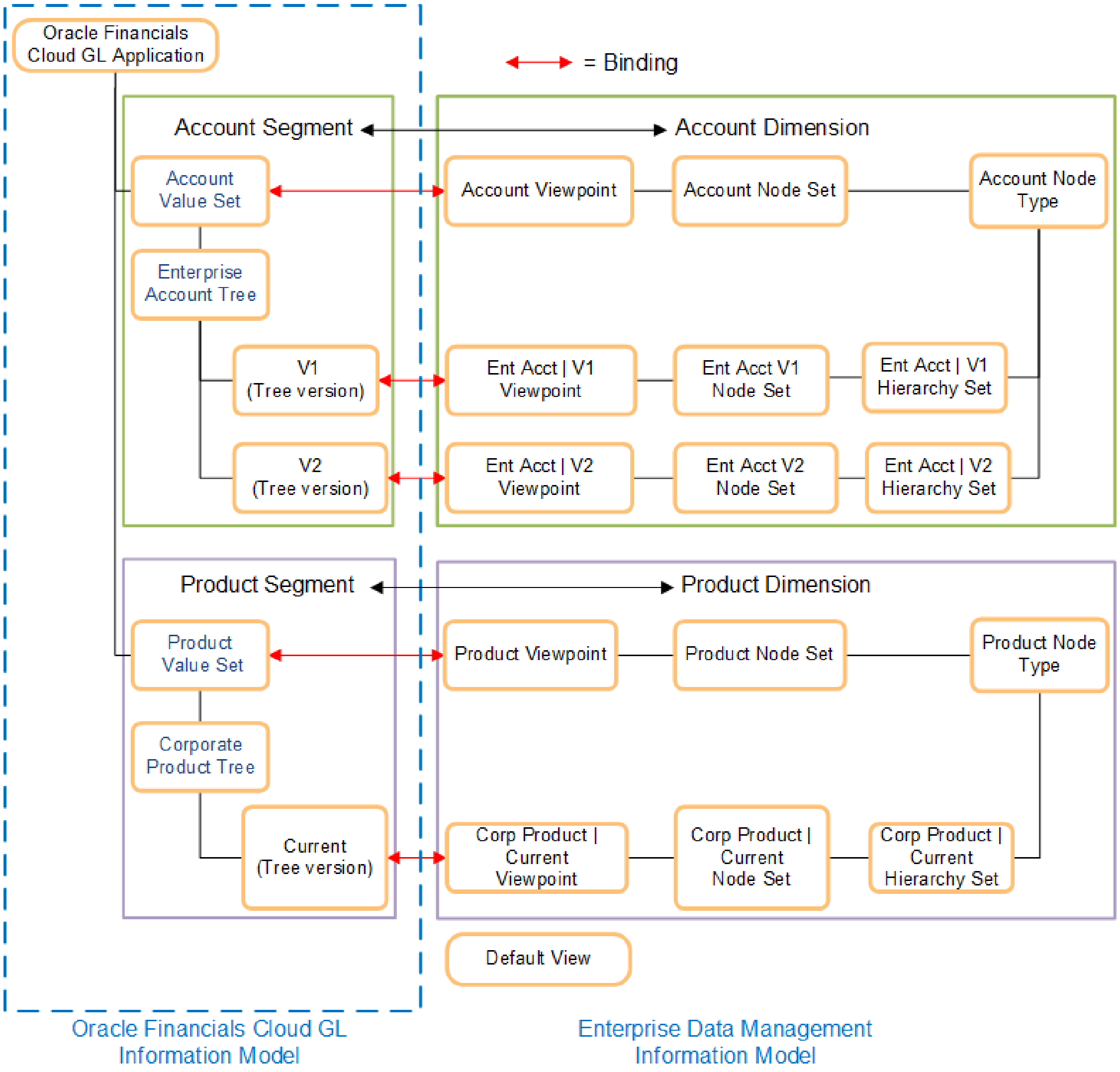
- Trees and tree versions
- Hierarchy relationships of segment values in tree versions
- Global and context-sensitive descriptive flexfields

For more information about Oracle Financials Cloud General Ledger concepts, see Financial Structures in *Implementing Enterprise Structures and General Ledger*.

Managing data for Oracle Financials Cloud General Ledger in Cloud EDM involves:

- Registering Oracle Financials Cloud General Ledger Applications
- Importing Oracle Financials Cloud General Ledger Dimensions
- Modifying Registered Oracle Financials Cloud General Ledger Applications
- Getting Started with Data Management
- Exporting Oracle Financials Cloud General Ledger Dimensions

You register Oracle Financials Cloud General Ledger applications by using a wizard that prompts you for information, such as the segment value sets and the tree codes and version names. The diagram below shows how the information model for an Oracle Financials Cloud General Ledger application is mapped to the information model in Cloud EDM.



When you register an Oracle Financials Cloud General Ledger application, these data chain objects are created for that application in Cloud EDM:

- **Default view**—enables you to manage all of the data chain objects for the application. See Understanding Views.
- For every segment value set in your application, these data chain objects are created:
  - **Dimension**—provide a way to categorize data values. See Understanding Dimensions.
  - **Viewpoint**—provides a subset of nodes for you to work with. The viewpoint points to the node set. See About Viewpoints.
  - **List-type node set**—defines the group of nodes available in a viewpoint. For segment value sets, a list node set is created that contains all of the available nodes without any hierarchical data. See Working with Node Sets.

  - **Node type**— displays a collection of an application's nodes that share a common business purpose. The node type has all of the application-specific properties. See Working with Node Types.
- For each combination of tree code and tree version name in your application, these data chain objects are created:
  - **Viewpoint**—provides a subset of nodes for you to work with. The viewpoint points to the node set. See About Viewpoints.
  - **Hierarchy-type node set**—defines the group of nodes available in a viewpoint. For every combination of tree code and tree version name, a hierarchical node set is created that contains the nodes for that tree code and version name in a hierarchy. The node set points to the hierarchy set. See Working with Node Sets.
  - **Hierarchy set**—defines the parent-child relationships for a group of nodes. The hierarchy set uses the node types for both parent and child nodes. See Working with Hierarchy Sets.

> **Note:**
>
> You do not have to register trees for a segment. If you do not register any trees for a segment, the export data file will not contain any hierarchical data. See Exporting Oracle Financials Cloud General Ledger Dimensions.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working on the data set, you can export the dimension back to the external application.

> **Note:**
>
> When the dimensions and data chain objects are created in Cloud EDM, they are bound to the segments, value sets, and trees in the Oracle Financials Cloud General Ledger application. These bindings ensure that the registered data chain objects conform to the external application requirements, and they prevent users from making changes that would prevent an application's data from being imported or exported. See Understanding Binding Rules.

Registering Oracle Financials Cloud General Ledger Applications

When you register an Oracle Financials Cloud General Ledger application into Oracle Fusion Cloud Enterprise Data Management, an application and default view are created to represent the external application. Dimensions, node types, and a list node set are created to represent segments and their associated value sets. Viewpoints, hierarchical node sets, and hierarchy sets are created to represent the trees and tree versions.

Videos

Your Goal	Watch This Video
Get an overview of how to register Oracle Financials Cloud General Ledger applications	Registering Oracle Financials Cloud General Ledger Applications

Your Goal	Watch This Video
Learn how to use Cloud EDM to manage chart of account segment values and trees in Oracle Financials Cloud General Ledger.	▶ Integrating Oracle Financials Cloud General Ledger

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

> **Note:**
>
> The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which segments and associated value sets in your chart of accounts you want to register. After you define your segments and value sets, you add the trees for each segment during the registration process. See Working with Oracle Financials Cloud General Ledger Applications.

To register an application:

1. From **Applications**, click **Register**.
2. Select **Financials Cloud General Ledger** as the application type.
3. Enter an application **Name** and, optionally, a **Description**.

> **Note:**
>
> The application type, name, and description are displayed in the Application list.

4. If you are connecting to an external Oracle Financials Cloud General Ledger server, click **Add**, and define the connection settings. When you export data from your Cloud EDM application to a comma-delimited file, you can export to the external application server, where you can then import it into your Oracle Financials Cloud General Ledger application. See Exporting Oracle Financials Cloud General Ledger Dimensions.
   You can add multiple application connections.

Field	Valid Values	Example: connecting to an external application
Connection Name	Enter a required connection name that is unique to this application. **Note:** The connection name cannot contain spaces.	US Financial Cloud GL Connection
Connection Description	Enter an optional description.	Use this connection to connect to an Oracle Financials Cloud General Ledger server.
Instance Location	Enter a connection URL to the Oracle Financials Cloud General Ledger service in the format: `https://`*`ServiceName-IdentityDomain-ServiceType.DataCenter`*`.`**`oraclecloud.com`**	`https://fincloudgl-uscompany-fcgl.US11.oraclecloud.com` where • `ServiceName is fincloudgl` • `IdentityDomain is uscompany` • `ServiceType is fcgl` • `DataCenter is US11`
Identity Domain	Enter the identity domain of the external application. **Note:** This is the same Identity Domain used in the **Instance Location**.	`uscompany`

Field	Valid Values	Example: connecting to an external application
Username	Enter an authorized user name **Caution:** For exports to a connection, the authorized user must have the `FUN_FSCM_REST_SERVICE_ACCESS_INTEGRATION_PRIV` privilege in Oracle ERP Cloud.	`tom.smith` **Note:** You must enter the identity domain in one of these fields: **Identity Domain** or **Username**. For example, enter `uscompany` in **Identity Domain** or `uscompany.tom.smith` in **Username**.
Password	Enter a password.	xxxxxxxx
Primary Connection	Select this check box to make this the default for exporting to the external application.	

5. Click **Create**.
6. In **Active Languages**, enter the languages that have an Active status in the Oracle Financials Cloud application.

   > **Note:**
   >
   > As you type the name of a language, a list of supported languages that match your text entry are displayed. Select the language to add it to the Active Languages list.

7. Select a single **Base Language** for the application. You can choose from any of the languages that you added as active languages in the previous step.
8. **Optional:** Select the check boxes next to each field that apply to your application:
   - **Allow Multiple Root Nodes**: enables more than one root node in the hierarchy.
   - **Allow Duplicate Nodes**: enables shared nodes. See Understanding Shared Nodes.

     > **Note:**
     >
     > By default, duplicate nodes are not enabled. They should only be enabled for trees that will not be published in Oracle Financials Cloud (that is, they do not have the **Publish Hierarchies to Cube** export option enabled. See Managing Export Options.).

9. In **Max Depth**, enter the depth limit for the data in your tree structure, or enter `-1` if your tree structure does not have a depth limit.
10. In **Tree Labels**, optionally add one or more allowed values for tree labels. The labels that you enter are added as allowed values for the `FCGL.Tree Label` property.
    Tree labels can be a maximum of 50 characters each, and they cannot contain leading or trailing spaces.
11. Continue the registration. See Adding Global Descriptive Flexfields.

Adding Global Descriptive Flexfields

You can define up to 50 descriptive flexfields during registration, and then import and export them to Oracle Financials Cloud General Ledger.

About Descriptive Flexfields

Descriptive flexfields are user-defined fields that enable you to capture additional attributes about segment values. For example, a Product segment could contain descriptive flexfields for Color and Size. There are two types of descriptive flexfields:

- Global descriptive flexfields are applied to all segments in an application, and both the flexfield and its associated property must be unique. You cannot reuse either the name or the property of a global descriptive flexfield in either another global or a context-sensitive flexfield.
- Context-sensitive descriptive flexfields are applied only to specific segments, and both the flexfield and its associated property can be reused in other segments.

To illustrate the usage of global versus context-sensitive descriptive flexfields, suppose you have three segments: Account, Entity, and Product. If you have an attribute that applies to all three segments, you can create a global descriptive flexfield and it will be added to all of your segments. However, if you have an attribute that applies to only two of the segments, you can create a context-sensitive flexfield and add it only to the two segments to which it applies.

For more information about descriptive flexfields, see Overview of Flexfields in *Oracle Applications Cloud Configuring and Extending Applications*.

To add a global descriptive flexfield:

1. In Add Global Descriptive Flexfields, click **Add**.
2. Enter the **Flexfield Name**, and then specify whether or not the flexfield is required. The name must be unique for global descriptive flexfields.
3. In **Column**, click the drop down menu and select the attribute location (1-50) for the flexfield. This field determines the location of the value for this flexfield in the import and export files.
   When you select an attribute location for either a global or a context-sensitive descriptive flexfield, that location is no longer available to be selected by another flexfield. For example, if you select the `Attribute 3` location for a global descriptive flexfield, when you add another global or context-sensitive descriptive flexfield, `Attribute 3` is no longer displayed in the drop down menu.
4. In **Property**, click the drop down menu and perform an action:
   - To create a new string property for the flexfield, select **(Add New Property)** and then click **Next**. Continue to Adding a New Property for a Flexfield.

> **Note:**
>
> When you create a property for the flexfield, it is always created with the string data type. If you want to use a Boolean, memo, or node property for the flexfield, you must create the property first prior to registration (see Creating Properties Manually) and then select it as an existing property (see below).

- Select an existing property for the flexfield and then click **Next** to return to Add Descriptive Flexfields. The property that you select must be a node level property with the string, memo, node, or Boolean data type.

> **Note:**
>
> When you select an existing property for a global descriptive flexfield, that property is removed from the drop down list and you can no longer select it for other global or context-sensitive descriptive flexfields.

Adding a New Property for a Flexfield

1. By default, the property name is "DFF {Flexfield Name}". Optionally, change the name or enter a description for the property, and then click **Next**.
2. Specify the parameters for the property:

Parameter	Description
String Case	Specify whether the property should be in `Upper Case`, `Lower Case`, or `Mixed Case`.
Invalid Characters	Specify the invalid characters for the property. Click in the **Invalid Characters** field and then select the invalid characters from the provided list.
Minimum Length	Enter the minimum text length for the property. The default value is 0.
Maximum Length	Enter the maximum text length for the property. The default value is 240, and you cannot increase the maximum past 240.
Use Allowed Values List	Select to restrict entry in the property field to only those values listed in **Allowed Values**.
Include Blank Entry	Select to allow this property to be left blank.
Allowed Values	Enter the values that you want to be displayed for this property. Enter a value and then hit `Tab` to enter the next value. Select **Use Allowed Values List** to restrict user entry only to the values in this field.
Default Value	Enter a default value for the property.
Inheritance	Select whether or not to use positional inheritance for this property. See Inheriting Properties.

3. Click **Next** to return to Add Global Descriptive Flexfields. The global descriptive flexfield that you added is displayed with a default name of the attribute location and the flexfield name that you entered, for example, "Attribute 1 - Color".
4. Click **Add** to add another flexfield, or click **Next** to continue the registration. See Registering Segments and Trees.

Registering Segments and Trees

Use these steps to register segments and trees for Oracle Financials Cloud applications.

Registering Segments

To register segments:

1. In **Add Segments**, click **Add**.
2. Enter the **Value Set Code** for the segment, and optionally, the **Description**.

   > **Caution:**
   >
   > Value Set Codes are case-sensitive in the General Ledger system. Ensure that you enter the Value Set Codes in Oracle Fusion Cloud Enterprise Data Management exactly as they are defined in General Ledger.
   > If you have entered a Value Set Code incorrectly, you must rename it by modifying your application registration (see Modifying Registered Oracle Financials Cloud General Ledger Applications). Do not rename a Value Set Code from the dimension inspector.

3. Select the check box next to **Natural Account Segment** if this segment is used as the natural account in a chart of accounts.
4. In **Value Type**, specify if the values in the value set are text or numeric.
5. Enter the **Maximum Character Length** for the values in the value set, and click **Next**.
6. Perform an action, based on the value type that you selected in step 4:
   - For text value types:
     a. Enter a minimum and maximum value for the values in the value set. You can enter alphanumeric or numeric values. For example, if you enter `XXXXX` in the minimum field and `99999` in the maximum field, users can enter values starting with X, Y, or Z, or any numeric value up to 99999.
     b. Select **Upper Case** if the value set values must contain only capital letters.
     c. Select **Zero Fill** to specify that the value must have as many characters as the maximum character length.
        For example, if the maximum character length is 5 and a user enters a value of `AAA`, if zero fill is enabled the system indicates that the user must modify the value to contain five characters (such as `00AAA`). If zero fill is not enabled, the system will accept the value as entered.
   - For numeric value types, enter minimum and maximum values for the values in the value set.

> **Note:**
>
> Zero Fill is enforced automatically for numeric value types. If the maximum character length is 4 and a user enters `222`, the system indicates that the user must modify the value to contain four characters (such as `0222`).
>
> If you do not want to enforce Zero Fill on a numeric value, you can change the value type to text and enter numeric minimum and maximum values. This enforces that users enter numeric values but does not enforce zero fill on those values.

7. Click **Next**.

Adding Context Sensitive Descriptive Flexfields

Context-sensitive descriptive flexfields enable you to capture additional attributes about segment values for specific segments only. The procedure for adding a context-sensitive descriptive flexfield to a segment is similar to adding a global descriptive flexfield, with these differences:

- Context-sensitive descriptive flexfields can be reused for more than one segment. If a context-sensitive descriptive flexfield applies to more than one segment, you must add it to each segment to which it applies.
- The properties associated with context-sensitive flexfields can also be reused. Unlike global descriptive flexfields, when you select a property to associate with a context-sensitive flexfield, that value does not get removed from the **Property** drop down menu and you can select it again for a different context-sensitive flexfield.

See Adding Global Descriptive Flexfields.

Registering Trees

Next, you define the trees for the segment. Trees consist of tree codes and tree version names. Each tree code can have multiple tree versions.

> **Note:**
>
> You do not have to enter a tree for a segment. If you do not enter at least one tree for a segment, when you export the segment the `GLSegmentHierInterface.csv` file containing hierarchy data will not be generated. See Exporting Oracle Financials Cloud General Ledger Dimensions.

To define trees for the segment:

1. Click **Add**.
2. Enter the **Tree Code** and the **Tree Version Name**.
3. Enter the starting and ending dates for the version, and click **Next**.

> **Note:**
>
> The ending date must be equal to or later than the starting date.

4. **Optional**: Select **Allow Duplicate Nodes** to permit shared nodes within the tree that you are adding. For trees that are being added, the value of this field defaults to the setting at the application level (see Registering Oracle Financials Cloud General Ledger Applications), but it can be overridden at the tree level. For existing trees, the value is read from the **Allow Shared Nodes** setting on the hierarchy set. See Inspecting a Hierarchy Set.
5. To continue adding tree codes and version names, click **Add**. Otherwise, click **Next** to return to **Add Segments**.
6. To continue adding segments, click **Add**. Otherwise, click **Next** to continue registration. See Adding, Modifying, or Removing Financial Categories.

Adding, Modifying, or Removing Financial Categories

Financial categories identify groups of accounts for reporting with Oracle Transactional Business Intelligence. By default, Oracle Financials Cloud General Ledger applications in Oracle Fusion Cloud Enterprise Data Management have the same set of financial categories as default Oracle Financials Cloud General Ledger applications. These values are stored in the FINANCIAL_CATEGORY lookup table in Oracle Financials Cloud. If you have modified these values in this lookup table for your Oracle Financials Cloud General Ledger application, you must make the same modifications to the categories for that application in Cloud EDM.

To add, modify, or remove financial categories:

1. Perform an action:
   - Add a financial category:
     a. Click **Add**.
     b. Enter a label and value for the financial category.
     c. Click **Next** to add the category.
   - Edit a financial category:
     a. Click the name of the financial category.
     b. Edit the label or value for the financial category.
     c. Click **Next** to save your changes.
   - Remove a financial category:
     a. Find the financial category that you want to remove.
     b. Click ⊖ .
2. When you have finished editing the financial categories, click **Next**.
   The Review the Registration Before Applying Changes page is displayed. Continue with Applying Registration Changes.

Applying Registration Changes

On the Review the Registration Before Applying Changes page, review the data. If the data is correct, click **Apply**.

> **Note:**
>
> If you need to change any of the data, click **Back** to navigate to the **Financial Categories** page, or from the left panel select one of the following to navigate to that registration step:
>
> - **Financial Setup**
> - **Segments**
> - **Financial Categories**

The application and its associated data chain elements are created:

- An application and default view are created to represent the external application.
- Dimensions, node types, and a list node set are created to represent segments and their associated value sets.
- Viewpoints, hierarchical node sets, and hierarchy sets are created to represent the trees and tree versions.

Predefined Properties for Oracle Financials Cloud General Ledger Applications

The following properties are predefined for Oracle Financials Cloud General Ledger applications. All Oracle Financials Cloud General Ledger properties use the FCGL namespace.

> **Caution:**
>
> Do not use inherited property values in any of the segment value properties, and avoid using Derived properties that contain positional logic (such as `ancestors`, `parent`, `children`, and `bottom`) when creating expressions to derive their default values. If your expression uses positional logic, use Derived and Stored properties instead.
>
> When you export data to the external Oracle Financials Cloud General Ledger application, segment values and their properties are exported from the bound *list* viewpoint, and relationship information is exported from the *hierarchy* viewpoints. Therefore, inherited properties and derived expressions using positional logic will result in different property values for nodes in the list and hierarchy viewpoints.

For more information, see Working with Properties.

Property	Data Type	Level	Description
Account Type	String	Node	Account type of account segment values **Note:** This property is required for nodes in the natural account segment.
Allow Budgeting	Boolean	Node	Specifies whether budget entry is allowed for a given segment value
Allow Posting	Boolean	Node	Specifies whether posting is allowed for a given segment value
Enabled	Boolean	Node	Specifies whether segment value is enabled
End Date	Date	Node	End Date for the segment value
Financial Category	String	Node	Identifies the financial category
Reconcile	Boolean	Node	Enables reconciliation
Start Date	Date	Node	Start Date for the segment value
Summary	Boolean	Node	Denotes summarization
Third Party Control Account	String	Node	Identifies the third-party control account
Tree Label	String	Relationship	Identifies tree labels for summary nodes

Importing Oracle Financials Cloud General Ledger Dimensions

Oracle Fusion Cloud Enterprise Data Management supports importing from flat files that contain data from the segments (which includes the value sets and trees) from your Oracle Financials Cloud General Ledger application. You cannot import directly from Oracle Financials Cloud General Ledger applications.

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- The data file for import must be a ZIP file that contains two comma-delimited (CSV) files. (The import into Cloud EDM is not case-sensitive).
  - `GlSegmentValuesInterface.csv`— contains all the nodes in the dimension, as well as the properties for them and any global or context-sensitive descriptive flexfields that were added to the segment.
  - `GlSegmentHierInterface.csv` — contains the parent/child structure for the individual trees (tree code /tree version name combinations) as well as tree labels.

  There are two methods for creating the import ZIP file:

  - To export General Ledger segment values and hierarchies from an existing Oracle Financials Cloud General Ledger application, follow the instructions in this article:

Exporting Fusion General Ledger segment (value set) values and hierarchies data in the FBDI format (Doc ID 2889994.1).

  - To manually generate the ZIP files: see Import Segment Values and Hierarchies in *File-Based Data Import for Oracle Financials Cloud* for the file formats. You can download the `ChartofAccountsSegmentValuesandHierarchiesImportTemplate.xlsm` template from this link, which enables you to enter the segment values and hierarchies that you want to import and then generate the CSV files in the correct format.

    > **Tip:**
    >
    > The `ChartofAccountsSegmentValuesandHierarchiesImportTemplate.xlsm` template formats dates as `YYYY/MM/DD`. This format may differ from the date formats used by other application types. For example, Universal applications date formats default to the regional settings for the user running the import.

- Value set codes, tree codes, and tree versions in the import file that are not registered in the Cloud EDM application will not be processed during import (the records will be skipped).
- When importing using the Replace or Merge modes, only the hierarchy sets that are used by trees that are included in the import hierarchy file are cleared before running the import. Trees not in the hierarchy file are not affected.

  > **Note:**
  >
  > When importing using the Reset mode, all trees are cleared as part of the import and only the trees in the import file are created.

- When you import with global or context-sensitive descriptive flexfields, warnings are displayed for the following:
  - The import file contains values in the columns designated for global or context-sensitive descriptive flexfields but the application registration does not.
  - The application registration contains either global or context-sensitive descriptive flexfields but the import file does not contain values in the columns designated for global or context-sensitive descriptive flexfields.

  If the import file contains the columns designated for global or context-sensitive descriptive flexfields (that is, the requisite number of commas in the comma-delimited file) but there are no values in those fields, the import process does not display a warning.

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. In the left column, select the dimension that you want to import data into.
3. In **File Name**, specify an import file name. The import file must be a ZIP file that contains the `GlSegmentValuesInterface.csv` and `GlSegmentHierInterface.csv` files.
4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.

5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*) or a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

Modifying Registered Oracle Financials Cloud General Ledger Applications

After you have registered an Oracle Financials Cloud General Ledger application, you can modify the registration to change some of the application settings. For example, you can add or remove a segment or tree, add or remove global or context sensitive descriptive flexfields, or change the financial categories. The application, dimensions, and data objects are updated based on your changes. The dimension bindings are updated to reflect the modifications you make.

> **Note:**
>
> If you remove a tree, it is unbound from the corresponding viewpoint, node set, and hierarchy set, but those data chain objects are not deleted. For example, if you remove the Corporate Account tree from the Account segment, the Corporate Account viewpoint, node set, and hierarchy set become unbound, but they are not deleted and they can be reused. The Corporate Account tree is no longer imported or exported from the application.
>
> If you remove a segment, the corresponding data chain objects for that segment and all of the trees in the segment are unbound from that segment, but not deleted. The removed segment is no longer imported or exported from the application. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for an Oracle Financials Cloud General Ledger application:

1. To modify the application settings, from **Applications**, find your application, click ⋮, and then select **Modify Registration**.
   - To modify the financial setup settings (for example, to allow multiple active tree versions or to change the maximum depth), see Registering Oracle Financials Cloud General Ledger Applications.
   - To add or remove global descriptive flexfields, see Adding Global Descriptive Flexfields
   - To change or add segments and trees or to add or remove context sensitive descriptive flexfields, see Registering Segments and Trees.
   - To change or add financial categories, see Adding, Modifying, or Removing Financial Categories.
2. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

Note:

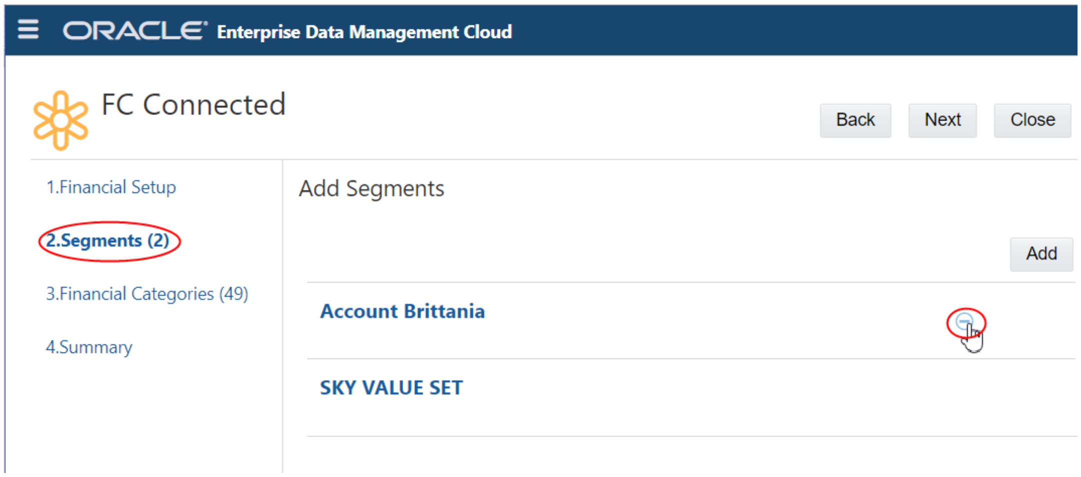
If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.

If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Removing Segments, Trees, or Financial Categories

To remove segments, trees, or financial categories from an application:

1. From **Applications**, find the application, click ⋮, and then select **Modify Registration**.
2. Navigate to the segment, tree, or financial category that you want to remove, and then click ⊖ .
   To navigate to a tree, click the segment name that contains the tree and then select the tree that you want to remove.



Binding Rules for Oracle Financials Cloud General Ledger Applications

Binding rules ensure that an Oracle Financials Cloud General Ledger application in Oracle Fusion Cloud Enterprise Data Management conforms to the requirements of the external application. The following table describes the binding rules for Oracle Financials Cloud General Ledger.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.

Object	Rule
Hierarchy Set	You cannot edit the node type in bound hierarchy sets.
Node Set	For bound node sets: • You cannot edit the node type in bound node sets. • For hierarchy node sets (trees), you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. • For list node sets (value sets), the node types are also checked. • For hierarchy node sets (trees), the hierarchy set is checked and then the node types are checked.
Dimension/Segment (Value Sets and Trees)	When you edit a viewpoint that is bound to a dimension, you can select a different viewpoint for the binding. The new viewpoint and all of the associated data objects in the data chain are checked for binding rules.

Note:

You can remove a segment (dimension) by deleting it from the application registration. The dimension will continue to be displayed in the application inspector with all of its bound viewpoints to the value set and trees, and you can create data chain objects using the dimension, but it cannot be used for importing or exporting data.

You can remove trees from the segment by deleting them from the application registration. The viewpoints for the trees are not deleted, but they become unbound and cannot be used for importing or exporting data. See Modifying Registered Oracle Financials Cloud General Ledger Applications.

Oracle Financials Cloud General Ledger Validations

Validations enforce data integrity. The following validations run for Oracle Financials Cloud General Ledger applications. They all have the Oracle Financials Cloud General Ledger source validation type.

In addition to these Oracle Financials Cloud General Ledger validations, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

Note:

For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

Validation Name	Description	Scope	Level
Duplicate (Shared Nodes)	A tree version contains a duplicate node at the leaf level but **Allow Duplicate Nodes** is not selected. **Note:** Duplicate nodes at the rollup level are never allowed in Oracle Financials Cloud General Ledger applications.	Viewpoint	Tree or Hierarchy Set
Equivalent Nodes	Nodes with a Summary Flag set to False are not in at least one location in all required related viewpoints. See Configuring Related Viewpoints. **Note:** When you have a list viewpoint configured with several related hierarchy viewpoints, you must run this validation against the list viewpoint in order to see if the node is missing in any of the related hierarchy viewpoints.	Related Viewpoints	Tree or Node Set
Maximum Depth Check	Node cannot have a level greater than the **Max Depth**. If the **Max Depth** is `-1` then the validation is not run.	Viewpoint	Tree or Hierarchy Set

Validation Name	Description	Scope	Level
Maximum Length Node Name Text/ Numeric	Node name exceeds the **Maximum Character Length**	Node	Value or Node
Minimum/ Maximum Node Name Text	Node name falls outside of the text range. **Note:** Numeric names are allowed when the **Value Type** is text as long as a numeric value is entered as either the minimum or maximum values (or both).	Node	Value or Node
Minimum/ Maximum Node Name Numeric	Node name falls outside of the numeric range. **Note:** Text names are not allowed when the **Value Type** is numeric.	Node	Value or Node
Multiple Root Nodes	Multiple root nodes exist but **Allow Multiple Root Nodes** is not selected.	Viewpoint	Tree or Node Set
Node Start / End Date	The End Date property is greater than the Start Date property for a node in a value set.	Node	Property
Numeric Node Name	- Node name contains characters other than 0-9 plus valid thousands and decimal separators and **Value Type** is Numeric. - Node name is not zero-filled up to the **Maximum Character Length** of the name. Zero fill is automatically enforced when the segment Value Type is numeric. See Registering Segments and Trees.	Node	Value or Node
Parent Summary Flag	The parent node must have the summary flag set to allow children.	Node	Tree or Hierarchy Set

Validation Name	Description	Scope	Level
Summary Flag and Allow Posting Properties	Summary and Allow Posting properties are both enabled. Only one can be enabled at a time. If a node has children, Summary must be enabled. If Allow Posting is enabled, the node cannot have children.	Node	Property
Summary Node with Missing Children	A node marked as a summary node does not have at least one child in a hierarchy viewpoint.	Node	Tree or Hierarchy Set
Tree Labels on Summary Node Only	A tree label exists on a node where the Summary Flag is not set to True.	Node	Property
Uppercase Node Name	Name contains lowercase characters and **Uppercase** is selected.	Node	Value or Node
Zero fill	Name contains fewer characters than the **Maximum Character Length** and **Value Type** is numeric or **Zero Fill** is selected.	Node	Value or Node

Invalid Characters, First Characters, and Values for Oracle Financials Cloud General Ledger

When you register an Oracle Financials Cloud General Ledger application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` property. For a list of these characters and values, see Essbase Character and Word Limitations in *Implementing Enterprise Structures and General Ledger*.

You can edit these values by editing the property parameters. For example, you can add additional invalid characters or invalid first characters, such as `&`, by editing the property parameters on the `Core.Name` property for your application or node type. See Editing Property Parameters.

Copying Bindings to Create Trees or Tree Versions

For Oracle Financials Cloud General Ledger applications only, you can copy an existing binding to create a new bound tree or tree version that uses an existing version as a starting point.

There are two ways to create new trees or tree versions in Oracle Financials Cloud General Ledger applications:

- If you want to start with an empty and unbound tree and tree version, you can modify the application registration and add a new tree or tree version. This will create a new blank tree or tree version that is not bound to an external application.
- If you want to create a new bound tree or tree version that contains hierarchical data from an existing tree or tree version, you can copy a binding of that existing tree or tree version. This will create a new bound tree or tree version that is bound to an external application.

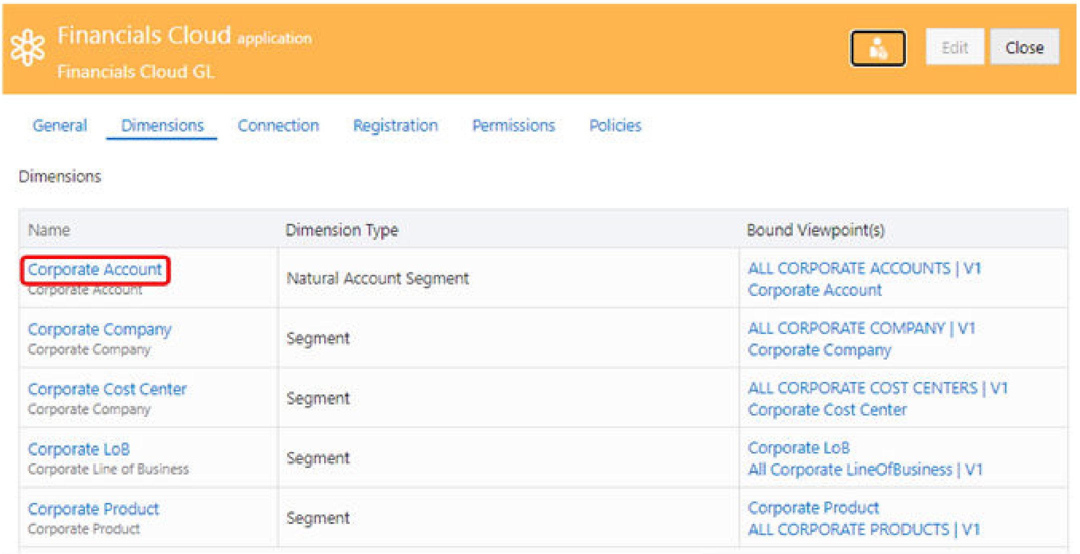
Considerations

- You can copy hierarchy-type bindings only; you cannot copy node-type bindings.

- When you copy a binding, the following data chain objects are copied and bound for that binding as well:
  - Hierarchy set (using the original node type)
  - Node set (using the new hierarchy set)
  - Viewpoint (using the new node set)
- When you copy a binding, the same validations are run as when you create a new tree version in application registration:.
  - The combination of tree code and tree version name must be unique.
  - The version start date must be before the version end date.
  - The version start and end dates must not overlap with another tree version in the same tree code.
- After you copy a binding and its associated data chain objects, if you modify the application registration the new hierarchy set and node set are displayed as bound tree and tree versions in the registration wizard.
- When you copy a binding, the viewpoint that gets created is added to the view where you copied the binding from.

Copying a Binding

1. From **Applications**, click the name of the application that contains the dimension with the binding that you want to copy. In this example, we have an Oracle Financials Cloud General Ledger application that contains a Corporate Account dimension with one tree version: `V1`, and we want to create a second bound tree version that uses `V1` as a starting point.
2. In the application inspector, click **Dimensions**.
3. Click the name of the dimension with the binding that you want to copy. In this example, we will select the `Corporate Account` dimension.



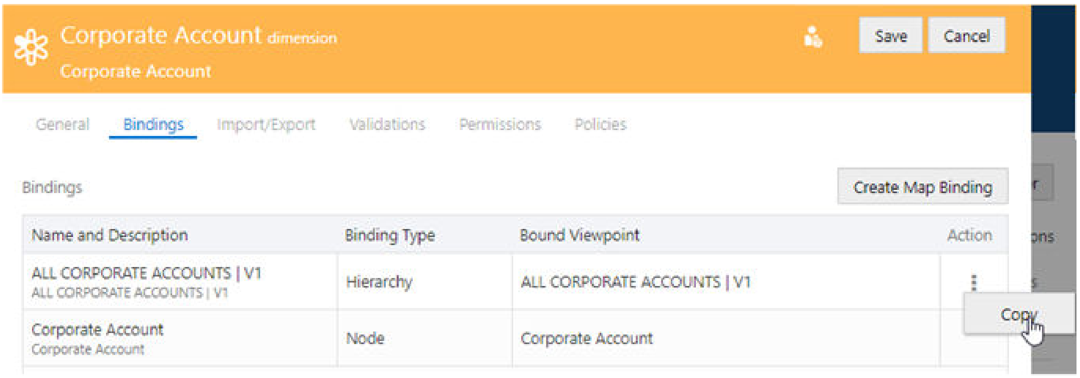
Dimensions

Name	Dimension Type	Bound Viewpoint(s)
Corporate Account Corporate Account	Natural Account Segment	ALL CORPORATE ACCOUNTS | V1 Corporate Account
Corporate Company Corporate Company	Segment	ALL CORPORATE COMPANY | V1 Corporate Company
Corporate Cost Center Corporate Company	Segment	ALL CORPORATE COST CENTERS | V1 Corporate Cost Center
Corporate LoB Corporate Line of Business	Segment	Corporate LoB All Corporate LineOfBusiness | V1
Corporate Product Corporate Product	Segment	Corporate Product ALL CORPORATE PRODUCTS | V1

.

4. From the dimension inspector, click **Bindings**.

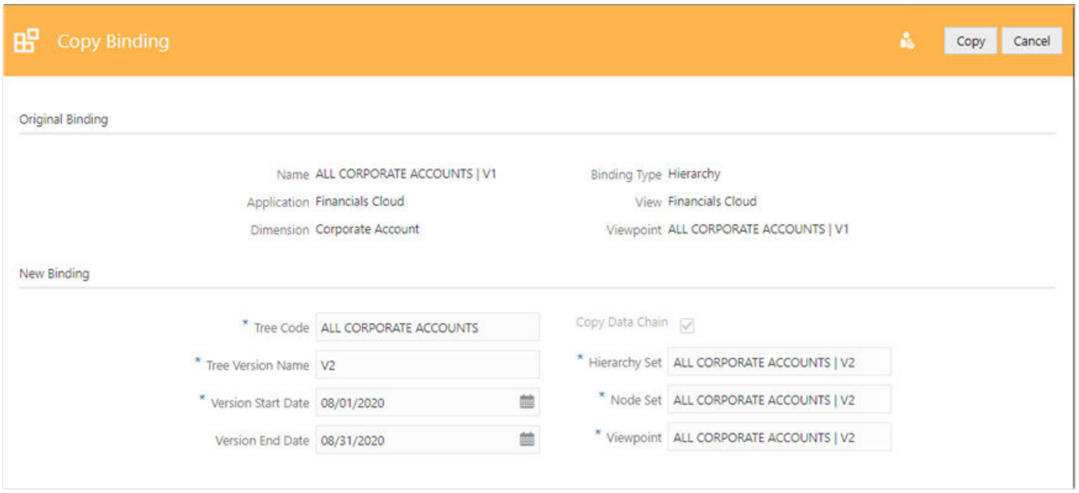
5. Click **Edit**, and then in the Actions column for the binding that you want to copy, click ⋮ and select **Copy**. In this example, you copy the `ALL CORPORATE ACCOUNTS| V1` binding.



Bindings

Name and Description	Binding Type	Bound Viewpoint	Action
ALL CORPORATE ACCOUNTS | V1 ALL CORPORATE ACCOUNTS | V1	Hierarchy	ALL CORPORATE ACCOUNTS | V1	⋮
Corporate Account Corporate Account	Node	Corporate Account	

6. In the **Copy Binding** dialog box, enter the tree code and tree version name, as well as the tree start and end dates for the new binding.
By default, the names of the new hierarchy set, node set, and viewpoint that are created for the binding are populated with the tree code and tree version name (in this example, `ALL CORPORATE ACCOUNTS| V2`).



Original Binding

Name	ALL CORPORATE ACCOUNTS | V1	Binding Type	Hierarchy
Application	Financials Cloud	View	Financials Cloud
Dimension	Corporate Account	Viewpoint	ALL CORPORATE ACCOUNTS | V1

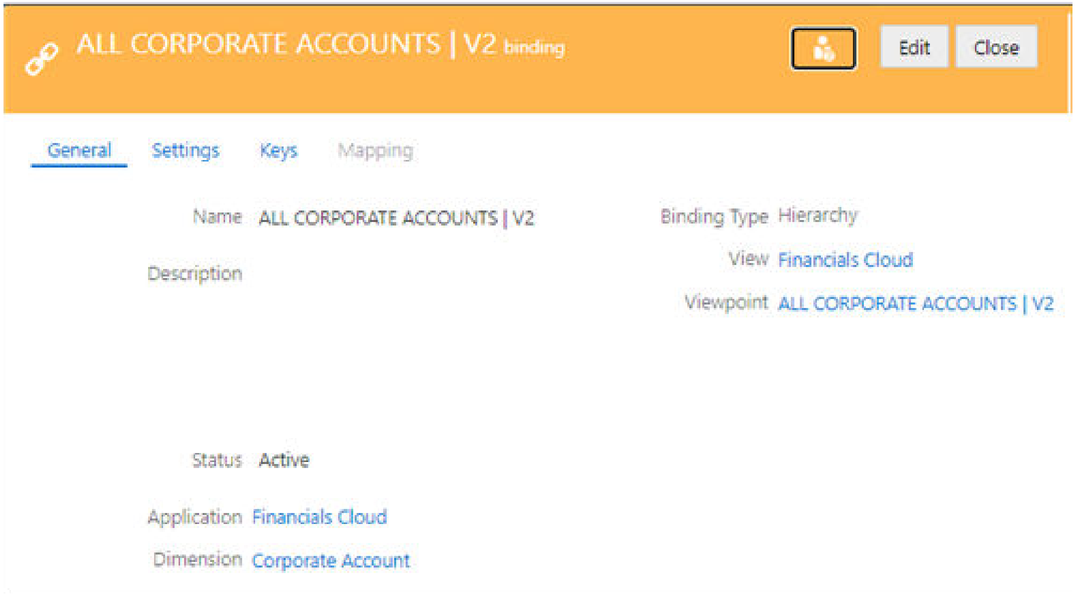
New Binding

* Tree Code	ALL CORPORATE ACCOUNTS	Copy Data Chain	☑
* Tree Version Name	V2	* Hierarchy Set	ALL CORPORATE ACCOUNTS | V2
* Version Start Date	08/01/2020	* Node Set	ALL CORPORATE ACCOUNTS | V2
Version End Date	08/31/2020	* Viewpoint	ALL CORPORATE ACCOUNTS | V2

7. **Optional**: Change the default names of the hierarchy set, node set, or viewpoint.

8. Click **Copy**.
The binding is copied and a new bound hierarchy set, node set, and viewpoint are created. The viewpoint is added to the view that you copied the binding from.

9. After the copy operation is completed, the new binding is displayed in the binding inspector. Review the binding settings, and then click **Close**.



Name ALL CORPORATE ACCOUNTS | V2

Description

Binding Type Hierarchy

View Financials Cloud

Viewpoint ALL CORPORATE ACCOUNTS | V2

Status Active

Application Financials Cloud

Dimension Corporate Account

Exporting Oracle Financials Cloud General Ledger Dimensions

You can export from one or more viewpoints that are bound to a dimension in an external Oracle Financials Cloud General Ledger application. You export to comma-delimited files, and you can specify connection information to upload the files to your Oracle Financials Cloud General Ledger server.

Nodes are exported starting with the top node of the node set. For example, a dimension can contain United States as a top node, with all of the states under it. However, if the node set's top node is Texas, only Texas and the nodes under it are exported. For tree viewpoints, nodes are exported starting with the top node of the hierarchy set.

Best Practice

It's a best practice to validate the viewpoints bound to the dimension prior to performing the export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension.
- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- Nodes and hierarchical relationships are exported in the order they are in Oracle Fusion Cloud Enterprise Data Management. If you want nodes and hierarchical relationships to be exported in alphanumeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.

> **Note:**
>
> To preserve the user-defined order of the members in the hierarchies when importing the file into Oracle Financials Cloud General Ledger, enable **Account Hierarchy User-Defined Sort Order Using File-Based Data Import** through the Functional Setup Manager (FSM) General Ledger opt-in features.
>
> For more information about preserving the user-defined sort order when importing account hierarchies, see Account Hierarchy User-Defined Sort Order Using File-Based Data Import in *Implementing Enterprise Structures and General Ledger*.

- Oracle Financials Cloud General Ledger exports dates in the `YYYY/MM/DD` format. This format may differ from the date formats used by other application types.
- When you export dimensions, a ZIP file is created that contains the following CSV (comma-delimited) files. (The import into Oracle Financials Cloud General Ledger is case-sensitive).
  - The `GlSegmentValuesInterface.csv` file contains all the nodes in the dimension, as well as the properties for them and any global or context-sensitive descriptive flexfields that were added to the segment.
  - The `GlSegmentHierInterface.csv` file contains the parent-child structure for the individual trees (combinations of tree codes and tree version names) as well as the tree labels.

    > **Note:**
    >
    > If the dimension that you are exporting has a list type node set and does not have any trees registered, the `GlSegmentHierInterface.csv` file is not generated.

- When you export dimensions, the Data Source column in the generated `GlSegmentValuesInterface.csv` file is automatically populated with the keyword `EDMCS`. This clears the values in all of the descriptive flexfield attribute columns for a value set being imported into Oracle Financials Cloud General Ledger if another value is not already present in those columns.

  > **Caution:**
  >
  > Ensure that all descriptive flexfields that are used by a value set have been registered for the Oracle Financials Cloud General Ledger application, and that data values have been defined for them in Cloud EDM to prevent clearing data unintentionally.

  For more details, see How Segment Value and Hierarchy Import Data Is Processed in *Implementing Enterprise Structures and General Ledger*.
- If you entered valid connection information to an Oracle Financials Cloud General Ledger server during registration, the ZIP file is uploaded to that server. You can import the file into your Oracle Financials Cloud General Ledger application. See Overview of External Data

Integration Services for Importing Data in *Oracle Financials Cloud Implementing Common Features for Financials and Project Portfolio Management*.

- For Oracle Financials Cloud General Ledger dimension exports to a connection only, you can configure export options to include additional post-export tasks that initiate the import and publishing processes in the external Financials Cloud environment and complete the synchronization of segment values and trees. See Oracle Financials Cloud General Ledger Export Options.

> **Caution:**
>
> After you export from Cloud EDM, note these restrictions when importing into your external Oracle Financials Cloud General Ledger application:
>
> - You cannot import a change on the Account Type of the natural account segment after it has been applied in the external General Ledger application.
> - For the Summary Flag in all segments:
>   - You can import a change from **Y** to **N**, but this may require restructuring of hierarchies where the segment value has children
>   - You cannot import a change from **N** to **Y**. This change must be made in the external General Ledger application.

> **Tip:**
>
> When exporting Oracle Financials Cloud General Ledger dimensions from Cloud EDM to an application connection using the REST API (see Exporting Dimensions Use Cases in *REST API for Oracle Enterprise Data Management Cloud Service*), the Document ID of the zip file transferred to Oracle Financials Cloud is returned in the result for the Cloud EDM export job.
>
> This document ID can then be passed to the ERP Integration Service API to import the zip file as segment values and hierarchies in Financials Cloud (see Create an inbound or outbound set of data in *REST API for Oracle Financials Cloud*). This eliminates an additional call to the Financials Cloud SOAP API to retrieve the Document ID for the EDM export file.

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. In the left column, select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File** or **Connection**, and then do one of the following:
   - If you selected **Connection**, select a connection to export to, specify an export file name, and then click **Export**.
   - If you selected **File**, an export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example FCGL_Account_20200123.csv, and can be edited before running the export. Your

browser settings define the download location. Accept or edit the default file name, and then click **Export**.

> **Note:**
>
> The **Connection** option in **Export Target** is available only if you entered connection information to an Oracle Financials Cloud General Ledger server during application registration.

5. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*) or a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

37

Working with E-Business Suite General Ledger Applications

You use Oracle Fusion Cloud Enterprise Data Management to manage charts of accounts segment values for value sets in Oracle E-Business Suite General Ledger. Users create new segment values, edit their properties, and maintain hierarchies using Cloud EDM.

This information can be:

1. Exported from E-Business Suite General Ledger

   Imported into Cloud EDM
2. Managed in Cloud EDM
3. Exported from Cloud EDM
4. Imported into E-Business Suite General Ledger

The following types of General Ledger data can be managed in Cloud EDM:

- Charts of accounts segment values
- Segment value properties

  > **Caution:**
  >
  > If you modify the predefined properties that get created by the registration system, do not use property inheritance or positional logic for deriving property values. See Predefined Properties for E-Business Suite General Ledger Applications.

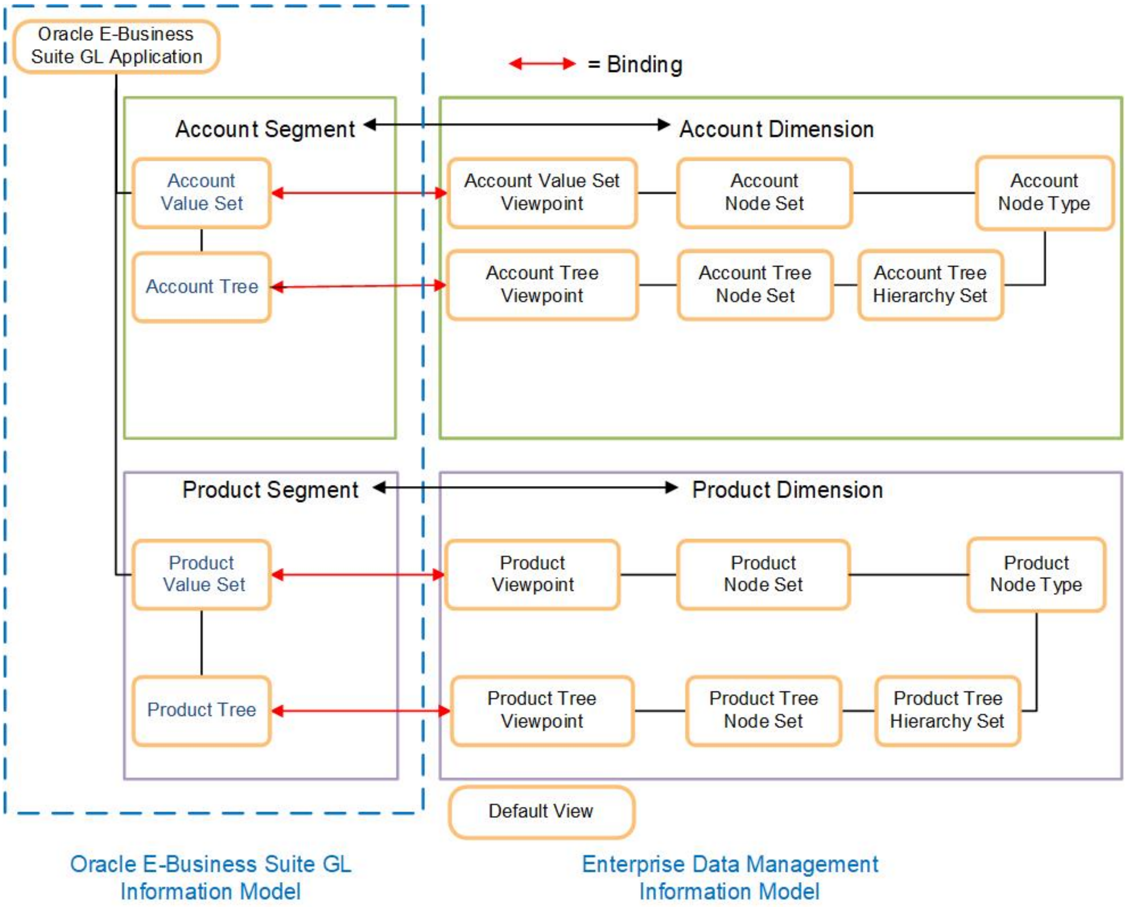
- Hierarchy relationships of segment values
- Global and context-sensitive descriptive flexfields

> **Note:**
>
> This integration between Cloud EDM and E-Business Suite General Ledger is available for E-Business Suite General Ledger Release 12.1.3 or higher.

Managing data for E-Business Suite General Ledger in Cloud EDM involves:

- Registering E-Business Suite General Ledger Applications
- Importing E-Business Suite General Ledger Dimensions
- Modifying Registered E-Business Suite General Ledger Applications
- Getting Started with Data Management
- Exporting E-Business Suite General Ledger Dimensions

You register E-Business Suite General Ledger applications by using a wizard that prompts you for information, such as active language and segment information.



When you register an E-Business Suite General Ledger application, these data chain objects are created for that application in Cloud EDM:

- **Default view**—enables you to manage all of the data chain objects for the application. See Understanding Views.
- For every segment value set in your application, these data chain objects are created:
  - **Dimension**—provide a way to categorize data values. See Understanding Dimensions.
  - **Viewpoint**—provides a subset of nodes for you to work with. The viewpoint points to the node set. See About Viewpoints.
  - **List-type node set**—defines the group of nodes available in a viewpoint. For segment value sets, a list node set is created that contains all of the available nodes without any hierarchical data. See Working with Node Sets.
  - **Node type**— displays a collection of an application's nodes that share a common business purpose. The node type has all of the application-specific properties. See Working with Node Types.

After you register an application, you can import data into the dimensions to populate the nodes, properties and hierarchy sets. Optionally, you can modify the registration. When you are done working on the data set, you can export the dimension back to the external application.

> **Note:**
>
> When the dimensions and data chain objects are created in Cloud EDM, they are bound to the segments and value sets in the E-Business Suite General Ledger application. These bindings ensure that the registered data chain objects conform to the external application requirements, and they prevent users from making changes that would prevent an application's data from being imported or exported. See Understanding Binding Rules.

Registering E-Business Suite General Ledger Applications

When you register an Oracle E-Business Suite General Ledger application into Oracle Fusion Cloud Enterprise Data Management, an application and default view are created to represent the external application. Dimensions, node types, and a list node set are created to represent segments and their associated value sets. Viewpoints, hierarchical node sets, and hierarchy sets are created to represent the hierarchies.

Before you begin:

- Ensure that you have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

  > **Note:**
  >
  > The Service Administrator assigns roles, see Understanding Application Roles and Permissions.

- Decide which segments and associated value sets in your chart of accounts you want to register. See Working with E-Business Suite General Ledger Applications.

To register an application:

1. From **Applications**, click **Register**.
2. Select **E-Business Suite General Ledger** as the application type.
3. Enter an application **Name** and, optionally, a **Description**.

   > **Note:**
   >
   > The application type, name, and description are displayed in the Application list.

4. Click **Create**.
5. In **Active Languages**, enter the languages that have an Active status in the E-Business Suite General Ledger application and then click **Next**.

> **Note:**
>
> As you type the name of a language, a list of supported languages that match your text entry are displayed. Select the language to add it to the Active Languages list.

6. In **Add Segments**, click **Add**.
7. Configure segment parameters and then click **Next**.
   - Enter the **Segment/Value Set Code** for the segment
   - Enter the **Segment Tree Label** This label is added to the end of the viewpoint and node set names for the hierarchy.
   - Enter **Segment Value Set Description**
   - Select **Natural Account Segment** if this segment is used as the natural account in a chart of accounts.
   - Select **Allow Duplicate Nodes** if duplicate nodes are allowed in the segment.
   - Enter the **Max Length** for the values in the value set.
   - Enter the **External Segment/Value Set Code** for the segment.
8. On the Review the Registration Before Applying Changes page, review the data. If the data is correct, click **Apply**.

The application and its associated data chain elements are created:

- An application and default view are created to represent the external application.
- Dimensions, node types, and a list node set are created to represent segments and their associated value sets.
- Viewpoints, hierarchical node sets, and hierarchy sets are created.

Adding Global Descriptive Flexfields

You can define up to 50 descriptive flexfields during registration, and then import and export them to Oracle E-Business Suite General Ledger.

About Descriptive Flexfields

Descriptive flexfields are user-defined fields that enable you to capture additional attributes about segment values. For example, a Product segment could contain descriptive flexfields for Color and Size. There are two types of descriptive flexfields:

- Global descriptive flexfields are applied to all segments in an application, and both the flexfield and its associated property must be unique. You cannot reuse either the name or the property of a global descriptive flexfield in either another global or a context-sensitive flexfield.
- Context-sensitive descriptive flexfields are applied only to specific segments, and both the flexfield and its associated property can be reused in other segments.

To illustrate the usage of global versus context-sensitive descriptive flexfields, suppose you have three segments: Account, Entity, and Product. If you have an attribute that applies to all three segments, you can create a global descriptive flexfield and it will be added to all of your segments. However, if you have an attribute that applies to only two of the segments, you can create a context-sensitive flexfield and add it only to the two segments to which it applies.

For more information about descriptive flexfields, see Overview of Flexfields in *Oracle Applications Cloud Configuring and Extending Applications*.

To add a global descriptive flexfield:

1. In Add Global Descriptive Flexfields, click **Add**.
2. Enter the **Flexfield Name**, and then specify whether or not the flexfield is required. The name must be unique for global descriptive flexfields.
3. In **Column**, click the drop down menu and select the attribute location (1-50) for the flexfield. This field determines the location of the value for this flexfield in the import and export files.
   When you select an attribute location for either a global or a context-sensitive descriptive flexfield, that location is no longer available to be selected by another flexfield. For example, if you select the `Attribute 3` location for a global descriptive flexfield, when you add another global or context-sensitive descriptive flexfield, `Attribute 3` is no longer displayed in the drop down menu.
4. In **Property**, click the drop down menu and perform an action:
   - To create a new property for the flexfield, select **(Add New Property)** and then click **Next**. Continue to Adding a New Property for a Flexfield.
   - Select an existing property for the flexfield and then click **Next** to return to Add Descriptive Flexfields. When you select an existing property for a global descriptive flexfield, that property is removed from the drop down list and you can no longer select it for other global or context-sensitive descriptive flexfields.

Adding a New Property for a Flexfield

1. By default, the property name is "DFF {Flexfield Name}". Optionally, change the name or enter a description for the property, and then click **Next**.
2. Specify the parameters for the property:

Parameter	Description
String Case	Specify whether the property should be in `Upper Case`, `Lower Case`, or `Mixed Case`.
Invalid Characters	Specify the invalid characters for the property. Click in the **Invalid Characters** field and then select the invalid characters from the provided list.
Minimum Length	Enter the minimum text length for the property. The default value is 0.
Maximum Length	Enter the maximum text length for the property. The default value is 240, and you cannot increase the maximum past 240.
Use Allowed Values List	Select to restrict entry in the property field to only those values listed in **Allowed Values**.
Include Blank Entry	Select to allow this property to be left blank.
Allowed Values	Enter the values that you want to be displayed for this property. Enter a value and then hit `Tab` to enter the next value. Select **Use Allowed Values List** to restrict user entry only to the values in this field.
Default Value	Enter a default value for the property.

Parameter	Description
Inheritance	Select whether or not to use positional inheritance for this property. See Inheriting Properties.

3. Click **Next** to return to Add Global Descriptive Flexfields. The global descriptive flexfield that you added is displayed with a default name of the attribute location and the flexfield name that you entered, for example, "Attribute 1 - Color".
4. Click **Add** to add another flexfield, or click **Next** to continue the registration.

Registering Segments

Use these steps to register segments for Oracle E-Business Suite General Ledger applications.

To register segments:

1. In **Add Segments**, click **Add**.
2. Enter the **Segment/Value Set Code** for the segment, the **Segment Hierarchy Label**, and optionally, the **Segment Value Set Code Description**.
3. Select the check box next to **Natural Account Segment** if this segment is used as the natural account in a chart of accounts.
4. Select **Allow Duplicate Nodes** if allowing shared nodes.
5. Enter the **Maximum Character Length** for the values in the value set.
6. Enter the **External Segment/Value Set Code** and click **Next**.
7. For each segment added, you have the option of defining context sensitive descriptive flexfields for the segment.

Adding Context Sensitive Descriptive Flexfields

Context-sensitive descriptive flexfields enable you to capture additional attributes about segment values for specific segments only. The procedure for adding a context-sensitive descriptive flexfield to a segment is similar to adding a global descriptive flexfield, with these differences:

- Context-sensitive descriptive flexfields can be reused for more than one segment. If a context-sensitive descriptive flexfield applies to more than one segment, you must add it to each segment to which it applies.
- The properties associated with context-sensitive flexfields can also be reused. Unlike global descriptive flexfields, when you select a property to associate with a context-sensitive flexfield, that value does not get removed from the **Property** drop down menu and you can select it again for a different context-sensitive flexfield.

See Adding Global Descriptive Flexfields

Applying Registration Changes

On the Review the Registration Before Applying Changes page, review the data. If the data is correct, click **Apply**.

Note:

If you need to change any of the data, click **Back** to navigate to the **Segments** page, or from the left panel select one of the following to navigate to that registration step:

- **Financial Setup**
- **Global Descriptive Flexfield**
- **Segments**

The application and its associated data chain elements are created:

- An application and default view are created to represent the external application.
- Dimensions, node types, and a list node set are created to represent segments and their associated value sets.
- Viewpoints, hierarchical node sets, and hierarchy sets are created to represent the segments.

Predefined Properties for E-Business Suite General Ledger Applications

The following properties are predefined for Oracle E-Business Suite General Ledger. All E-Business Suite General Ledger properties use the EBSGL namespace.

Caution:

Do not use inherited property values in any of the segment value properties, and avoid using Derived properties that contain positional logic (such as `ancestors`, `parent`, `children`, and `bottom`) when creating expressions to derive their default values. If your expression uses positional logic, use Derived and Stored properties instead.

When you export data to the external E-Business Suite General Ledger application, segment values and their properties are exported from the bound *list* viewpoint, and relationship information is exported from the *hierarchy* viewpoints. Therefore, inherited properties and derived expressions using positional logic will result in different property values for nodes in the list and hierarchy viewpoints.

For more information, see Working with Properties.

Property	Data Type	Level	Description
Account Type	String	Node	Account type of account segment values **Note**: This property is required for nodes in the natural account segment.

Property	Data Type	Level	Description
Allow Budgeting	Boolean	Node	Specifies whether budget entry is allowed for a given segment value
Allow Posting	Boolean	Node	Specifies whether posting is allowed for a given segment value
Enabled	Boolean	Node	Specifies whether segment value is enabled
End Date	Date	Node	End Date for the segment value
Reconcile	Boolean	Node	Enables reconciliation
Roll Up Group	String	Node	Identifies the roll up group
Start Date	Date	Node	Start Date for the segment value
Summary	Boolean	Node	Denotes summarization
Third Party Control Account	String	Node	Identifies the third-party control account

Importing E-Business Suite General Ledger Dimensions

Oracle Fusion Cloud Enterprise Data Management supports importing from flat files that contain data from the segments (which includes the value sets and hierarchies) from your Oracle E-Business Suite General Ledger application. You cannot import directly from E-Business Suite General Ledger applications.

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- The data file for import must be a ZIP file that contains two comma-delimited (CSV) files. (The import into Cloud EDM is not case-sensitive). The hierarchy file is optional.
  - `GL_HIERARCHY_INTERFACE.CSV`— contains the parent/child structure for the individual hierarchies.
  - `GL_SEGVALUES_INTERFACE.CSV` — contains all the nodes in the dimension, as well as the properties for them.

  These files contain header records. A way to get these files is to register an application and export it to get the zip file with the header records in place. You can then update that file and save the CSV files in the correct format. For more information see Import and Export File Format.
- Value set codes and hierarchies in the import file that are not registered in the Cloud EDM application will not be processed during import (the records will be skipped).

To import a dimension:

1. From **Applications**, scroll to your application, then click ⋮, and then select **Import**.
2. In the left column, select the dimension that you want to import data into.
3. In **File Name**, specify an import file name. The import file must be a ZIP file that contains the `GL_HIERARCHY_INTERFACE.CSV` and `GL_SEGVALUES_INTERFACE.CSV` files.

   > **Note:**
   >
   > The hierarchy file is optional.

4. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes.
5. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
6. Click **Import**.

You can also import dimensions using EPM Automate (see importDimension in *Working with EPM Automate*) or a REST API (see Importing Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

Import and Export File Format

When you import and export data forOracle E-Business Suite General Ledger dimensions, you must use a comma-delimited (`.csv`) file.

Considerations

- The import file's data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a child node's row precedes its parent node's row, an error occurs when the file is imported.
- The import ignores duplicate rows.
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)

The following table describes the comma-delimited file's format in relation to E-Business Suite General Ledger staging tables.

Staging Table Column	File Header	File Content
VALUE_SET_CODE	VALUE_SET_CODE	Constant - Set to the Segment Value Set Code
VALUE	VALUE	Core.Name
STATUS	N/A	N/A Not included in export file
START_DATE_ACTIVE	START_DATE_ACTIVE	EBSGL.Start Date
END_DATE_ACTIVE	END_DATE_ACTIVE	EBSGL.End Date
SUMMARY_FLAG	SUMMARY_FLAG	EBSGL.Summary
SORT_ORDER	SORT_ORDER	CoreStats.Level

Staging Table Column	File Header	File Content
ENABLED_FLAG	ENABLED_FLAG	EBSGL.Enabled
ACCOUNT_TYPE	ACCOUNT_TYPE	EBSGL.Account Type
ALLOW_POSTING	ALLOW_POSTING_FLAG	EBSGL.Allow Posting
ALLOW_BUDGETING	ALLOW_BUDGETING_FLAG	EBSGL.Allow Budgeting
THIRD_PARTY_CTRL_ACCT	THIRD_PARTY_CTRL_ACCOUNT	EBSGL.Third Party Control Account
RECONCILE	RECONCILE_FLAG	EBSGL.Reconcile
FINANCIAL_CATEGORY	FINANCIAL_CATEGORY	EBSGL.Roll Up Group
DESC_AR	DESC_US	Either the property for the Description like EBSGL.Description US or a Constant Value of Blank If used as an active language then a property will be used to hold the value otherwise it is just a blank column.
DESC_CS	DESC_AR	
DESC_D	DESC_CS	
DESC_DK	DESC_D	
DESC_E	DESC_DK	
DESC_EL	DESC_E	
DESC_ESA	DESC_EL	
DESC_F	DESC_ESA	
DESC_FRC	DESC_F	
DESC_HR	DESC_FRC	
DESC_HU	DESC_HR	
DESC_I	DESC_HU	
DESC_IS	DESC_I	
DESC_IW	DESC_IS	
DESC_JA	DESC_IW	
DESC_KO	DESC_JA	
DESC_NL	DESC_KO	
DESC_LT	DESC_NL	
DESC_PL	DESC_LT	
DESC_PT	DESC_PL	
DESC_PTB	DESC_PT	
DESC_N	DESC_PTB	
DESC_RO	DESC_N	
DESC_RU	DESC_RO	
DESC_S	DESC_RU	
DESC_SF	DESC_S	
DESC_SK	DESC_SF	
DESC_SL	DESC_SK	

Staging Table Column	File Header	File Content
DESC_TH	DESC_SL	
DESC_TR	DESC_TH	
DESC_US	DESC_TR	
DESC_ZHS	DESC_ZHS	
DESC_ZHT	DESC_ZHT	

Modifying Registered E-Business Suite General Ledger Applications

After you have registered an Oracle E-Business Suite General Ledger application, you can modify the registration to change some of the application settings. For example, you can add or remove a segment or change the financial categories. The application, dimensions, and data objects are updated based on your changes. The dimension bindings are updated to reflect the modifications you make.

> **Note:**
>
> If you remove a segment, the corresponding data chain objects for that segment and all of the hierarchies in the segment are unbound from that segment, but not deleted. The removed segment is no longer imported or exported from the application. See Understanding Modifying Applications.

Before you begin:

To modify an application, you need *Owner* or *Metadata Manager* permission on the application.

To modify the registration for an E-Business Suite General Ledger application:

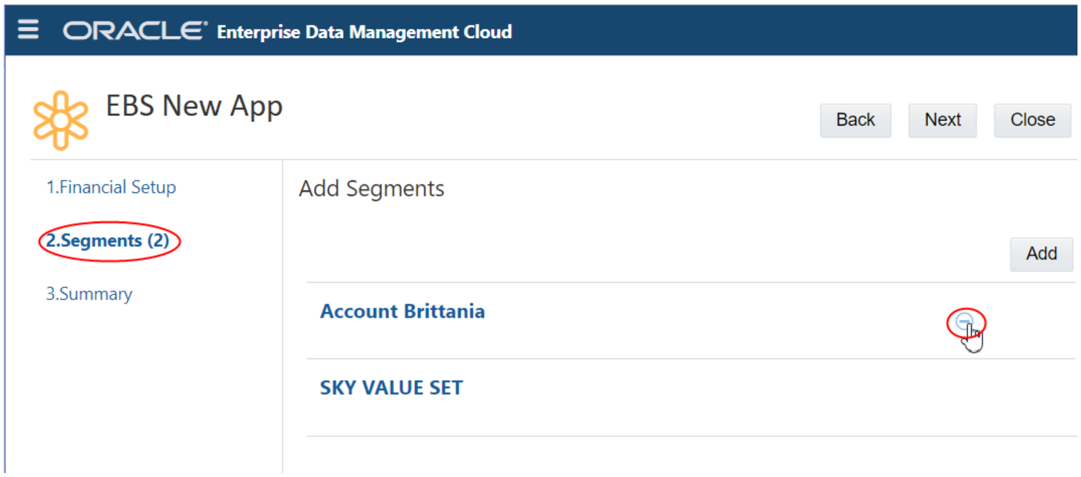
1. To modify the application settings, from **Applications**, find your application, click ⋮, and then select **Modify Registration**.
2. To change or add segments, see Registering E-Business Suite General Ledger Applications.
3. Click **Apply** when you are done.
   The data chain objects are updated with your modifications.

> **Note:**
>
> If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
>
> If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Removing Segments

To remove segments from an application:

1. From **Applications**, find the application, click ⋮, and then select **Modify Registration**.
2. Navigate to the segment that you want to remove, and then click ⊖ .



Binding Rules for E-Business Suite General Ledger Applications

Binding rules ensure that an Oracle E-Business Suite General Ledger application in Oracle Fusion Cloud Enterprise Data Management conforms to the requirements of the external application. The following table describes the binding rules for E-Business Suite General Ledger.

Object	Rule
Node Type	You cannot remove any required binding properties from a bound node type.
Hierarchy Set	You cannot edit the node type in bound hierarchy sets.

Object	Rule
Node Set	For bound node sets: • You cannot edit the node type in bound node sets. • For hierarchy node sets (hierarchies), you must select a hierarchy set that passes its own binding rules and any binding rules up the data chain.
Viewpoint	For bound viewpoints, you must select a node set that passes its own binding rules and any binding rules up the data chain. • For list node sets (value sets), the node types are also checked. • For hierarchy node sets (hierarchies), the hierarchy set is checked and then the node types are checked.
Dimension/Segment (Value Sets)	When you edit a viewpoint that is bound to a dimension, you can select a different viewpoint for the binding. The new viewpoint and all of the associated data objects in the data chain are checked for binding rules.

E-Business Suite General Ledger Validations

Validations enforce data integrity. The following validations run for Oracle E-Business Suite General Ledger applications. They all have the Oracle Financials Cloud General Ledger source validation type.

In addition to these E-Business Suite General Ledger validations, system and generic validations are run on nodes, hierarchies, viewpoints, and properties in the application. See System Validations and Generic Predefined Validations.

> **Note:**
>
> For validations with a scope of Viewpoint in the table below, for hierarchy-set bound viewpoints the validation is run for the bound viewpoint.

Validation Name	Description	Scope	Level
Maximum Length Node Name Text/ Numeric	Node name exceeds the **Maximum Character Length**	Node	Value or Node
Node Start/End Date	The Start Date property is greater than the End Date property for a node in a value set.	Node	Property
Parent Summary Flag	The parent node must have the summary flag set to allow children.	Node	Tree or hierarchy set

Validation Name	Description	Scope	Level
Summary Flag and Allow Posting	Summary and Allow Posting properties are both enabled. Only one can be enabled at a time. If a node has children, Summary must be enabled. If Allow Posting is enabled, the node cannot have children.	Node	Property
Summary Node with Missing Children	A node with Summary Flag enabled must have children.	Node	Tree or Hierarchy Set
Uppercase Node Name	Name contains lowercase characters and **Uppercase** is selected.	Node	Value or Node

Invalid Characters, First Characters, and Values for E-Business Suite General Ledger

When you register a E-Business Suite General Ledger application, validations for invalid characters, first characters, and values are automatically configured for the `Core.Name` property.

You can edit these values by editing the property parameters. See Editing Property Parameters.

Exporting E-Business Suite General Ledger Dimensions

You can export from one or more viewpoints that are bound to a dimension in an external Oracle E-Business Suite General Ledger application. You export to comma-delimited files. The files are designed to be loaded into E-Business Suite General Ledger staging tables. You must create your own process to do this.

Best Practice

It's a best practice to validate the viewpoints bound to the dimension prior to performing the export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

Before you begin:

- To export, you need a minimum of *Data Manager* permission on the application or dimension.
- Nodes and hierarchical relationships are exported in the order they are in Oracle Fusion Cloud Enterprise Data Management. If you want nodes and hierarchical relationships to be exported in alphanumeric order, you can open the dimension's hierarchy set and clear **Use Custom Order**.
- You can edit the binding keys for a dimension to change the order of the export columns or to specify the direction of each column in the export file. See Editing Binding Keys.

- You can export a defined keyword instead of a null or blank value. See Managing Export Options.
- When you export dimensions, a ZIP file is created that contains the following CSV (comma-delimited) files. (The import into E-Business Suite General Ledger is case-sensitive).
  - `GL_HIERARCHY_INTERFACE.CSV` — contains the parent-child structure for the individual hierarchies.
  - `GL_SEGVALUES_INTERFACE.CSV` — contains all the nodes in the dimension as well as the properties for them.

To export a dimension:

1. From **Applications**, find your application, then click ⋮, and then select **Export**.
2. In the left column, select the dimension to export.
3. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
4. From **Export Target**, select **File**. An export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example EBSGL_Account_20200123.csv, and can be edited before running the export. Your browser settings define the download location.
5. Click **Export**.

   The output of this file needs to be loaded into the E-Business Suite General Ledger staging tables.
6. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

You can also export dimensions using EPM Automate (see exportDimension in *Working with EPM Automate*) or a REST API (see Exporting Dimensions in *REST API for Oracle Enterprise Data Management Cloud Service*).

38 Working with Universal Applications

Universal applications let you manage data for external applications that do not have a corresponding application type in Oracle Fusion Cloud Enterprise Data Management. You can use Universal applications to build Cloud EDM solutions that suit any business need.

Universal applications provide flexibility because they do not enforce conformity to the requirements of an external application. A Universal application does not include binding rules or validations other than those that apply to all application types. In addition, there are no restrictions on properties that you can add or remove. For example, you can use Universal applications to manage enterprise data for a human resources external application.

Best Practice

Use the application type that matches your external application. The application type enforces conformity to your external application and requires minimal configuration. See the list of application types Understanding Application Types. If your application type is not in this list use the Universal application type.

Process Flow

The basic flow for creating a Universal application is described below:

Task	See
1. Register a new Universal application.	Registering a Universal Application
2. Add a dimension to the application.	Adding, Removing, or Modifying a User Defined Dimension
3. Add node types to the dimension.	Adding or Modifying a Node Type for a User Defined Dimension
4. Add or create properties for each node type that you added to the dimension.	Adding an Existing Property to a Node Type for a User Defined Dimension Creating a Custom Property
5. Configure import and export settings for the dimension.	User Defined Dimension Import and Export Settings
6. Repeat steps 2-5 for each dimension that you need to add to the application.	NA

For more information, see:

- Importing a User Defined Dimension
- Binding Rules for Universal Applications
- Validations for Universal Applications
- Modifying a Universal Application
- Exporting a User Defined Dimension

Registering a Universal Application

When you register a Universal application, you create its dimensions. For each dimension, you specify node types, properties, and settings for importing and exporting.

Videos

Your Goal	Watch This Video
Learn about registering Universal applications.	Registering Universal Applications

For more information, see:

- Understanding Registering Applications
- Working with Universal Applications

To register an application, perform the following steps:

1. Click **Applications**.
2. Click **Register**.

   The Choose an Application Type page displays.
3. Click **Universal**.
4. Enter the application **Name**, and optionally a **Description**
5. Use the drop down menu to specify the **External System Type**. If your external system is not displayed in the list, scroll down to select **Other**, and then enter an **External System Type Label**.

   The External System Type is displayed on the Application Summary page to help differentiate multiple Universal applications.
6. Click **Next**.

   The Application Summary page displays.
7. Click **Create**.

   The **Add the Dimensions** page displays.
8. For each dimension you need to add or modify, perform the steps in Adding, Removing, or Modifying a User Defined Dimension.
9. To remove a dimension, hover the cursor over the row containing the dimension, then click ⊖ .

   > **Note:**
   >
   > When you remove a dimension, its data objects and properties are unbound but are not deleted. This preserves the data contained by the data objects and properties.
10. After you finish adding dimensions, click **Next**.

    The Review the Registration Before Applying Changes page displays.

11. Review the data. If the data is correct, click **Apply**.

> **Note:**
>
> If you need to change any of the data, click **Back**, then select and modify the dimension to be changed.

When you click the **Apply** button, Oracle Fusion Cloud Enterprise Data Management creates the default view for the application and the data chain objects bound to the dimensions.

Adding, Removing, or Modifying a User Defined Dimension

When you configure a dimension for a Universal application, you specify settings such as whether nodes form a list or a hierarchy and whether the dimension allows shared nodes. You also add node types, define properties for node types, and specify settings for importing and exporting data.

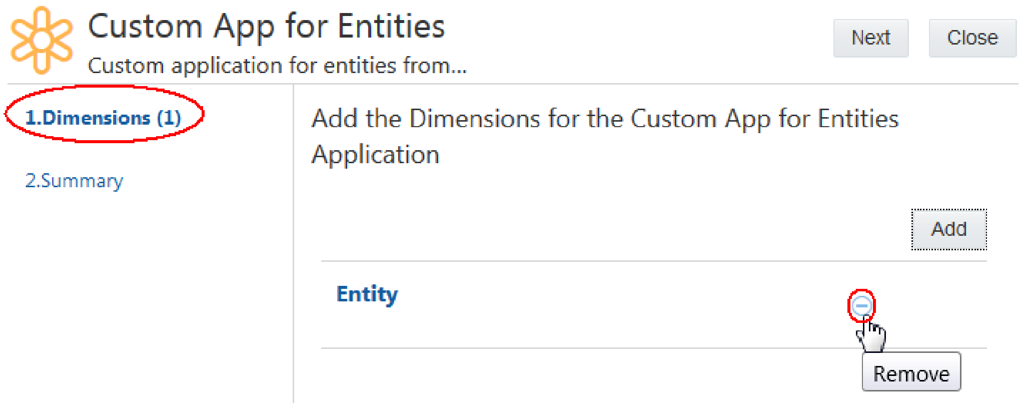
To add, remove, or modify a user-defined dimension, you must be on the Add the Dimensions page. The following topics describe the ways to access this page:

- Registering a Universal application, see Registering a Universal Application.
- Modifying a Universal application, see Modifying a Universal Application.

If you want to change the viewpoint bound to a dimension, see Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications.

Perform the following steps to add, remove, or modify a dimension in a Universal application:

1. From the Add the Dimensions page, perform one of the following steps:
   - To add a new dimension click **Add**.
   - To modify an existing dimension click its link.
   - To remove a dimension click ⊖ .


Custom App for Entities
Custom application for entities from...
Next
Close
1.Dimensions (1)
2.Summary
Add the Dimensions for the Custom App for Entities Application
Add
Entity
Remove

For add or modify, the **Dimension Basic Info** page displays.

2. Specify the **Dimension Name**, and optionally a **Dimension Description**.
3. From **Dimension Type** specify whether the nodes are a hierarchy or list.

4. (Optional) In **External Dimension Name**, enter the name the external application uses for the dimension, segment value set, or other data domain.
5. (Optional) Select **Allow Shared Nodes** if nodes can have multiple parent nodes in a hierarchy, see Understanding Shared Nodes.
6. Click **Next**.

   The Node Types for Dimension page displays.
7. For each node type you need to add or modify, perform the steps in Adding or Modifying a Node Type for a User Defined Dimension.
8. To remove a node type, hover the cursor over the row containing the node type, and then click ⊖ .

   > **Note:**
   >
   > When you remove a node type, the node type and its related data objects are unbound but are not deleted. This preserves the data contained by the node type.
9. After you are done adding node types to the dimension, click **Next**.

   The Import and Export for Dimension page displays.
10. (Optional) Configure import and export settings for the dimension, see User Defined Dimension Import and Export Settings.
11. Click **Next**.

    The Add the Dimensions page displays, from which you can add or modify another dimension or finish registering the application.

User Defined Dimension Import and Export Settings

When you add or modify a dimension, you are prompted to configure how the dimension will be imported and exported. The following table describes the settings you can configure:

Setting	Description
Default Node Type	The node type that the import process assigns to nodes if the import file does not have a column for node types.
Default Parent Node Type	The node type that the import process assigns to parent nodes if the import file does not have a column for parent node types.
Top Node Indicator	A value in the Parent column that indicates the node is the top node of a hierarchy. This typically is the value the external application uses to represent a top node. If an empty field indicates a top node, do not specify a value.
Use Node Type Column	Specifies that import and export files can have a column for node types.
Node Type Column Header	The column header for the column that specifies the node type of bottom nodes. The column header is used only if the **Use Node Type Column** check box is selected.
Use Parent Node Type Column	Specifies that import and export files can have a column for parent node types.

Setting	Description
Parent Node Type Column Header	The column header for the column that specifies the node type of parent nodes. The column header is used only if the **Use Parent Node Type Column** check box is selected.

For an example, see Import and Export File Format for User Defined Dimensions.

Adding or Modifying a Node Type for a User Defined Dimension

When you configure a node type in a Universal application, you specify settings such as whether the node type has a uniqueness qualifier. You can also add existing properties and create custom properties.

To add or modify a node type for a user-defined dimension, you must be on the Node Types for Dimension page. The following list describes the ways to access this page:

- Add or modify a dimension, see Adding, Removing, or Modifying a User Defined Dimension.
- Modify an application, see Modifying a Universal Application.

To add a node type or modify an existing node type, take the following steps:

1. From the Node Types for Dimension page, perform one of the following steps:
   - Add a new node type by clicking **Add**.
   - Modify an existing node type by clicking its link.

   The Node Types for Dimension page displays.
2. Specify the **Node Type Name** and optionally a **Node Type Description**.
3. (Optional) Enter a **Node Type Qualifier** and then select the **Node Type Qualifier Position**, see Working with Node Type Qualifiers
4. Click **Next**.

   A page for adding properties displays.
5. Add or modify node type properties by performing the steps in the following topics:
   - To create a new property, see Creating a Custom Property.
   - To add a property that has been defined for another node type or is in the `Core` or `CoreStats` namespace, see Adding an Existing Property to a Node Type for a User Defined Dimension.
   - To modify an existing property, see Modifying a Property for a Node Type in a User Defined Dimension.
6. To change the order of the export file columns for the properties, you must edit the binding keys on the dimension. See Editing Binding Keys.
7. To exclude data for a property from imports and exports, hover the cursor over the row containing the property, then click ✕ .

> **Note:**
>
> Although the data contained by the property will not be imported and exported, the property is not removed from the node type. This preserves the property data.

8. Click **Next**.

   The Node Types for Dimension page displays. From this page you can add or modify a node type or finish configuring the dimension.

Adding an Existing Property to a Node Type for a User Defined Dimension

To add an existing property, you must be on the Properties for Dimension Node Type page. You access this page when you perform any of the following tasks:

- Add or modify a node type, see Adding or Modifying a Node Type for a User Defined Dimension.
- After you add an existing property, create a custom property, or modify a property.

Existing properties can either be system-defined properties such as `Core.Description` or properties defined by registered applications. For example, if a Universal application defined a `Custom.Account Type` property, the property can then be used by other node types.

> **Tip:**
>
> Existing properties are listed on the Properties page, see Working with Properties.

To add an existing property to a node type, take the following steps:

1. Click **Select**.
2. Select the property, and then click **OK**.

   The Property for Dimension Node Type page displays.
3. If the page contains a **Property App Override Parameters** section, configure any settings that apply to the property. The settings depend upon the property template, see Understanding Property Data Type Parameters.

   For example, the `Core.Description` system property lets you configure the minimum and maximum lengths of its values.
4. In **Column Header**, specify the column header of the import and export file column for the property.

   > **Note:**
   >
   > Column headers must be unique for each property in the application.
   > See Reserved Column Names for a list of labels that cannot be used as column headers.
5. If the page contains **Sync Direction**, specify whether the dimension should be available for importing, exporting, or both.

> **Note:**
>
> Statistical properties in the `CoreStats` namespace are not included in imports and exports.

6. Click **Next** to return to the **Properties for Dimension** page.

   From this page you can add another existing property, create a new property, modify a property, or finish configuring the node type.

Creating a Custom Property

To add a custom property, you must be on the Properties for Dimension Node Type page. You access this page when you perform any of the following tasks:

- Add or modify a node type, see Adding or Modifying a Node Type for a User Defined Dimension.
- After you add an existing property, create a custom property, or modify a property.

> **Tip:**
>
> If during registration of a Universal application, you need to go back to add dimension properties:
>
> 1. Click **Back** until you get to the **Add the Dimensions for the <*app name*> Application** screen.
> 2. Click on the name of the dimension for which you want to add properties.
> 3. Click **Next** on the **Dimension Basic Info** screen.
> 4. Click the name of the node type for which you want to add properties.
> 5. Click **Next** on the **Node Type for Dimension (<*dim name*>)** screen.
>
>   You're now on the **Properties for Dimension (*dim name*) Node Type (*node type*)** screen where you can make changes or add a property.
> 6. Click **Create** to add a new property.

When you create a custom property, the registration process adds the property to the Properties page. The property is assigned to the `Custom` namespace, and other node types will be able to select the property. For example, if you define an `Account Type` property, other node types representing accounts can select that property, see Working with Properties.

To add a custom property to a registered node type, take the following steps.

1. Click **Create**.

   The New Property for Dimension Node Type page displays.
2. Select the **Property Template** that matches the type of data the property will contain.

   For example, if the property values must be whole numbers, select `Custom.Integer Template`.
3. Enter the **Property Name**, and optionally a **Property Description**, and then click **Next**.

   The **Property for Dimension** page displays.

4. For **Property Level**, if the property template is in the `Custom` namespace (for example, `Custom.String Template`), specify whether the property applies to nodes or to relationships between nodes.

   > **Note:**
   >
   > If the property template is not in the `Custom` namespace (for example, `PLN.Alias`), the **Property Level** field is read-only.

5. If the page contains a **Property App Override Parameters** section, you can specify additional settings for the property. The parameters displayed on the page depend upon the property template, see Understanding Property Data Type Parameters.

   For example, suppose a property uses the `Custom.Integer` property template. That template enables you to specify the minimum and maximum values for the property.

6. In **Column Header**, specify the column header of the import and export file column for the property.

   > **Caution:**
   >
   > You must enter a column header.

7. If the page contains **Sync Direction**, specify whether the dimension should be available for importing, exporting, or both.

   > **Note:**
   >
   > Statistical properties in the `CoreStats` namespace are not included in imports and exports.

8. Leave the **Allowed Value Mode** dropdown list set to the default value.

   - > **Note:**
     >
     > You do not need to set this option; it is not used at this time.

     Code – (Default value) Use to map the incoming value to the code value of the property. For example, the code value for the aggregation property is "+".
   - **Display** – Use to map the incoming value to the display value of the property. For example, the display value for the aggregation property is "Addition".

9. Click **Next** to return to the **Properties for Dimension** page.

   From this page you can add another existing property, create a new property, modify a property, or finish configuring the node type.

Modifying a Property for a Node Type in a User Defined Dimension

To modify a custom node type property, you must be on the Properties for Dimension Node Type page. You access this page when you perform any of the following tasks:

- Add or modify a node type, see Adding or Modifying a Node Type for a User Defined Dimension.
- After you add an existing property, create a custom property, or modify a property.

> **Tip:**
>
> If during registration of a Universal application, you need to go back to update dimension properties:
>
> 1. Click **Back** until you get to the **Add the Dimensions for the <*app name*> Application** screen.
> 2. Click on the name of the dimension for which you want to update properties.
> 3. Click **Next** on the **Dimension Basic Info** screen.
> 4. Click the name of the node type for which you want to update properties.
> 5. Click **Next** on the **Node Type for Dimension (<*dim name*>)** screen.
>
>   You're now on the **Properties for Dimension (*dim name*) Node Type (*node type*)** screen where you can make changes or add a property.
> 6. Click the name of a property to edit it or click **Create** to add a new property.

To modify a property, you must be familiar with adding properties, see Adding an Existing Property to a Node Type for a User Defined Dimension and Creating a Custom Property.

1. In the Properties for Dimension Node Type page, click the link for the property.
2. If the New Property for Dimension Node Type page displays, change the applicable settings, and then press **Next**.

   > **Note:**
   >
   > After a property is registered, you can't change its name, data type, or property level.
3. In the Property for Dimension Node Type page, change the applicable settings, then click **Next**. The settings you can change depend upon the data type of the property, see Understanding Property Data Type Parameters.
4. Click **Next** to return to the **Properties for Dimension** page.

   From this page you can add another existing property, create a new property, modify a property, or finish configuring the node type.

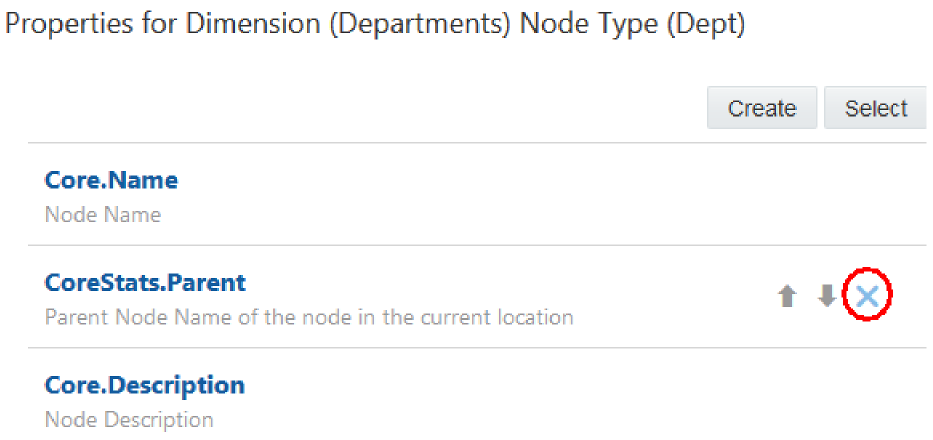
Removing a Property from a Node Type in a User Defined Dimension

If you need to remove a property from a node type for a user-defined dimension, you'll need to perform both of these actions:

- Delete the property from the dimension by modifying the application registration.
- Delete the property from the node type.

To delete the property from the application registration:

1. From **Applications**, click ⋮ next to the application, and then select **Modify Registration**.
2. Click the dimension, and then click the node type that contains the property.
3. Click **Next**.
4. Hover over the far right of the property that you want to delete and click ✕ .



Properties for Dimension (Departments) Node Type (Dept)

Core.Name
Node Name

CoreStats.Parent
Parent Node Name of the node in the current location

Core.Description
Node Description

5. Click **Yes** to confirm the deletion.

To delete the property from the node type:

1. From **Node Types**, click ⋮ next to the node type, and then select **Inspect**.
2. Click **Properties**, and then click **Edit**.
3. Click **Manage**.
4. Clear the check mark from the property that you want to delete, and then click **Save**.

Importing a User Defined Dimension

When you import data for a dimension in a Universal application, you import nodes and their property values. If the dimension is a hierarchy, the import also specifies parent-child relationships between the nodes.

Before you begin:

- See Importing Dimensions for import options, considerations, and information on import modes.
- When you import properties that use Boolean values the following rules apply:
  - If the value in the import file matches the Boolean True or Boolean False Value property definition parameter, then that value is used.
  - Otherwise, if the value in the import file is 1, T, True, Y, or Yes then True is used for the value.
  - Otherwise, False is used for the value.

- You can import data only into properties for which the **Sync Direction** is set to either Import or Both. To review or modify properties, see Modifying a Property for a Node Type in a User Defined Dimension.
- When you import to an existing dimension in Merge or Replace mode, the system checks the node type of the nodes that you are importing, as follows:
  - For applications that do not have a node type column:
    - If the node already exists in the bound dimension, the system finds and uses the existing node and node type when it runs the import.
    - If the node does not already exist in the bound dimension, the system imports the node using the default node type (if the node does not have children) or default parent node type (if the node does have children).
  - For applications that have a node type column, that column is used to create the node during import, as follows:
    - If the node type for the node in the import file is valid for this viewpoint, the node type in the import file is used. A node type is valid for a viewpoint if it is used in the hierarchy set or node set.
    - If the node type for the node in the import file is not valid for this viewpoint, the record is skipped.
    - If the node type for the node in the import file is blank, the system imports the node using the default node type (if the node does not have children) or default parent node type (if the node does have children)..

      > **Note:**
      >
      > If the default node type is not valid for the viewpoint, the system imports the node using the first valid node type alphabetically in the viewpoint.

  - For applications that have a parent node type column, if your dimension contains multiple nodes that have the same name but different node types, the parent node type column is used to locate the node during import, as follows:
    - If the node type for the node in the import file is valid for this viewpoint, it is used to find the node in the node type. If the node cannot be found, the record is skipped.
    - If the node type for the node in the import file is not valid for this viewpoint, the record is skipped.
    - If the node type for the node in the import file is blank, the system uses the logic in the previous bullet for default node and default parent node to try to locate the correct parent. If the parent cannot be determined using this logic, the record is skipped.

1. Create the import file, see Import and Export File Format for User Defined Dimensions.
2. Click **Applications**.
3. In the **Actions** column for the application, click ⋮, and then select **Import**.
4. In the **Dimensions** section, click the dimension into which you will import data.
5. Specify the import file.

   The **File Name** text box displays the name of the import file.

6. In **Import Mode**, select **Merge**, **Replace**, or **Reset**. See Importing Dimensions for details on import modes. .
7. **Optional**: In the Summary section, click the dimension name to inspect the dimension and view the import options. Close the inspector dialog to return to the import screen.
8. Click **Import**.

   A message indicating the data was successfully imported is displayed.
9. Check the **Messages** text box to see if any rows could not be imported. If so, fix the issues and then repeat these steps.

Import and Export File Format for User Defined Dimensions

When you import and export data for user-defined dimensions, you must use a comma-delimited (`.csv`) file.

The following list describes the format and other considerations for import and export files:

- The file must have the following columns:
  - **Name**: The name of the node.
  - **Parent**. The name of the parent node, if the dimension is a hierarchy. If the node does not have a parent node, this field must be empty.

  > **Note:**
  >
  > The column header for **Name** defaults to **Node**. Column headers for Name and Parent can be customized during registration.

- If the dimension has properties, the file contains a column for each property. The column header is the value specified when the property was created, see Adding an Existing Property to a Node Type for a User Defined Dimension and Creating a Custom Property.
- You can create, edit, and delete constant columns in your export file. See *Managing Constant Keys for Universal Applications* in Editing Binding Keys.
- The way in which the dimension is configured determines whether there are columns for node types and how top nodes are indicated, see User Defined Dimension Import and Export Settings.
- Data is imported in row order. Rows for parent nodes must precede rows for child nodes. If a row for a child node precedes the row of its parent node, an error occurs when the file is imported.
- You can place the columns in any order in the import file.
- If a column header does not match those described above, the import ignores the column.

  > **Tip:**
  >
  > This means that an import file can contain information that will not be imported. For example, an import file can include a column for comments.

- The import ignores duplicate rows.

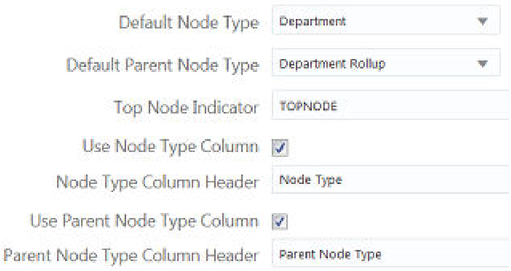
- Strings that contain the delimiter, quote, or line terminator (`CR/LF`) character for the file or that start with characters from ASCII 35 and below (such as `Tab`, `!` or `#`) will be surrounded by quotes. (For example, `"# Children"`, `"Accumulated Depreciation, Equipment"`.)
- Dates and timestamps for supported locales must be in one of these Java date format patterns:
  - Short
  - Default
  - Long
  - Medium

> **Note:**
>
> For information on Java date and time formats, see Using Predefined Formats.

Example 38-1 Import File for a Dimension

Suppose you have a Departments dimension defined as follows:

- There are two node types, which are named Department and Department Rollup.
- Both node types use the `Description` property.
- The Department Rollup node type had a property named `Company`, which has been configured to use the abbreviation **CO** as the column header.
- The dimension has been configured with the following import and export settings:
  - `TOPNODE` represents top nodes.
  - The files can have a column for node types. The column header is **Node Type**.
  - The files can have a column for parent node types. The column header is **Parent Node Type**.
  - If the import file does not contain the **Node Type** and **Parent Node Type** columns, imports will assign bottom nodes to the Department node type and parent nodes to the Department Rollup node type.



Default Node Type: Department

Default Parent Node Type: Department Rollup

Top Node Indicator: TOPNODE

Use Node Type Column: ☑

Node Type Column Header: Node Type

Use Parent Node Type Column: ☑

Parent Node Type Column Header: Parent Node Type

The following table shows the first few rows of an import file for the dimension in this example:

Name	Description	Node Type	Parent	Parent Node Type	CO
TD	Total Department	Department Rollup	TOPNODE		1
100	Resources	Department	TD	Department Rollup	

Name	Description	Node Type	Parent	Parent Node Type	CO
110	Facilities Resources	Department	100	Department	
111	West Region Resources	Department	100	Department	

Note:

The `TOPNODE` value in the Parent column indicates that TD is the root node. The column header for the Company property is **CO**.

Binding Rules for Universal Applications

Binding rules ensure that the Universal application in Oracle Fusion Cloud Enterprise Data Management conforms to the requirements of the external application.

The binding rules that are valid for all application types apply to Universal applications. See Binding Rules for All Application Types.

For more information, see:

- Working with Universal Applications
- Understanding Binding Rules

Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications

You can change the viewpoint bound to a user-defined dimension. Changing the viewpoint bound to a dimension may be necessary if you copied a viewpoint and now want to use the copy for import from and export to the application. When you change the viewpoint, you're informed if the viewpoint that you selected is missing any of the required node type properties or if it contains additional properties.

Considerations

When you change the bound viewpoint for a dimension, you may have additional actions to take to be able to use the selected viewpoint. A message is displayed if there is additional action to take. The additional actions are as follows:

- If the viewpoint being bound uses the same properties, no additional actions are necessary.
- If the viewpoint being bound uses the same node type with different properties, do one of the following:
  - Add the properties to node types for the dimension.
  - Modify the application registration to remove the properties from node types for the dimension.
- If the viewpoint being bound uses a different node type with additional properties, do one of the following:

- Modify the application registration to add the properties to node types for the dimension.
- Remove the additional properties from the node types.

For information on:

- Modifying an application's registration, see Modifying a Universal Application.
- Adding properties to a node type, see Adding, Removing, and Configuring a Node Type's Properties.
- Dimensions, see Understanding Dimensions.

For a business scenario that changes the bound viewpoint, see Changing the Bound Viewpoint to Export an Alternate Viewpoint.

To change the viewpoint bound to a dimension:

1. From **Applications**, find the application containing the dimension.
2. In the application's **Actions** column, click ⋮, then select **Inspect**.
3. Select **Dimensions** and then click the name of the dimension for which you want to change the bound viewpoint.

   The dimension's inspector displays.
4. Click **Edit**.
5. From **View**, select the view containing the viewpoint you want to bind to the dimension, and then select the viewpoint.

   > **Tip:**
   >
   > If the viewpoint that you want to select is not displayed, ensure that the hierarchy set and the application registration have the same setting for Allow Shared Nodes.
6. Click **Save**.

Validations for Universal Applications

Validations ensure data integrity in Universal applications.

The only validations for Universal applications are those that apply to all applications. See System Validations and Generic Predefined Validations.

For more information, see:

- Working with Universal Applications
- Understanding Validations and Constraints

Modifying a Universal Application

You can add, remove, and reconfigure dimensions for a Universal application.

If you are reconfiguring the dimension, you can change the following items:

- Various settings for the dimension. For example, you can change import and export settings.

  > **Note:**
  >
  > After a dimension has been registered, you cannot change its dimension type.

- Add or modify node types. You cannot remove node types from an application. If you want to use a different node type, you can change the bound viewpoint to a different viewpoint that has that node type.

  > **Note:**
  >
  > If you added a node type to a bound hierarchy set or node set via the Inspector, you can edit that node type in the application registration. See Inspecting a Hierarchy Set or Inspecting a Node Set.

- Add, remove, or modify node type properties.

  > **Note:**
  >
  > If you added a property to a bound node type by using the node type inspector, you can edit that property in the application registration. See Inspecting a Node Type.

- Change the order of the export file columns for properties.

> **Note:**
>
> When you remove dimensions and properties, the data objects and properties are unbound. This means that data for removed dimensions and properties will not be imported or exported. However, the data objects and properties are not deleted. This preserves your data should you need it.

For more information, see:

- Understanding Modifying Applications
- Working with Universal Applications

To modify a dimension or its node types or properties, take the following steps.

1. Click **Applications**.
2. In the **Actions** column for the application, click ⋮, and then select **Modify Application**.

   The application registration wizard displays.
3. Perform one of the following steps:
   - To add or modify a dimension, perform the steps in Adding, Removing, or Modifying a User Defined Dimension.

- To modify a node type, including node type properties, click the link for the dimension, click **Next**, then perform the steps in Adding or Modifying a Node Type for a User Defined Dimension.

4. Apply your changes to the application and its data objects by clicking **Apply** on the Review the Registration Before Applying Changes page. You access this page either when you finish modifying the dimension or by clicking the **Summary** link contained by the pages of the registration wizard.

> **Note:**
>
> If you make changes to the registration of an application that is not in Draft status but do not apply them, they are not saved when you close the registration wizard.
>
> If you make changes to the registration of an application in Draft status but do not apply them, they are saved and may be able to be updated and applied later. However, if you make changes to bound data objects outside of the registration wizard it may clear the saved changes because the registration must be kept in sync with the bound objects.

Exporting a User Defined Dimension

You can export data for a dimension in a Universal application to a comma-delimited (`.csv`) file.

Best Practice

It's a best practice to validate the viewpoints bound to the dimension prior to performing the export to ensure data integrity, see Understanding Validations and Constraints.

Use the **Validate Before Export** option to validate the viewpoint automatically when you export a dimension. See Managing Export Options.

For the CSV file format, see Import and Export File Format for User Defined Dimensions.

The nodes and hierarchical relationships are exported in alphanumeric order by default. If you want to change the order of exported nodes, before you export, open the hierarchy set and check **Use Custom Order**.

You can edit the binding keys for a dimension to change the order of the export columns. See Editing Binding Keys.

You can export a defined keyword instead of a null or blank value. See Managing Export Options.

You can export data only from properties whose direction is set to either Export or Both. To review or modify this setting, perform an action:

- Modify the **Sync Direction** of the property in application registration. See Modifying a Property for a Node Type in a User Defined Dimension.
- Modify the **Direction** of the property in the binding keys for the dimension. See Editing Binding Keys

> **Note:**
>
> You can also export data that is mapped to nodes in another application, see Exporting Mapping Data.

For more information, see:

- Working with Universal Applications
- Exporting Dimensions

1. Click **Applications**.
2. In the **Actions** column for the application, click ⋮, and then select **Export**.
3. In the **Dimensions** section, select the dimension to export. An export file name is generated. The default file name is *Application Name_Dimension Name_Date*, for example Custom1_Account_20200123.csv, and can be edited before running the export. Your browser settings define the download location.
4. **Optional**: In the Summary section, click the dimension name to modify the export options for the dimension in the inspector. See Inspecting Dimensions.
5. Click **Export**.
6. **Optional**: After the export is complete, if there are validation errors in the export you can click the viewpoint name to open the viewpoint and resolve them. See Resolving Validation Issues. When you have finished resolving the validation issues, from the viewpoint window click **Return to Export** to return to the export screen.

39 Working with Users Applications

Users applications let you manage a common set of enterprise users who can be associated with nodes in other applications in Oracle Fusion Cloud Enterprise Data Management. Nodes in Users applications do not count towards your record count.

The users that you manage in a Users app can be referenced in these ways:

- As node owners in approval workflows that use the Ownership approval method. See Understanding Ownership Approvals.

  > **Tip:**
  >
  > If the `Email Address` property contains the email address of a valid user in Cloud EDM, that user can be designated as a node owner for ownership approval policies. See Predefined Properties for Users Applications.

- As reference data on nodes. For example, nodes in the Cost Center dimension of a Planning application can use a node data type property with the User subtype to reference users in a Users application as the cost center managers. See Node and Node List Data Type Subtypes.

Considerations

While other application types in Cloud EDM are generally used to govern enterprise data from an external application, Users applications are unique in that they are only used internally to manage your users. As such, they different from other application types in several important ways:

- You do not use a registration wizard to register a Users application. Instead, when you register a Users application, you provide a name for the application and the system automatically creates the data chain objects. See Registering Users Applications.
- The following integration operations are not available:
  - Import
  - Export
  - Extract
  - Mapping Export
- After you register a Users application, you cannot modify the registration from the application worklist. Inspect the application to make changes to the viewpoints or data chain objects.
- The following tabs are not available in the application inspector:
  - Registration
  - Connections

Registering Users Applications

You must have the *Application - Create* role to register an application. After the registration, you are automatically granted the *Owner* permission on the application and the default view.

To register a Users application:

1. From **Applications**, click **Register**, and then select **Users**.
2. Enter an application name and, optionally, a description, and then click **Next**.
3. Review the application summary information, and then click **Create**.

When you create a Users application, the following objects are created. All objects are unbound.

Object	Name
Application	(Application name that you entered)
Default View	(Application name that you entered)
Viewpoints	• **User List** (list viewpoint) • **User Hierarchy** (hierarchy viewpoint)
Dimension	**Users**
Node types	• **User** • **Rollup** **Note**: These node types use the **User** and **User Rollup** specialty node type classes, and they are created with limited properties that cannot be changed. See Working with Specialty Node Type Classes.
Hierarchy set	**User Hierarchy** (uses the User and User Rollup node types)
Node sets	• **User List** (points to the User node type) • **User Hierarchy** (points to the User Hierarchy hierarchy set).

Predefined Properties for Users Applications

Users applications use the User and User Rollup specialty node types (see Working with Specialty Node Type Classes). The following properties are predefined for these node types.

You cannot add or remove properties from these node types, but you can rearrange the property order and set the Required flag. You can also change their labels in a viewpoint, and you can hide properties that you are not using in the viewpoint. See Configuring How a Viewpoint Displays Properties.

Table 39-1 Predefined Properties for User Node Types

Property	Namespace	Data Type	Level	Description
Name	`Core`	String	Node	Node name
Description	`Core`	String	Node	Node description

Table 39-1 (Cont.) Predefined Properties for User Node Types

Property	Namespace	Data Type	Level	Description
Username	`User`	String	Node	User name for the user. This does not have to be the Oracle Fusion Cloud Enterprise Data Management user name.
Email Address	`User`	String (email subtype)	Node	Email address for the user. **Note**: This is used to derive the values for the EDM Username and EDM User properties.
Title	`User`	String	Node	User title
First Name	`User`	String	Node	First name of the user
Middle Name	`User`	String	Node	Middle name of the user
Last Name	`User`	String	Node	Last name of the user
Full Name	`User`	String	Node	Full name of the user **Note**: By default, this is derived by concatenating the First, Middle, and Last name of the user.
Start Date	`User`	Date	Node	Start date for the user
End Date	`User`	Date	Node	End date for the user
EDM Username	`CoreStats`	String	Node	Returns the Cloud EDM username if the `User.Email Address` property matches a valid user.
EDM User	`CoreStats`	Boolean	Node	Returns True if the `User.Email Address` property matches a valid user.
Parent	`CoreStats`	Node	Relationship	Parent node name of the node in the current location

Table 39-1 (Cont.) Predefined Properties for User Node Types

Property	Namespace	Data Type	Level	Description
Level	`CoreStats`	Integer	Relationship	Level of the node in a hierarchy set
Bottom Node	`CoreStats`	Boolean	Relationship	Returns True if the node has no children

Table 39-2 Predefined Properties for User Rollup Node Types

Property	Namespace	Data Type	Level	Description
Name	`Core`	String	Node	Node name
Description	`Core`	String	Node	Node description
Parent	`CoreStats`	Node	Relationship	Parent node name of the node in the current location
Level	`CoreStats`	Integer	Relationship	Level of the node in a hierarchy set
Bottom Node	`CoreStats`	Boolean	Relationship	True if the node has no children

Part III
Business Scenarios

The following sections describe business scenarios and the tasks performed by administrators and users.

- Integrating with Planning
- Integrating with Oracle Financials Cloud General Ledger
- Integrating with External Applications Using a Universal Application
- Sharing Enterprise Data Across Applications
- Comparing Enterprise Data Across Applications
- Managing Enterprise Data as Lists
- Mapping Source Dimensions to Target Dimensions
- Migrating Enterprise Data from Data Relationship Management to Cloud EDM
- Changing the Bound Viewpoint to Export an Alternate Viewpoint
- Backing up and Restoring Cloud EDM
- Managing Governance Workflows and Approvals
- Aligning Applications with Subscriptions and Governance Workflow

For the purpose of these business scenarios:

- An administrator is someone who sets up users or configures Oracle Fusion Cloud Enterprise Data Management functionality.
- A user is someone who manages data.

40
Integrating with Planning

This scenario demonstrates how to manage your Planning application using the Planning wizard. The steps walk you through registering your Planning application and importing your dimensions, managing your enterprise data, and exporting changes back to your application.

The steps below are focused on the administrative tasks required to complete the register to export process within Oracle Fusion Cloud Enterprise Data Management. Only a general overview of data management tasks are provided with this scenario.

Administrator Steps

Administrator Prerequisites

A dimensional export from your Planning application is required when using the Planning wizard. You must prepare a dimension export that includes all the dimensionality you want to import and manage in Oracle Fusion Cloud Enterprise Data Management. Create this export and run the job so that the resulting .CSV file is generated and placed into your Planning Inbox/Outbox.

See Working with Planning and FreeForm Applications

Administrator Process

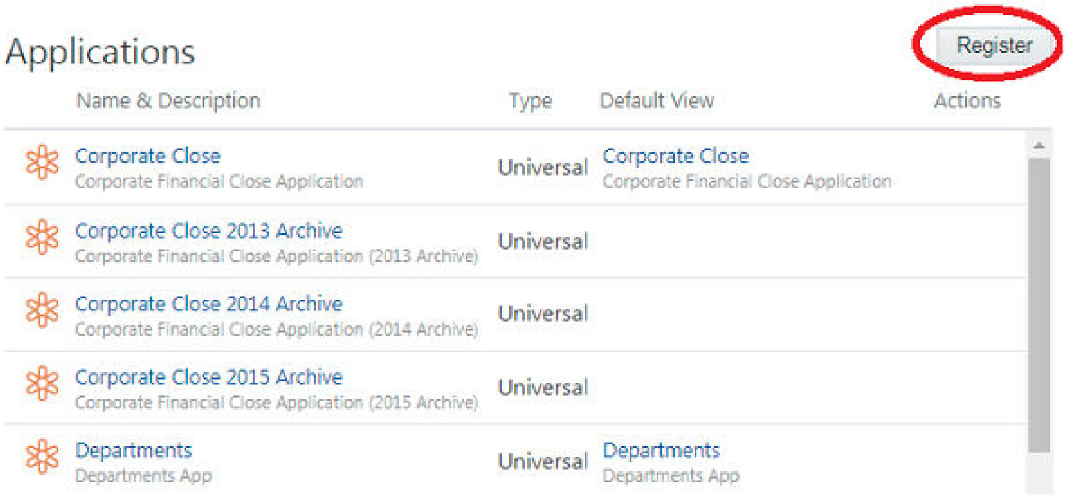
> **Note:**
>
> You need the Application - Create role to register the application. After you register an application you are automatically assigned the *Owner* permission for the application and the default view.

1. Register your Planning application. Registering Planning and FreeForm Applications
2. Import the dimensions registered. Planning and FreeForm Validations
3. Provide your users with the appropriate access to the data chain elements needed to manage the dimensionality for the Planning application. Working with Roles and Permissions
4. Export dimensionality back to your Planning application. Exporting Planning and FreeForm Dimensions

Administrator Detailed Steps

Step 1: Register Your Planning Application
To register your Planning application:

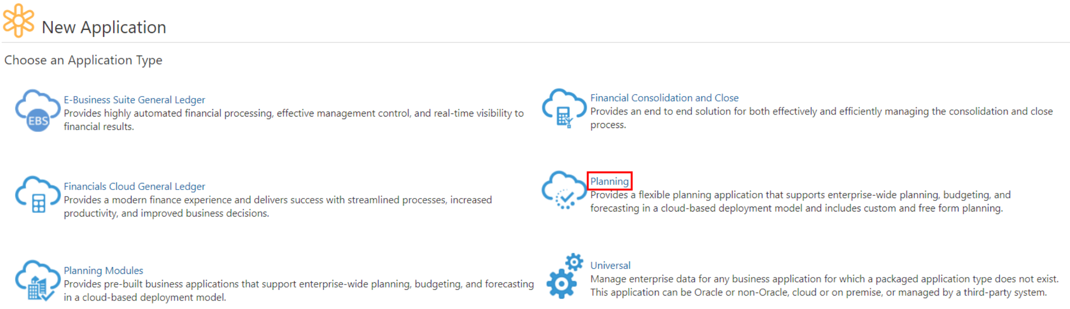
1. In **Applications**, click **Register**.



Applications

Name & Description	Type	Default View	Actions
Corporate Close Corporate Financial Close Application	Universal	Corporate Close Corporate Financial Close Application	
Corporate Close 2013 Archive Corporate Financial Close Application (2013 Archive)	Universal		
Corporate Close 2014 Archive Corporate Financial Close Application (2014 Archive)	Universal		
Corporate Close 2015 Archive Corporate Financial Close Application (2015 Archive)	Universal		
Departments Departments App	Universal	Departments Departments App	

2. Select **Planning**.



New Application

Choose an Application Type

E-Business Suite General Ledger
Provides highly automated financial processing, effective management control, and real-time visibility to financial results.

Financial Consolidation and Close
Provides an end to end solution for both effectively and efficiently managing the consolidation and close process.

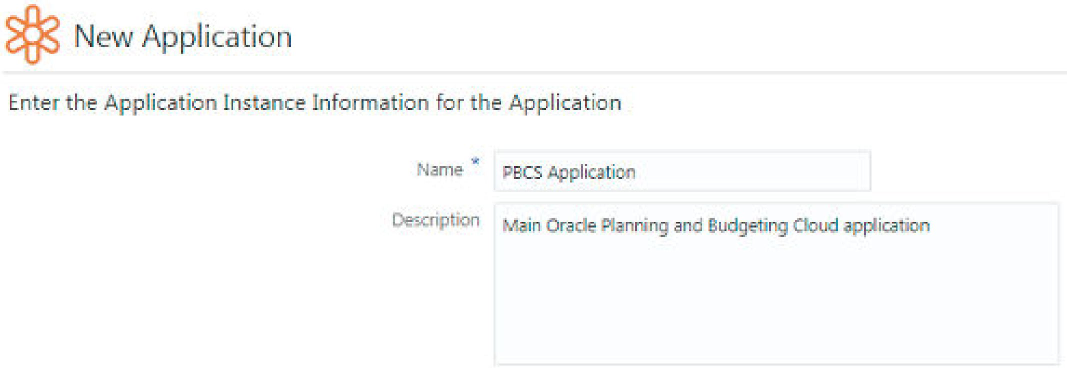
Financials Cloud General Ledger
Provides a modern finance experience and delivers success with streamlined processes, increased productivity, and improved business decisions.

Planning
Provides a flexible planning application that supports enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model and includes custom and free form planning.

Planning Modules
Provides pre-built business applications that support enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model.

Universal
Manage enterprise data for any business application for which a packaged application type does not exist. This application can be Oracle or non-Oracle, cloud or on premise, or managed by a third-party system.

3. Provide a name and description.



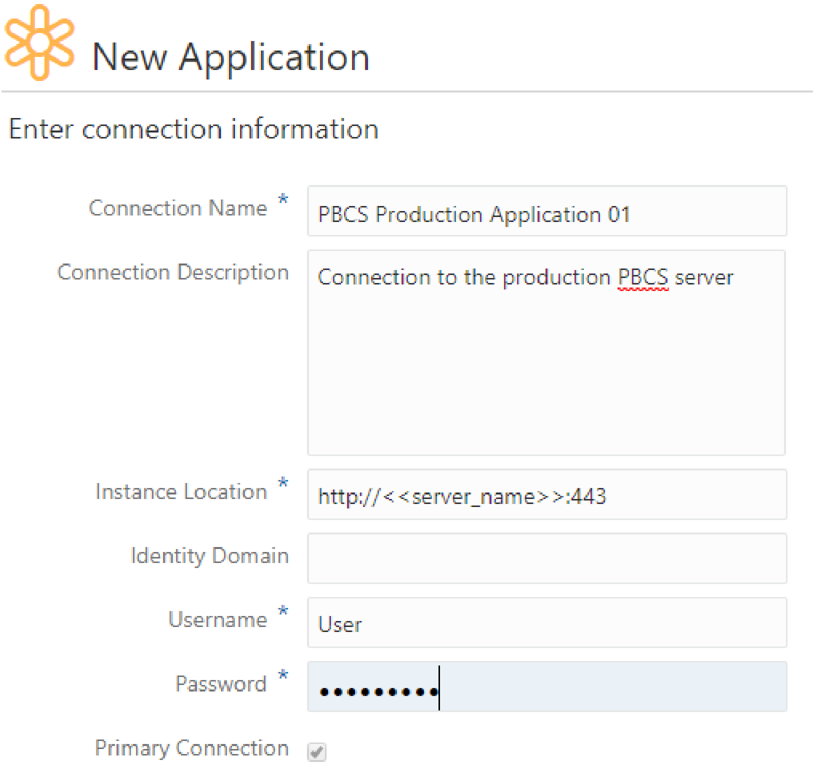
New Application

Enter the Application Instance Information for the Application

Name * PBCS Application

Description Main Oracle Planning and Budgeting Cloud application

4. Provide connection information to your Planning application:



New Application

Enter connection information

Connection Name * PBCS Production Application 01

Connection Description Connection to the production PBCS server

Instance Location * http://<<server_name>>:443

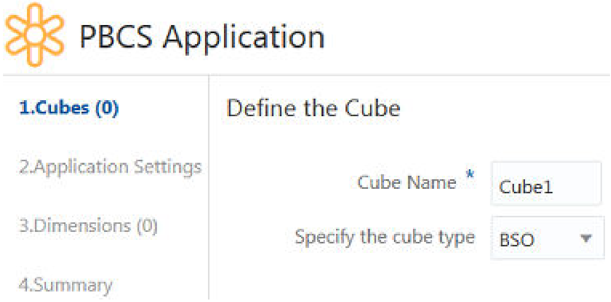
Identity Domain

Username * User

Password * •••••••••

Primary Connection ☑

5. Define your cubes.



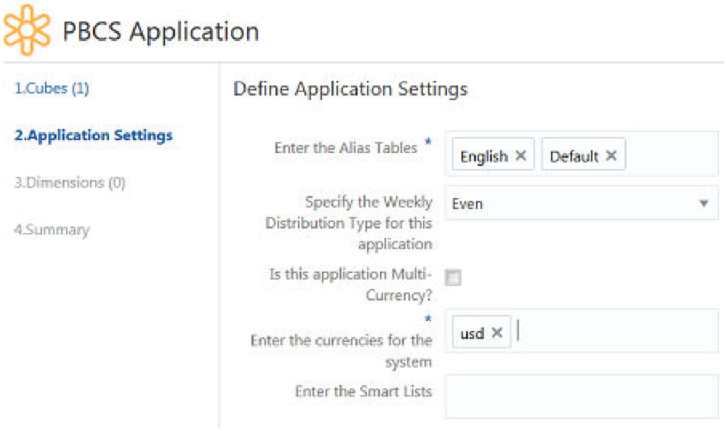
PBCS Application

Define the Cube

Cube Name * Cube1

Specify the cube type BSO

6. Configure your application settings.



PBCS Application

Define Application Settings

Enter the Alias Tables * English ✕ Default ✕

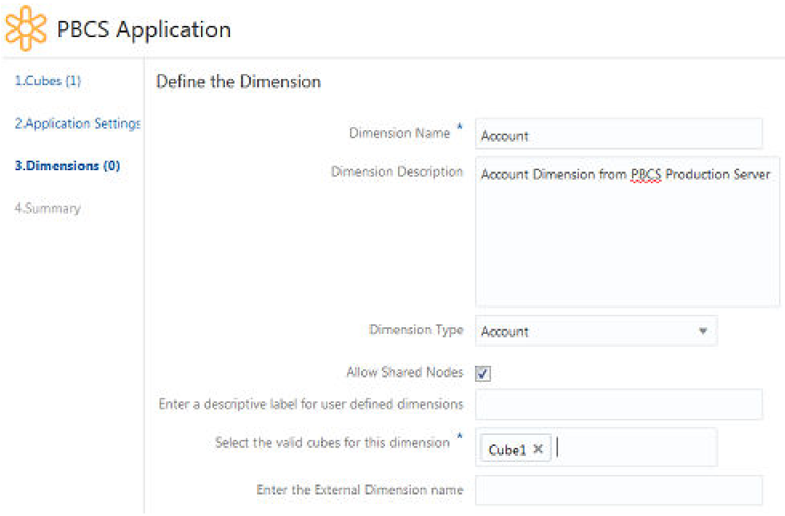
Specify the Weekly Distribution Type for this application Even

Is this application Multi-Currency? □

* Enter the currencies for the system usd ✕

Enter the Smart Lists

7. Define the dimensions you want to import from your Planning application.



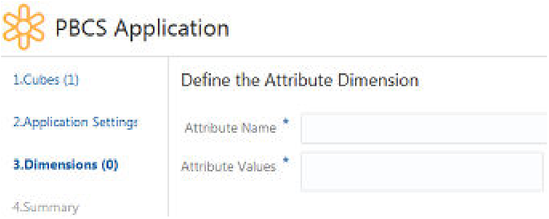
PBCS Application

- 1.Cubes (1)
- 2.Application Settings
- 3.Dimensions (0)
- 4.Summary

Define the Dimension

Dimension Name *	Account
Dimension Description	Account Dimension from PBCS Production Server
Dimension Type	Account
Allow Shared Nodes	☑
Enter a descriptive label for user defined dimensions	
Select the valid cubes for this dimension *	Cube1 ×
Enter the External Dimension name	

8. Define any attribute dimensions for base dimensions you are registering.



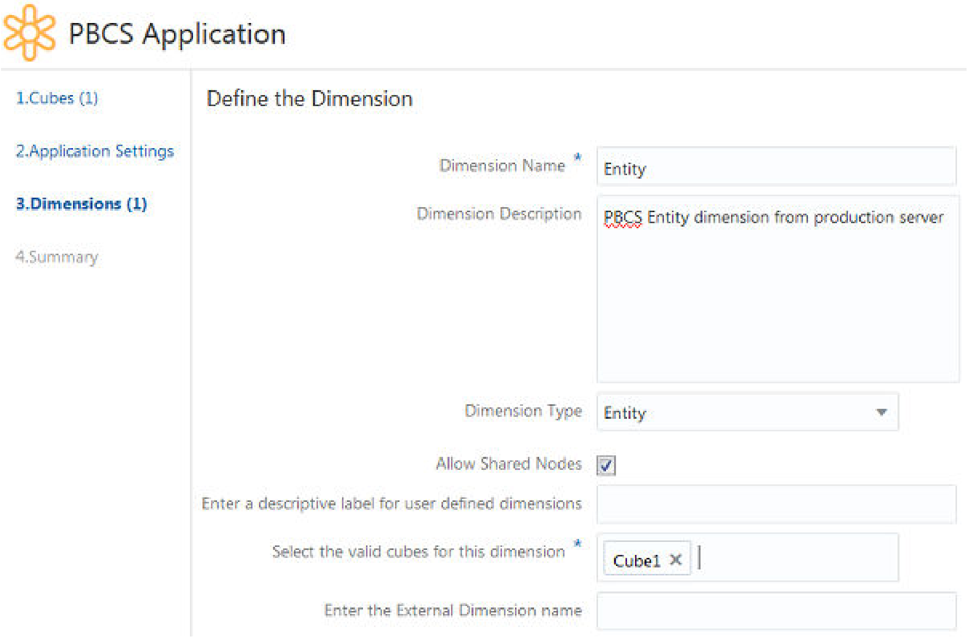
PBCS Application

- 1.Cubes (1)
- 2.Application Settings
- 3.Dimensions (0)
- 4.Summary

Define the Attribute Dimension

Attribute Name *	
Attribute Values *	

9. Add any other dimensions and their attribute dimensions to the registration.



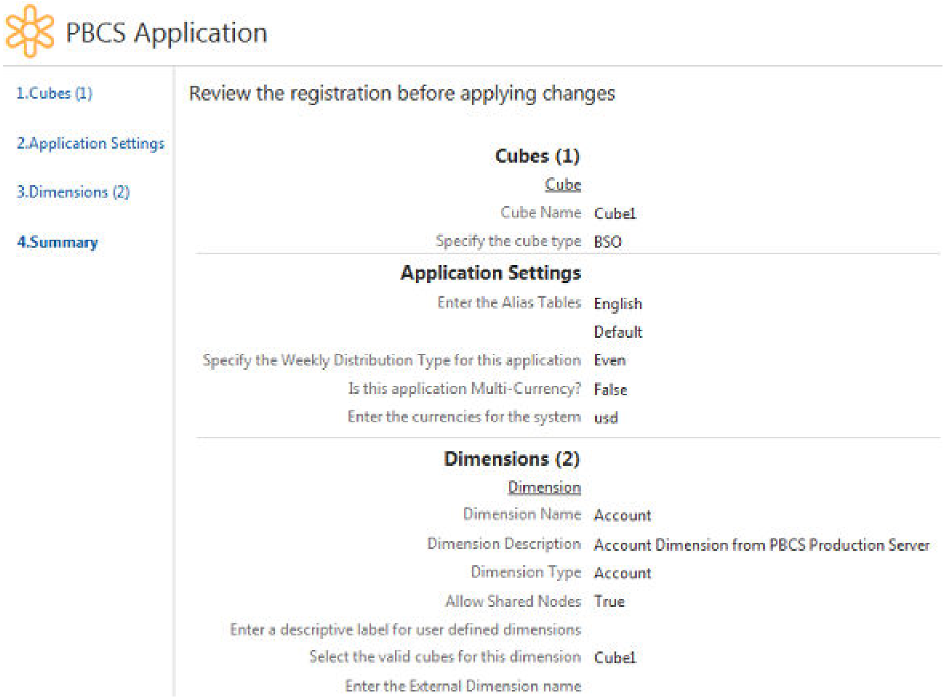
PBCS Application

- 1.Cubes (1)
- 2.Application Settings
- 3.Dimensions (1)
- 4.Summary

Define the Dimension

Dimension Name *	Entity
Dimension Description	PBCS Entity dimension from production server
Dimension Type	Entity
Allow Shared Nodes	☑
Enter a descriptive label for user defined dimensions	
Select the valid cubes for this dimension *	Cube1 ×
Enter the External Dimension name	

10. Review all registration information and click **Apply** to register your Planning application.



PBCS Application

Review the registration before applying changes

Cubes (1)

Cube

Cube Name: Cube1
Specify the cube type: BSO

Application Settings

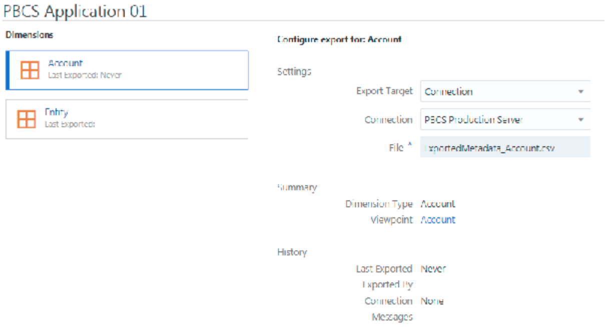
Enter the Alias Tables: English
Default
Specify the Weekly Distribution Type for this application: Even
Is this application Multi-Currency?: False
Enter the currencies for the system: usd

Dimensions (2)

Dimension

Dimension Name: Account
Dimension Description: Account Dimension from PBCS Production Server
Dimension Type: Account
Allow Shared Nodes: True
Enter a descriptive label for user defined dimensions
Select the valid cubes for this dimension: Cube1
Enter the External Dimension name

Step 2: Import the Dimensions You Registered



PBCS Application 01

Dimensions

Configure export for: Account

Settings

Export Target: Connection
Connection: PBCS Production Server
File *: ExportedMetadata_Account.csv

Summary

Dimension Type: Account
Viewpoint: Account

History

Last Exported: Never
Exported By
Connection: None
Messages

Step 3: Implement the Changes Made By Your Users

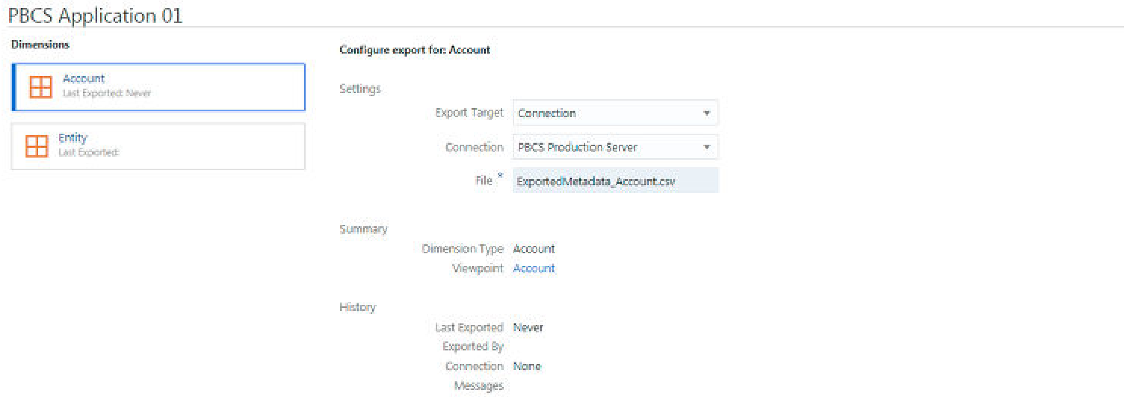
In this step, your users make changes to the enterprise data.

Step 4: Export Dimensionality Back to Your Planning Application

Before you export:

- Ensure the dimension is bound.
  - Understanding Binding Rules
  - Binding Rules for Planning and FreeForm Applications
- Validate the viewpoint.
  - Validating a Viewpoint
  - Resolving Validation Issues
  - Planning and FreeForm Validations

From the Applications card, select the Planning application’s connection. To import, user can specify Export Target, Connection, and File. Repeat this step for each dimension you want to export out of Cloud EDM and into your Planning application.



PBCS Application 01

Dimensions

- Account
  Last Exported: Never
- Entity
  Last Exported:

Configure export for: Account

Settings

Export Target	Connection
Connection	PBCS Production Server
File *	ExportedMetadata_Account.csv

Summary

Dimension Type	Account
Viewpoint	Account

History

Last Exported	Never
Exported By	
Connection	None
Messages	

After the files are exported, follow your Planning application's instructions on importing these changes.

User Steps

User Prerequisites

Your administrator must follow the steps in this scenario to successfully register your Planning application and import enterprise data elements. Then you can manage your enterprise data based on your security privileges.

User Process

This is a high level summary of the tasks you can perform:

- Make maintenance views and viewpoints, see Working with Viewpoints
- Use side by side layout and compare to rationalize nodes across viewpoints within a view, see Comparing Viewpoints
- Update nodes and properties and update alternate hierarchies for what if analysis, see Making Changes Interactively
- Map nodes from one hierarchy to another, see Creating Mapping Viewpoints
- Load enterprise data such as accounts, see Making Changes Using a Load File
- Browse draft and completed requests in the Requests list, see Working with Request Activity

Outcome

You now have registered your Planning and Budgeting Cloud application within Enterprise Data Management Cloud so you can proactively manage changes across this application and any others that are registered. You have imported the dimensionality you wish to manage and have successfully exported changes out of Enterprise Data Management Cloud and back into Planning and Budgeting Cloud. You have full request activity audit information to know who makes changes, when, and where.

41 Integrating with Oracle Financials Cloud General Ledger

This scenario demonstrates how to configure a Oracle Fusion Cloud Enterprise Data Management application so that you can manage Oracle Financials Cloud General Ledger value sets.

The steps walk you through creating a Cloud EDM application, populating the application by using a template to import account information from Oracle Financials Cloud General Ledger, making changes to the data, exporting the data back to Oracle Financials Cloud General Ledger, loading the changes in the General Ledger service, and publishing the account hierarchy.

Because Cloud EDM and Oracle Financials Cloud General Ledger are built on different data models, some of the terminology is different. The following table provides a guide to Cloud EDM terms and their Oracle Financials Cloud equivalents.

Cloud EDM Term	Oracle Financials Cloud General Ledger Equivalent
Nodes	Values
Dimension	Segment
List viewpoint	List of values in a value set
Hierarchy viewpoint	Relationships in a tree version within a tree code

Cloud EDM Administrator Steps

Cloud EDM Administrator Prerequisites

You must have a chart of accounts value set set up in Oracle Financials Cloud General Ledger. If your value set has a hierarchy, the tree structures must also be set up.

You must have the *Application - Create* role to register the application. After you register an application you are automatically assigned the *Owner* permission for the application and the default view. You must also have permission to upload files to the Universal Content Management (UCM) module in your Oracle Financials Cloud service (for example, the Financial Application Administrator job role).

Cloud EDM Administrator Process

1. Registering an Oracle Financials Cloud General Ledger application in Oracle Fusion Cloud Enterprise Data Management. See Registering Oracle Financials Cloud General Ledger Applications.
2. Importing Oracle Financials Cloud General Ledger chart of accounts value sets into Cloud EDM. See Importing Oracle Financials Cloud General Ledger Dimensions.

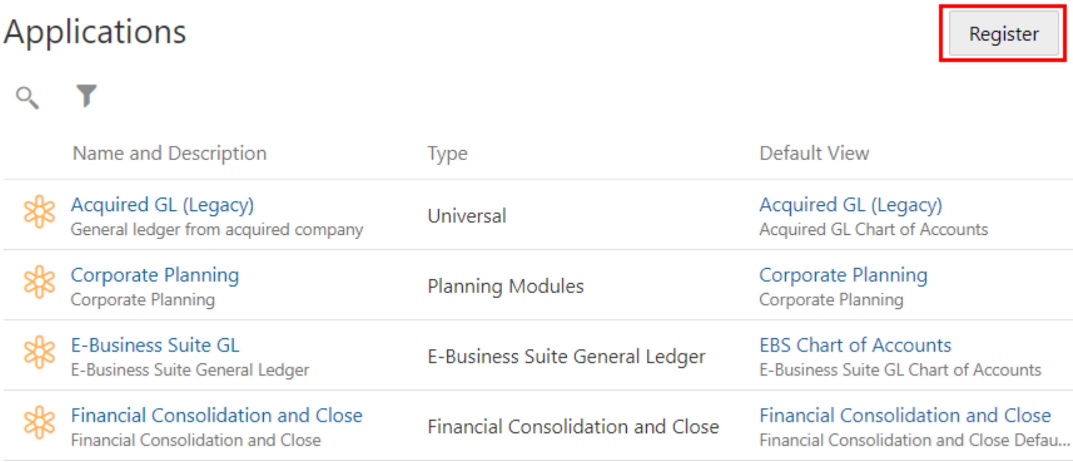
Detailed Steps

Related Topics

- Step 1: Register Your Oracle Financials Cloud General Ledger Application in Cloud EDM
- Step 2: Importing Your Oracle Financials Cloud General Ledger Value Set into Cloud EDM

Step 1: Register Your Oracle Financials Cloud General Ledger Application in Cloud EDM

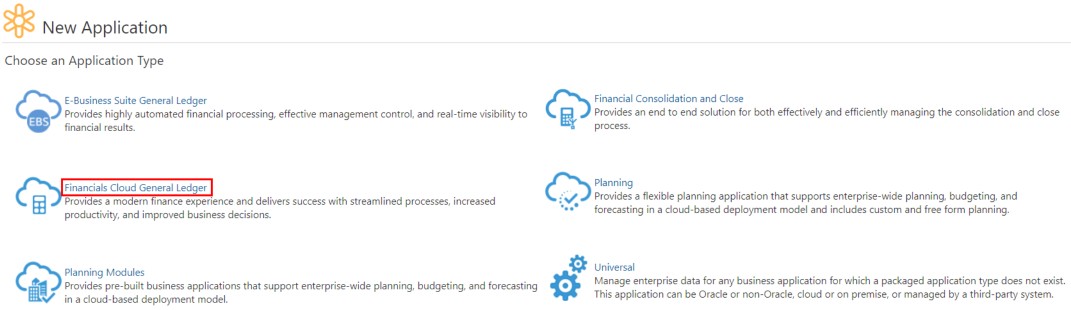
1. From Applications, click **Register**.



Applications

Name and Description	Type	Default View
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning
E-Business Suite GL E-Business Suite General Ledger	E-Business Suite General Ledger	EBS Chart of Accounts E-Business Suite GL Chart of Accounts
Financial Consolidation and Close Financial Consolidation and Close	Financial Consolidation and Close	Financial Consolidation and Close Financial Consolidation and Close Defau...

2. Select the **Financials Cloud General Ledger** application type.



New Application

Choose an Application Type

E-Business Suite General Ledger
Provides highly automated financial processing, effective management control, and real-time visibility to financial results.

Financial Consolidation and Close
Provides an end to end solution for both effectively and efficiently managing the consolidation and close process.

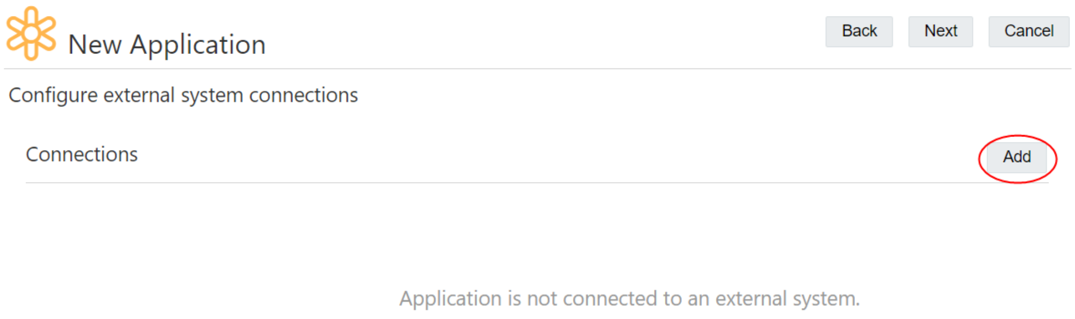
Financials Cloud General Ledger
Provides a modern finance experience and delivers success with streamlined processes, increased productivity, and improved business decisions.

Planning
Provides a flexible planning application that supports enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model and includes custom and free form planning.

Planning Modules
Provides pre-built business applications that support enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model.

Universal
Manage enterprise data for any business application for which a packaged application type does not exist. This application can be Oracle or non-Oracle, cloud or on premise, or managed by a third-party system.

3. Click **Add** to provide connection information to your external Oracle Financials Cloud General Ledger service.



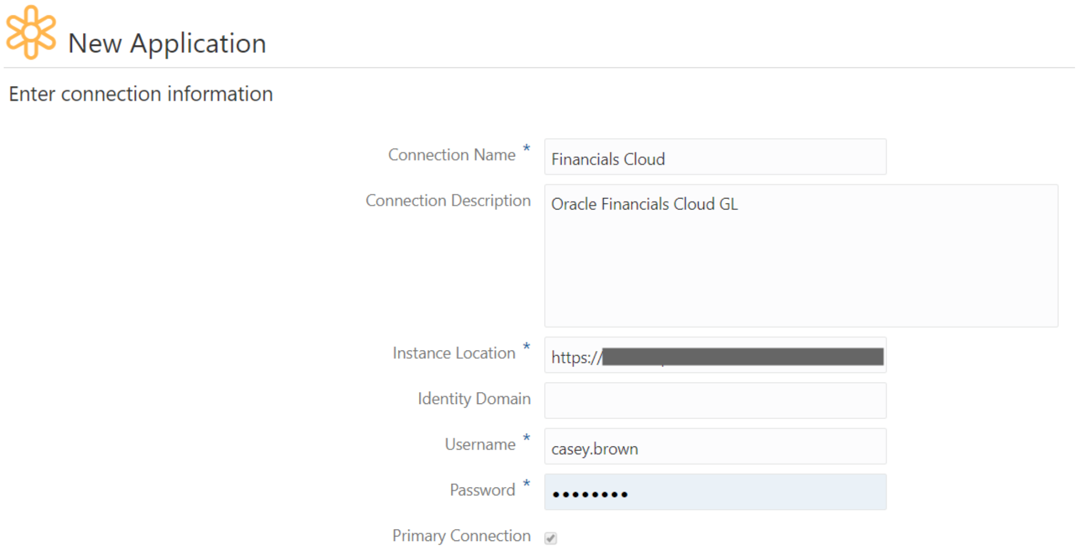
New Application

Configure external system connections

Connections

Application is not connected to an external system.

4. Enter the connection details for your external Oracle Financials Cloud General Ledger service, and then click **Next**.

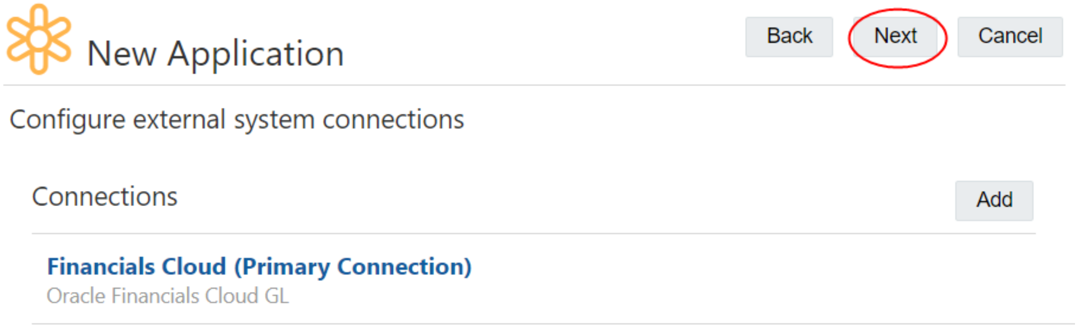


New Application

Enter connection information

Connection Name *	Financials Cloud
Connection Description	Oracle Financials Cloud GL
Instance Location *	https:/[redacted]
Identity Domain	
Username *	casey.brown
Password *	••••••••
Primary Connection	☑

5. Click **Next**.



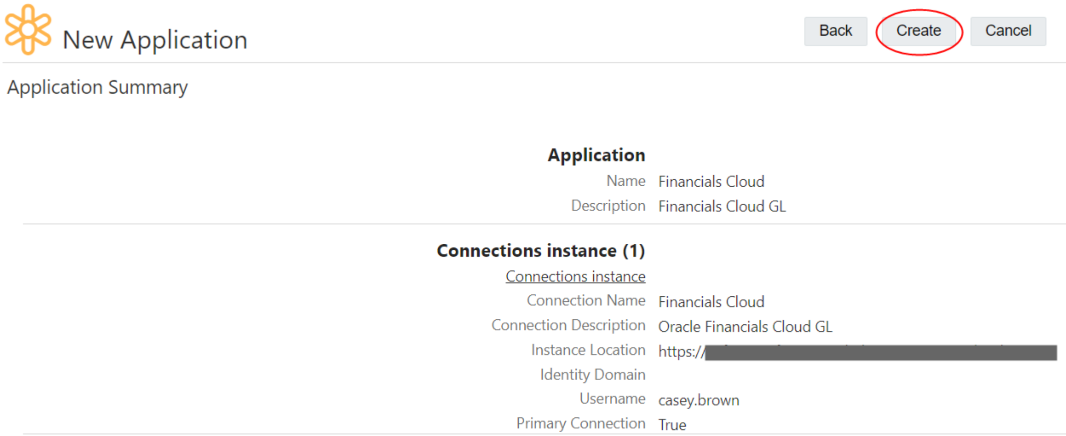
New Application

Configure external system connections

Connections

Financials Cloud (Primary Connection)
Oracle Financials Cloud GL

6. On the Application Summary screen, click **Create**.



Application Summary

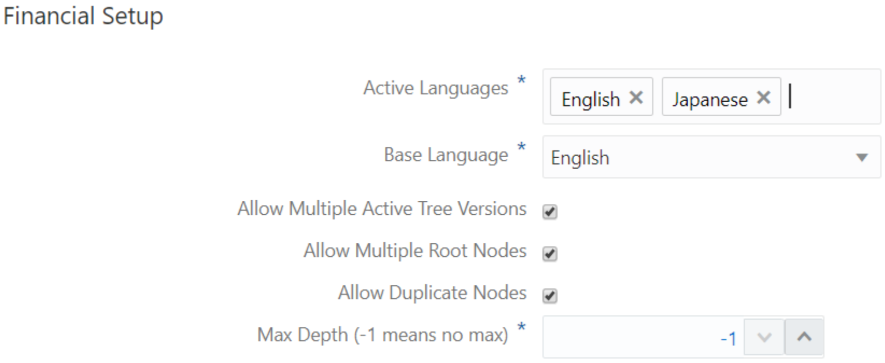
Application

Name: Financials Cloud
Description: Financials Cloud GL

Connections instance (1)

Connections instance
Connection Name: Financials Cloud
Connection Description: Oracle Financials Cloud GL
Instance Location: https://[redacted]
Identity Domain:
Username: casey.brown
Primary Connection: True

7. On the Financial Setup screen, enter the following information, and then click **Next**.
   - **Active Languages**: Languages that have an Active status in the Oracle Financials Cloud application.
   - **Base Language**: Select a single base language from any of the active languages.
   - **Allow Multiple Active Tree Versions**: Allows multiple active tree versions in the tree structure definition.
   - **Allow Multiple Root Nodes** : Enables more than one root node in the hierarchy.
   - **Allow Duplicate Nodes** : Enables shared nodes.
   - **Max Depth**: Enter the depth limit for the data in your tree structure, or enter `-1` if your tree structure does not have a depth limit.

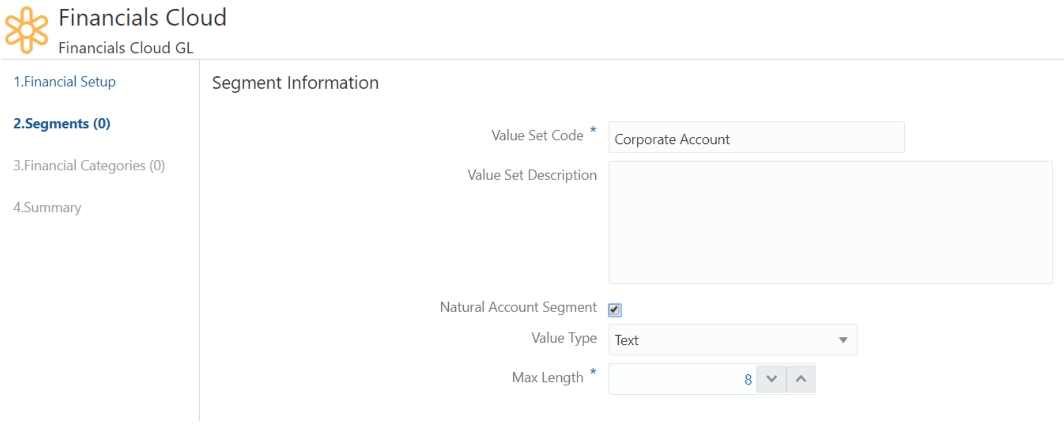


Financial Setup

Active Languages *: English ✕ Japanese ✕
Base Language *: English
Allow Multiple Active Tree Versions ☑
Allow Multiple Root Nodes ☑
Allow Duplicate Nodes ☑
Max Depth (-1 means no max) *: -1

8. On the Segment page, enter the following segment information, and then click **Next**.
   - **Value Set Code**: Enter the value set code from your Oracle Financials Cloud application.
   - **Value Set Description**: Enter a description for the value set.
   - **Natural Account Segment**: Select if this segment is used as the natural account in a chart of accounts.
   - **Value Type**: Specify if the values in the value set are text or numeric.

- **Max Length**: Enter the maximum character length for the values in the value set.

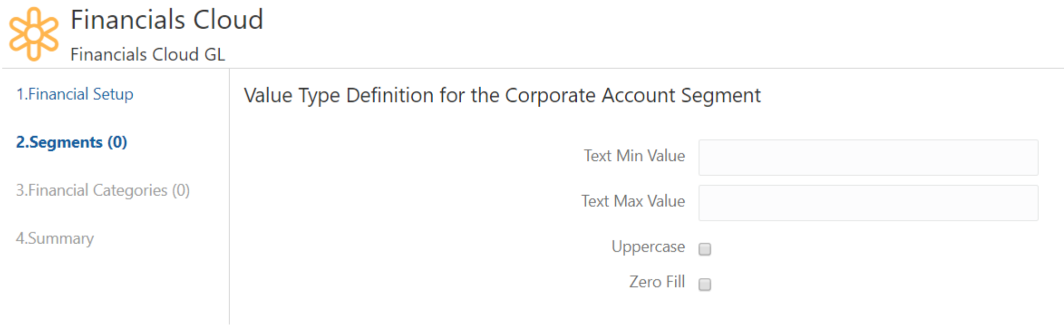


Financials Cloud
Financials Cloud GL

Segment Information

Value Set Code * Corporate Account
Value Set Description
Natural Account Segment ☑
Value Type Text
Max Length * 8

9. On the Value Type Definition page, enter the minimum and maximum values, specify if the text must be uppercase and zero-filled to the maximum character length, and then click **Next**.



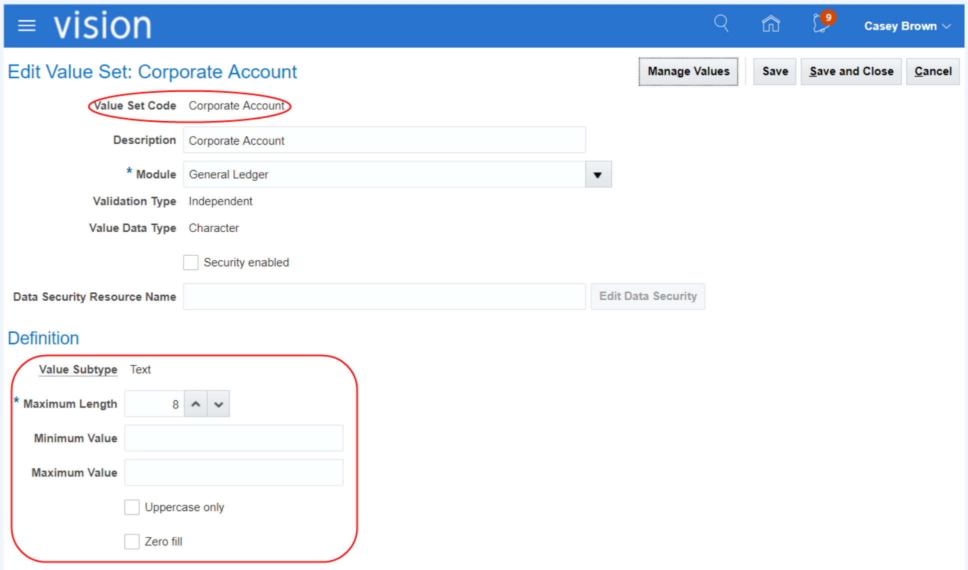
Financials Cloud
Financials Cloud GL

Value Type Definition for the Corporate Account Segment

Text Min Value
Text Max Value
Uppercase □
Zero Fill □

Note:

The settings for Segment Information and Value Type Definition fields in the previous steps should match the corresponding settings in your Oracle Financials Cloud value set. You can check these settings in Oracle Financials Cloud General Ledger by using the **Manage Value Sets** task.



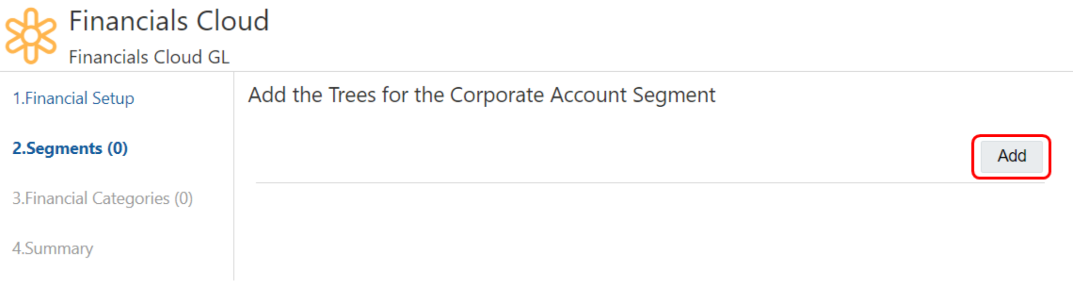
Edit Value Set: Corporate Account

Value Set Code: Corporate Account
Description: Corporate Account
* Module: General Ledger
Validation Type: Independent
Value Data Type: Character
□ Security enabled
Data Security Resource Name:

Definition

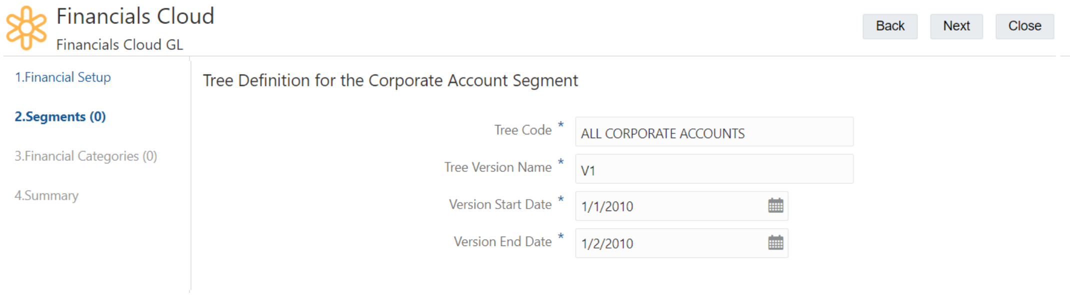
Value Subtype: Text
* Maximum Length: 8
Minimum Value:
Maximum Value:
□ Uppercase only
□ Zero fill

10. Next, you add trees for the Corporate Account segment. Click **Add**.



Add the Trees for the Corporate Account Segment

11. Enter the **Tree Code**, **Tree Version Name**, **Version Start Date**, and **Version End Date**, and then click **Next**.

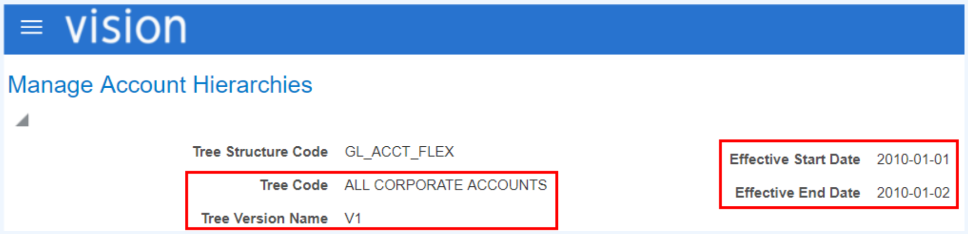


Tree Definition for the Corporate Account Segment

Tree Code *: ALL CORPORATE ACCOUNTS
Tree Version Name *: V1
Version Start Date *: 1/1/2010
Version End Date *: 1/2/2010

Note:

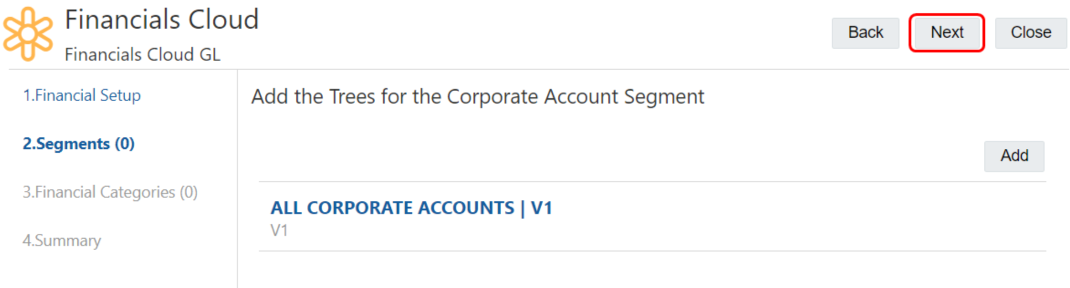
The tree definition should match the corresponding tree definition in your Oracle Financials Cloud account hierarchy. You can check these settings in Oracle Financials Cloud General Ledger by using the **Manage Account Hierarchies** task.



Manage Account Hierarchies

Tree Structure Code GL_ACCT_FLEX
Tree Code ALL CORPORATE ACCOUNTS
Tree Version Name V1
Effective Start Date 2010-01-01
Effective End Date 2010-01-02

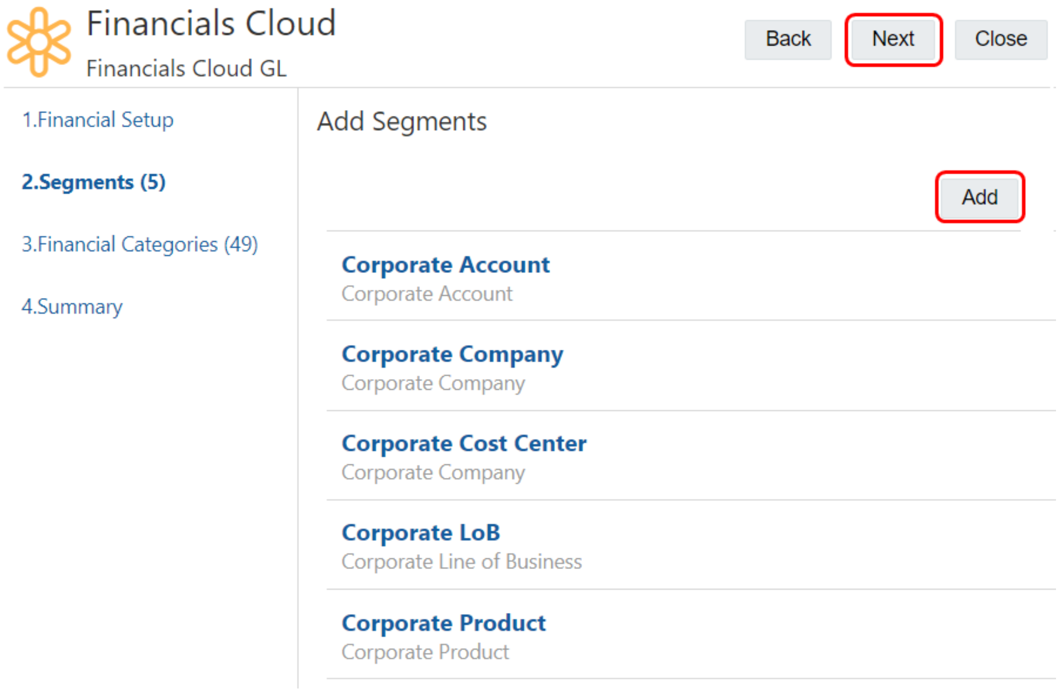
12. Click **Next** to return to the Add Segments page.



Add the Trees for the Corporate Account Segment

ALL CORPORATE ACCOUNTS | V1
V1

13. Click **Add** to continue adding segments and trees, and then click **Next** when you are finished. For this scenario, we added segments for Corporate Company, Corporate Cost Center, Corporate LoB, and Corporate Product.



Corporate Account
Corporate Account

Corporate Company
Corporate Company

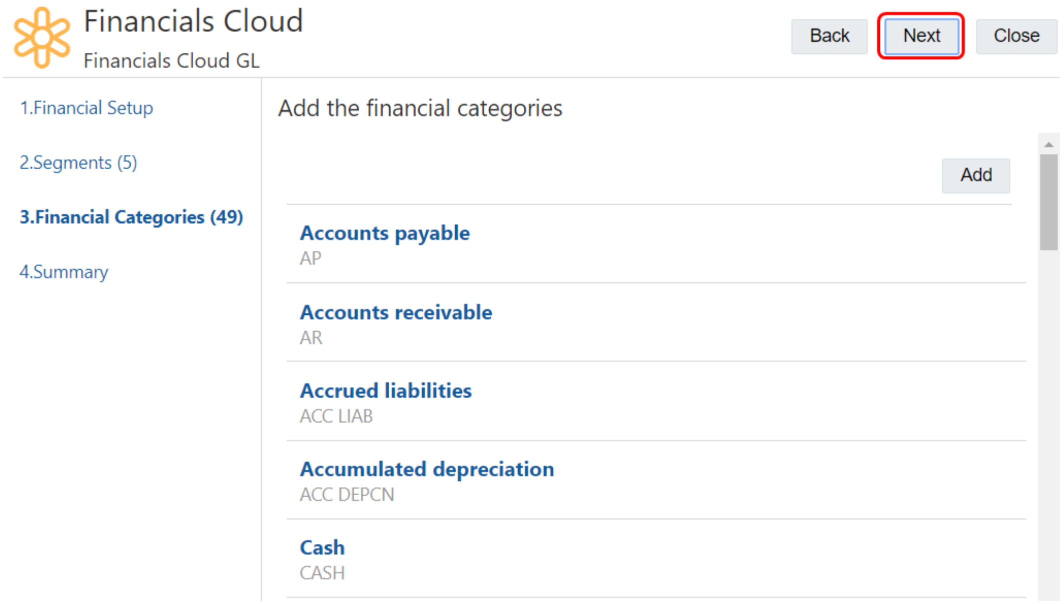
Corporate Cost Center
Corporate Company

Corporate LoB
Corporate Line of Business

Corporate Product
Corporate Product

14. On the Financial Categories page, click **Next**.

> **Note:**
>
> The Financial Categories page lists the default financial categories in Oracle Financials Cloud General Ledger. If you have modified the financial categories in your Oracle Financials Cloud General Ledger service, you can change the settings here to match the modified values.



Accounts payable
AP

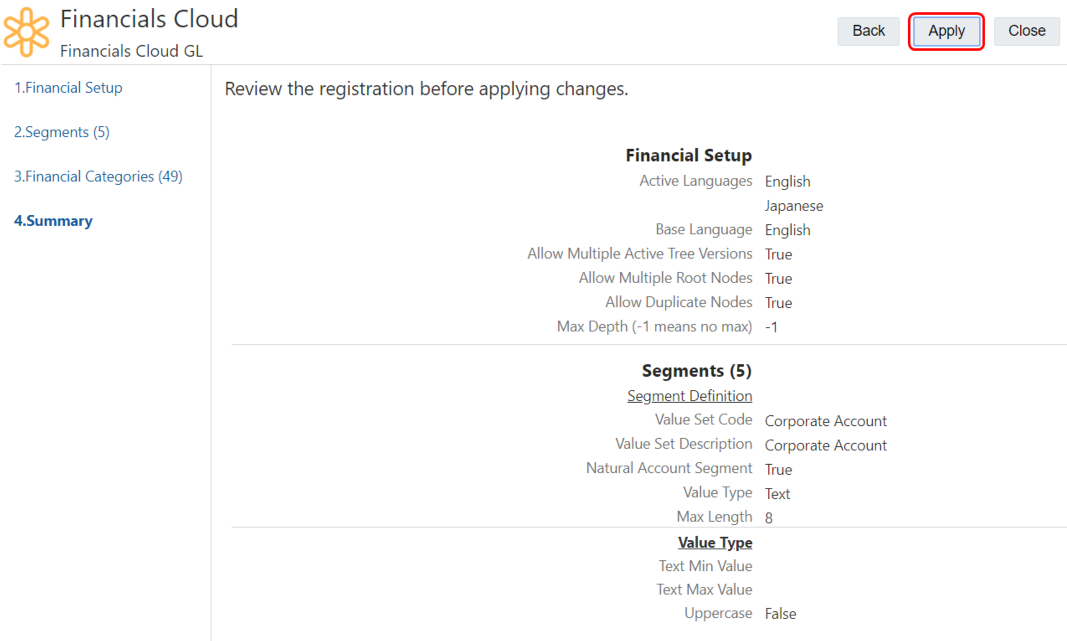
Accounts receivable
AR

Accrued liabilities
ACC LIAB

Accumulated depreciation
ACC DEPCN

Cash
CASH

15. Review the registration summary, and then click **Apply**.



Review the registration before applying changes.

Financial Setup

Active Languages	English Japanese
Base Language	English
Allow Multiple Active Tree Versions	True
Allow Multiple Root Nodes	True
Allow Duplicate Nodes	True
Max Depth (-1 means no max)	-1

Segments (5)

Segment Definition

Value Set Code	Corporate Account
Value Set Description	Corporate Account
Natural Account Segment	True
Value Type	Text
Max Length	8

Value Type

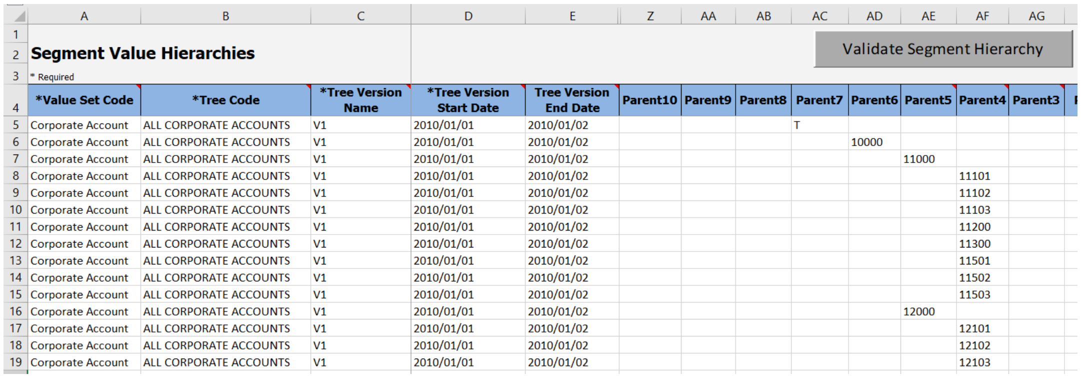
Text Min Value	
Text Max Value	
Uppercase	False

Step 2: Importing Your Oracle Financials Cloud General Ledger Value Set into Cloud EDM

Next, you import the data for the value sets that you want to manage in Oracle Fusion Cloud Enterprise Data Management. There are a few options for importing data from Oracle Financials Cloud General Ledger. This scenario will cover importing data by using the `ChartofAccountsSegmentValuesandHierarchiesImportTemplate.xlsm` template. For more information about using the template, see Import Segment Values and Hierarchies in *File-Based Data Import for Oracle Financials Cloud*

To generate the import files using the template:

1. In the `ChartofAccountsSegmentValuesandHierarchiesImportTemplate.xlsm` template, update the GL_SEGMENT_VALUES_INTERFACE and GL_SEGMENT_HIER_INTERFACE tabs with the segment values and segment value hierarchies from your Oracle Financials Cloud General Ledger value set.



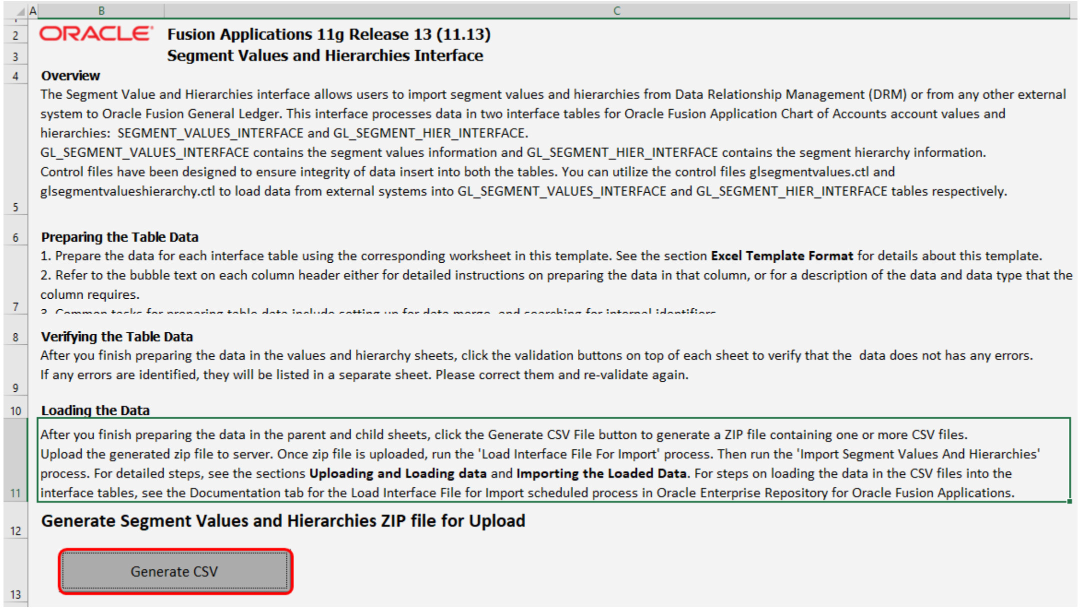
Segment Value Hierarchies

* Required

	A	B	C	D	E	Z	AA	AB	AC	AD	AE	AF	AG
4	*Value Set Code	*Tree Code	*Tree Version Name	*Tree Version Start Date	Tree Version End Date	Parent10	Parent9	Parent8	Parent7	Parent6	Parent5	Parent4	Parent3
5	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02				T				
6	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02					10000			
7	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02						11000		
8	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11101	
9	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11102	
10	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11103	
11	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11200	
12	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11300	
13	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11501	
14	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11502	
15	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							11503	
16	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02						12000		
17	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							12101	
18	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							12102	
19	Corporate Account	ALL CORPORATE ACCOUNTS	V1	2010/01/01	2010/01/02							12103	

> **Note:**
>
> Use the validation buttons on each tab to ensure that the data has no validation issues before you generate the import file.

2. On the Instructions and CSV Generation tab, click **Generate CSV**.



ORACLE **Fusion Applications 11g Release 13 (11.13)**
Segment Values and Hierarchies Interface

Overview
The Segment Value and Hierarchies interface allows users to import segment values and hierarchies from Data Relationship Management (DRM) or from any other external system to Oracle Fusion General Ledger. This interface processes data in two interface tables for Oracle Fusion Application Chart of Accounts account values and hierarchies: SEGMENT_VALUES_INTERFACE and GL_SEGMENT_HIER_INTERFACE.
GL_SEGMENT_VALUES_INTERFACE contains the segment values information and GL_SEGMENT_HIER_INTERFACE contains the segment hierarchy information.
Control files have been designed to ensure integrity of data insert into both the tables. You can utilize the control files glsegmentvalues.ctl and glsegmentvalueshierarchy.ctl to load data from external systems into GL_SEGMENT_VALUES_INTERFACE and GL_SEGMENT_HIER_INTERFACE tables respectively.

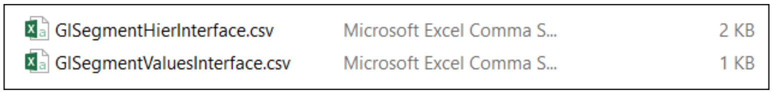
Preparing the Table Data
1. Prepare the data for each interface table using the corresponding worksheet in this template. See the section **Excel Template Format** for details about this template.
2. Refer to the bubble text on each column header either for detailed instructions on preparing the data in that column, or for a description of the data and data type that the column requires.
3. Common tasks for preparing table data include setting up for data merge, and searching for internal identifiers

Verifying the Table Data
After you finish preparing the data in the values and hierarchy sheets, click the validation buttons on top of each sheet to verify that the data does not has any errors.
If any errors are identified, they will be listed in a separate sheet. Please correct them and re-validate again.

Loading the Data
After you finish preparing the data in the parent and child sheets, click the Generate CSV File button to generate a ZIP file containing one or more CSV files.
Upload the generated zip file to server. Once zip file is uploaded, run the 'Load Interface File For Import' process. Then run the 'Import Segment Values And Hierarchies' process. For detailed steps, see the sections **Uploading and Loading data** and **Importing the Loaded Data**. For steps on loading the data in the CSV files into the interface tables, see the Documentation tab for the Load Interface File for Import scheduled process in Oracle Enterprise Repository for Oracle Fusion Applications.

Generate Segment Values and Hierarchies ZIP file for Upload

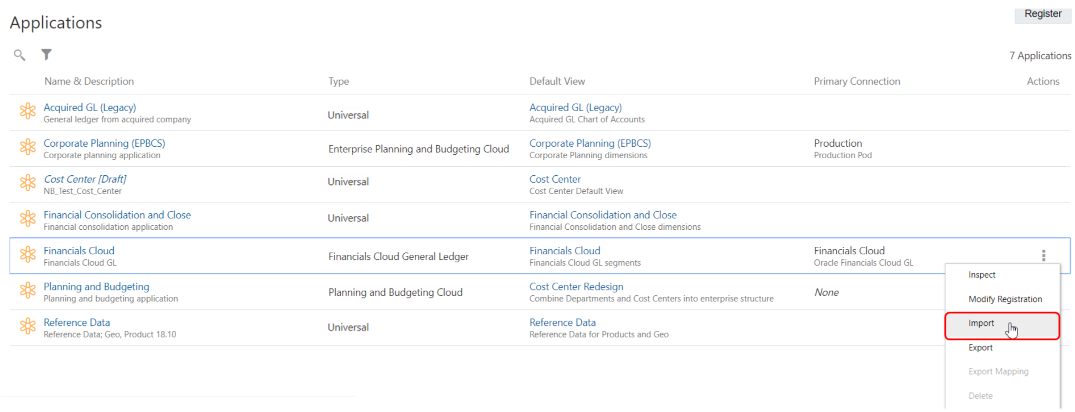
3. Verify that the template created a `glsegmentvaluesandhierarchy.zip` containing two CSV files:
   - `GlSegmentValuesInterface.csv`
   - `GlSegmentHierInterface.csv`



GlSegmentHierInterface.csv	Microsoft Excel Comma S...	2 KB
GlSegmentValuesInterface.csv	Microsoft Excel Comma S...	1 KB

To import the generated files:

1. From **Applications**, scroll to Financials Cloud, click ⋮, and then select **Import**.

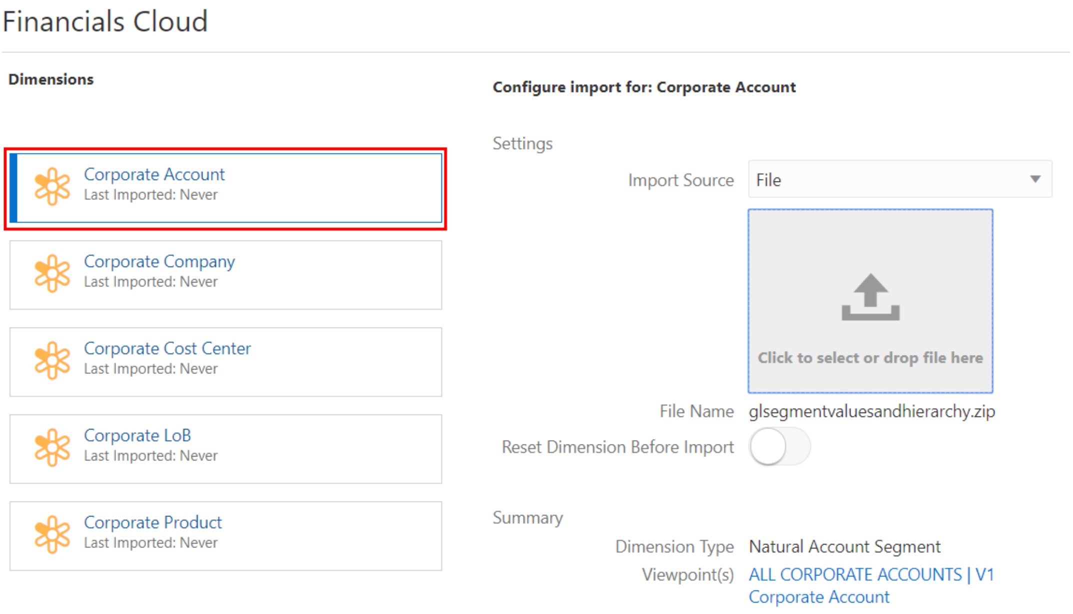


Applications

7 Applications

Name & Description	Type	Default View	Primary Connection	Actions
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts		
Corporate Planning (EPBCS) Corporate planning application	Enterprise Planning and Budgeting Cloud	Corporate Planning (EPBCS) Corporate Planning dimensions	Production Production Pod	
Cost Center [Draft] NB_Test_Cost_Center	Universal	Cost Center Cost Center Default View		
Financial Consolidation and Close Financial consolidation application	Universal	Financial Consolidation and Close Financial Consolidation and Close dimensions		
Financials Cloud Financials Cloud GL	Financials Cloud General Ledger	Financials Cloud Financials Cloud GL segments	Financials Cloud Oracle Financials Cloud GL	⋮
Planning and Budgeting Planning and budgeting application	Planning and Budgeting Cloud	Cost Center Redesign Combine Departments and Cost Centers into enterprise structure	*None*	
Reference Data Reference Data; Geo, Product 18.10	Universal	Reference Data Reference Data for Products and Geo		

2. In the left column, select the dimension that you want to import data into.



Financials Cloud

Dimensions

- Corporate Account
  Last Imported: Never
- Corporate Company
  Last Imported: Never
- Corporate Cost Center
  Last Imported: Never
- Corporate LoB
  Last Imported: Never
- Corporate Product
  Last Imported: Never

Configure import for: Corporate Account

Settings

Import Source: File

File Name: glsegmentvaluesandhierarchy.zip

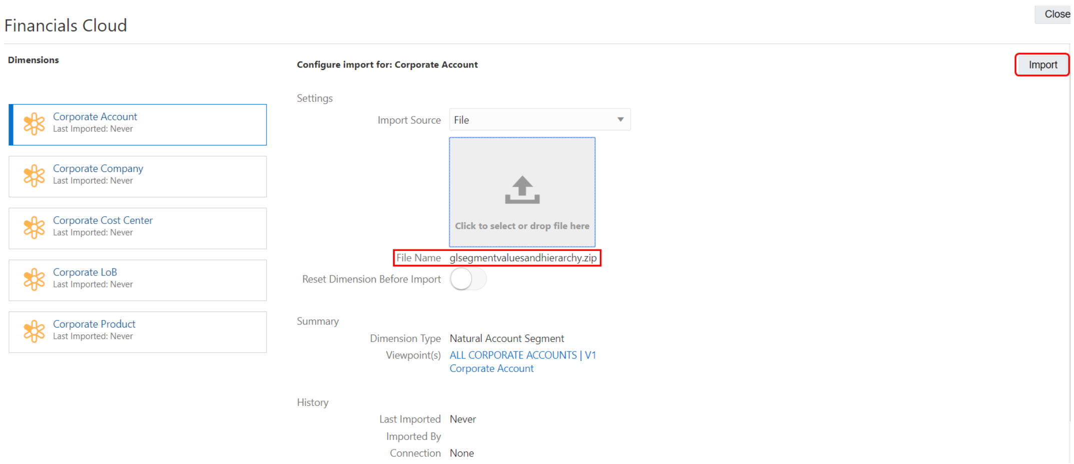
Reset Dimension Before Import: ○

Summary

Dimension Type: Natural Account Segment

Viewpoint(s): ALL CORPORATE ACCOUNTS | V1
Corporate Account

3. In **File Name**, specify the `glsegmentvaluesandhierarchy.zip` file that you created from the template, and then click **Import**.



Financials Cloud

Dimensions

- Corporate Account
  Last Imported: Never
- Corporate Company
  Last Imported: Never
- Corporate Cost Center
  Last Imported: Never
- Corporate LoB
  Last Imported: Never
- Corporate Product
  Last Imported: Never

Configure import for: Corporate Account

Settings

Import Source: File

File Name: glsegmentvaluesandhierarchy.zip

Reset Dimension Before Import: ○

Summary

Dimension Type: Natural Account Segment

Viewpoint(s): ALL CORPORATE ACCOUNTS | V1
Corporate Account

History

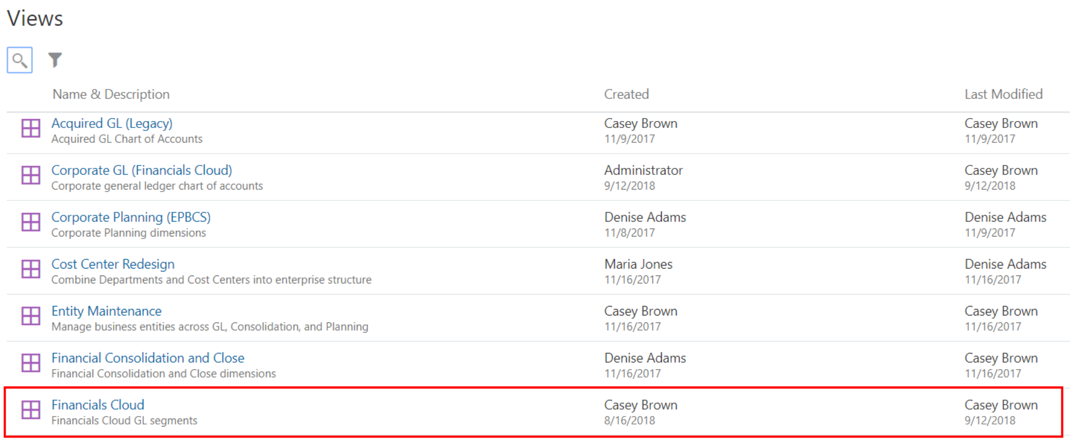
Last Imported: Never

Imported By:

Connection: None

To verify the data was imported successfully:

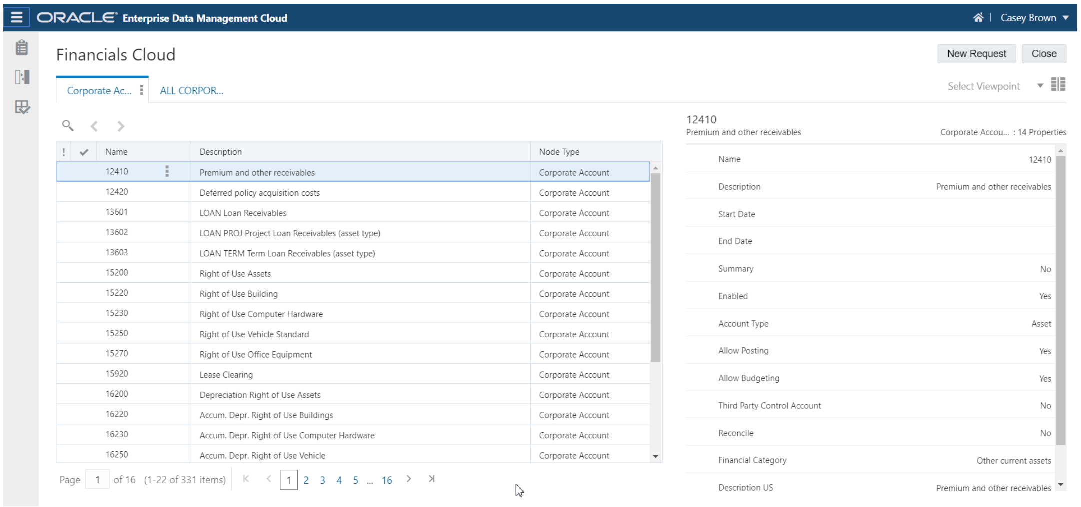
1. From **Views**, select the Financials Cloud default view that was created when you created the application.



Views

Name & Description	Created	Last Modified
Acquired GL (Legacy) Acquired GL Chart of Accounts	Casey Brown 11/9/2017	Casey Brown 11/9/2017
Corporate GL (Financials Cloud) Corporate general ledger chart of accounts	Administrator 9/12/2018	Casey Brown 9/12/2018
Corporate Planning (EPBCS) Corporate Planning dimensions	Denise Adams 11/8/2017	Denise Adams 11/9/2017
Cost Center Redesign Combine Departments and Cost Centers into enterprise structure	Maria Jones 11/16/2017	Denise Adams 11/16/2017
Entity Maintenance Manage business entities across GL, Consolidation, and Planning	Casey Brown 11/16/2017	Casey Brown 11/16/2017
Financial Consolidation and Close Financial Consolidation and Close dimensions	Denise Adams 11/16/2017	Casey Brown 11/16/2017
Financials Cloud Financials Cloud GL segments	Casey Brown 8/16/2018	Casey Brown 9/12/2018

2. Verify that the Corporate Account viewpoint contains the nodes in a list format.



Financials Cloud

Corporate Ac... ALL CORPOR...

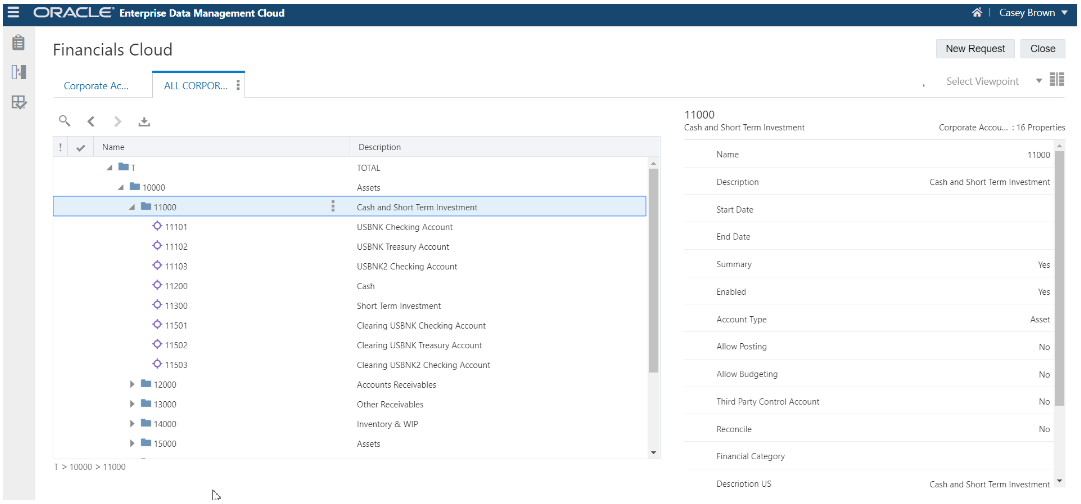
Name	Description	Node Type
12410	Premium and other receivables	Corporate Account
12420	Deferred policy acquisition costs	Corporate Account
13601	LOAN Loan Receivables	Corporate Account
13602	LOAN PROJ Project Loan Receivables (asset type)	Corporate Account
13603	LOAN TERM Term Loan Receivables (asset type)	Corporate Account
15200	Right of Use Assets	Corporate Account
15220	Right of Use Building	Corporate Account
15230	Right of Use Computer Hardware	Corporate Account
15250	Right of Use Vehicle Standard	Corporate Account
15270	Right of Use Office Equipment	Corporate Account
15920	Lease Clearing	Corporate Account
16200	Depreciation Right of Use Assets	Corporate Account
16220	Accum. Depr. Right of Use Buildings	Corporate Account
16230	Accum. Depr. Right of Use Computer Hardware	Corporate Account
16250	Accum. Depr. Right of Use Vehicle	Corporate Account

Page 1 of 16 (1-22 of 331 items) 1 2 3 4 5 ... 16

12410
Premium and other receivables
Corporate Accou... : 14 Properties

Name	12410
Description	Premium and other receivables
Start Date	
End Date	
Summary	No
Enabled	Yes
Account Type	Asset
Allow Posting	Yes
Allow Budgeting	Yes
Third Party Control Account	No
Reconcile	No
Financial Category	Other current assets
Description US	Premium and other receivables

3. Verify that the ALL CORPORATE ACCOUNTS viewpoint contains the node hierarchy.



Financials Cloud

Name	Description
T	TOTAL
10000	Assets
11000	Cash and Short Term Investment
11101	USBNK Checking Account
11102	USBNK Treasury Account
11103	USBNK2 Checking Account
11200	Cash
11300	Short Term Investment
11501	Clearing USBNK Checking Account
11502	Clearing USBNK Treasury Account
11503	Clearing USBNK2 Checking Account
12000	Accounts Receivables
13000	Other Receivables
14000	Inventory & WIP
15000	Assets

T > 10000 > 11000

11000
Cash and Short Term Investment
Corporate Accou... : 16 Properties

Name	11000
Description	Cash and Short Term Investment
Start Date	
End Date	
Summary	Yes
Enabled	Yes
Account Type	Asset
Allow Posting	No
Allow Budgeting	No
Third Party Control Account	No
Reconcile	No
Financial Category	
Description US	Cash and Short Term Investment

After you have imported and verified the data, you must assign permissions to your users. In this scenario, the user steps involve adding a node and exporting the data back to Oracle Financials Cloud General Ledger. Users must have *Data Manager* permission on a dimension to export data from that dimension. See Working with Permissions.

Cloud EDM User Steps

Cloud EDM User Prerequisites

You must have *Data Manager* permission on the Corporate Account dimension to export data from the dimension.

Cloud EDM User Process

The Financials Cloud application contains accounts for:

- `11101: USBNK Checking Account`
- `11102: USBNK Treasury Account`
- `11103: USBNK2 Checking Account`

In this scenario, you add account `11104: USBNK2 Treasury Account` and then export the change back to Oracle Financials Cloud General Ledger.

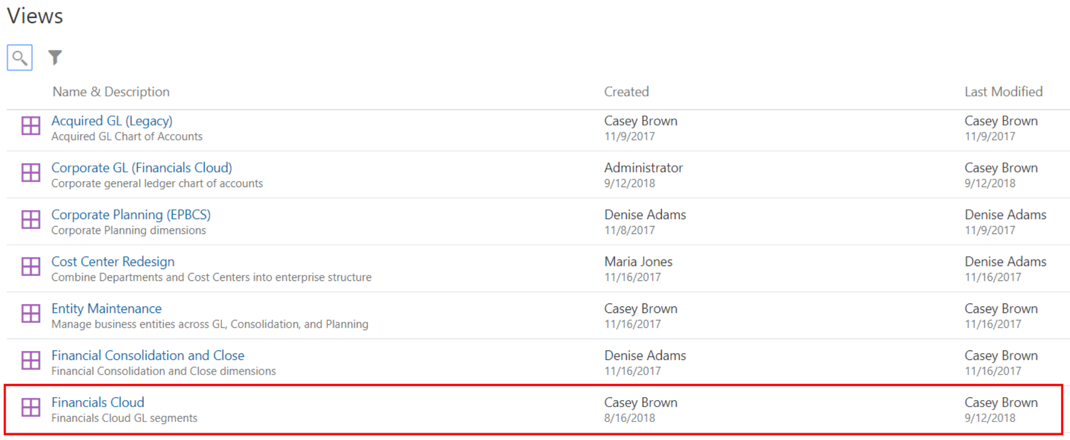
The procedure involves:

1. Adding a new account in Oracle Fusion Cloud Enterprise Data Management. See Adding a Node to a Hierarchy.
2. Exporting the modified chart of accounts data back to Oracle Financials Cloud General Ledger. See Exporting Oracle Financials Cloud General Ledger Dimensions.

Cloud EDM User Detailed Steps

To add a node:

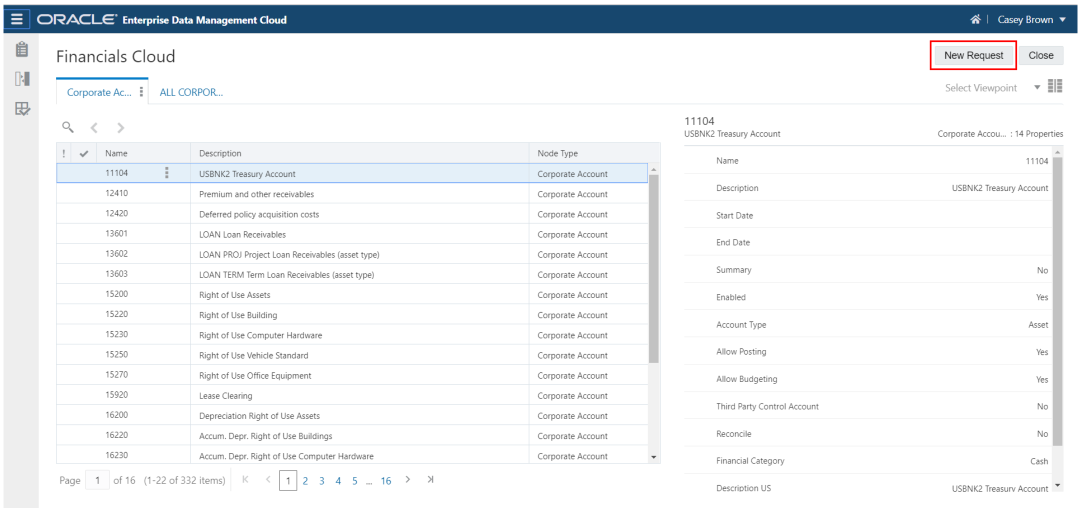
1. From **Views**, click the Financials Cloud default view.



Views

	Name & Description	Created	Last Modified
⊞	Acquired GL (Legacy) Acquired GL Chart of Accounts	Casey Brown 11/9/2017	Casey Brown 11/9/2017
⊞	Corporate GL (Financials Cloud) Corporate general ledger chart of accounts	Administrator 9/12/2018	Casey Brown 9/12/2018
⊞	Corporate Planning (EPBCS) Corporate Planning dimensions	Denise Adams 11/8/2017	Denise Adams 11/9/2017
⊞	Cost Center Redesign Combine Departments and Cost Centers into enterprise structure	Maria Jones 11/16/2017	Denise Adams 11/16/2017
⊞	Entity Maintenance Manage business entities across GL, Consolidation, and Planning	Casey Brown 11/16/2017	Casey Brown 11/16/2017
⊞	Financial Consolidation and Close Financial Consolidation and Close dimensions	Denise Adams 11/16/2017	Casey Brown 11/16/2017
⊞	Financials Cloud Financials Cloud GL segments	Casey Brown 8/16/2018	Casey Brown 9/12/2018

2. Click **New Request**.



Financials Cloud

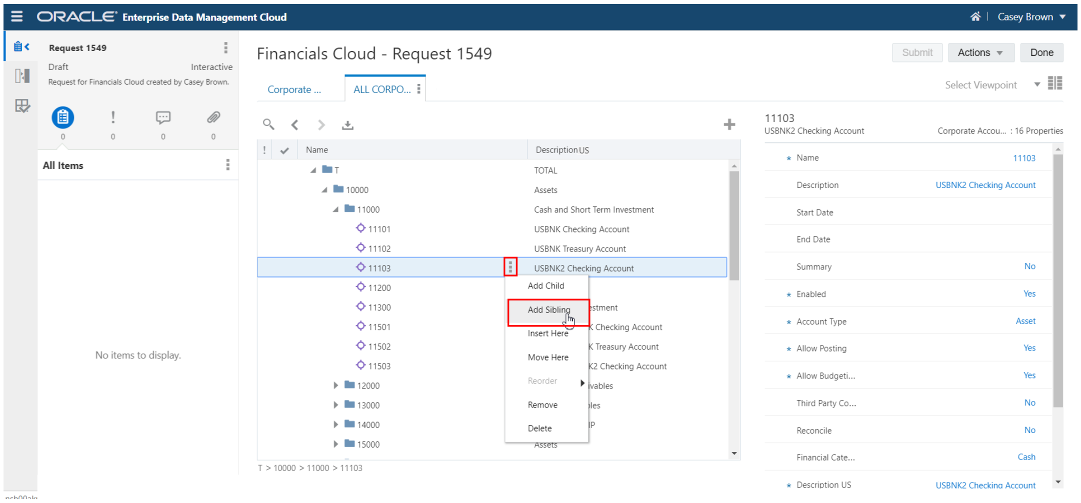
!	✓	Name	Description	Node Type
		11104	USBNK2 Treasury Account	Corporate Account
		12410	Premium and other receivables	Corporate Account
		12420	Deferred policy acquisition costs	Corporate Account
		13601	LOAN Loan Receivables	Corporate Account
		13602	LOAN PROJ Project Loan Receivables (asset type)	Corporate Account
		13603	LOAN TERM Term Loan Receivables (asset type)	Corporate Account
		15200	Right of Use Assets	Corporate Account
		15220	Right of Use Building	Corporate Account
		15230	Right of Use Computer Hardware	Corporate Account
		15250	Right of Use Vehicle Standard	Corporate Account
		15270	Right of Use Office Equipment	Corporate Account
		15920	Lease Clearing	Corporate Account
		16200	Depreciation Right of Use Assets	Corporate Account
		16220	Accum. Depr. Right of Use Buildings	Corporate Account
		16230	Accum. Depr. Right of Use Computer Hardware	Corporate Account

Page 1 of 16 (1-22 of 332 items) 1 2 3 4 5 ... 16

11104
USBNK2 Treasury Account
Corporate Accou... : 14 Properties

Name	11104
Description	USBNK2 Treasury Account
Start Date	
End Date	
Summary	No
Enabled	Yes
Account Type	Asset
Allow Posting	Yes
Allow Budgeting	Yes
Third Party Control Account	No
Reconcile	No
Financial Category	Cash
Description US	USBNK2 Treasury Account

3. In the ALL CORPORATE ACCOUNTS viewpoint, navigate to node `11103`, click ⋮, and select **Add Sibling**.



Request 1549
Draft Interactive
Request for Financials Cloud created by Casey Brown.

All Items

No items to display.

Financials Cloud - Request 1549

!	✓	Name	Description US
		T	TOTAL
		10000	Assets
		11000	Cash and Short Term Investment
		11101	USBNK Checking Account
		11102	USBNK Treasury Account
		11103	USBNK2 Checking Account
		11200	
		11300	estment
		11501	K Checking Account
		11502	K Treasury Account
		11503	K2 Checking Account
		12000	ivables
		13000	les
		14000	IP
		15000	Assets

T > 10000 > 11000 > 11103

11103
USBNK2 Checking Account
Corporate Accou... : 16 Properties

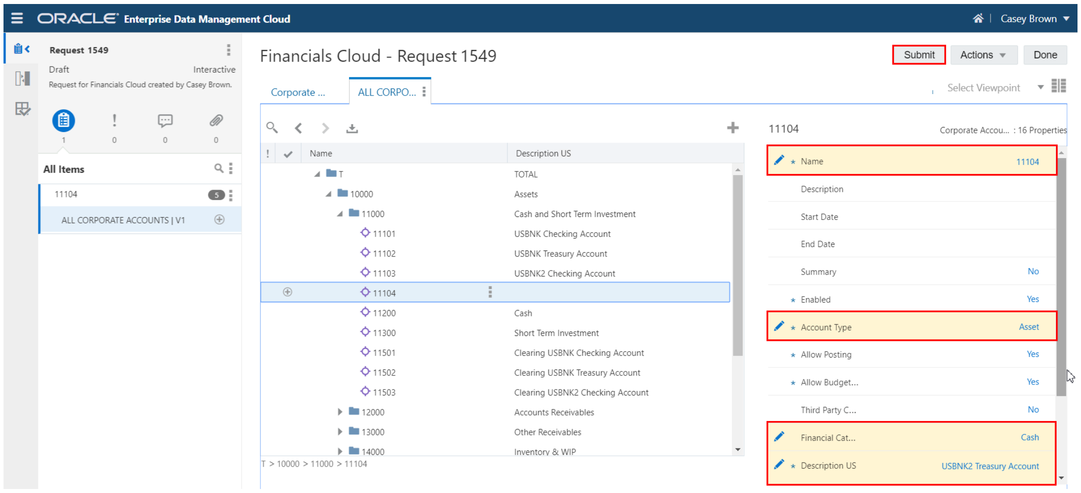
* Name	11103
Description	USBNK2 Checking Account
Start Date	
End Date	
Summary	No
* Enabled	Yes
* Account Type	Asset
* Allow Posting	Yes
* Allow Budgeti...	Yes
Third Party Co...	No
Reconcile	No
Financial Cate...	Cash
* Description US	USBNK2 Checking Account

4. In the properties panel, enter these properties and then click **Submit**.

   - **Name**: 11104
   - **Description US**: USBNK2 Treasury Account

   > **Note:**
   >
   > Add the description property for your base language (for example, **Description US**) to the displayed columns to view the description as shown in the images in this procedure.

   - **Account Type**: Asset
   - **Financial Category**: Cash


ORACLE Enterprise Data Management Cloud
Casey Brown
Request 1549
Draft
Interactive
Request for Financials Cloud created by Casey Brown.
All Items
11104
5
ALL CORPORATE ACCOUNTS | V1
Financials Cloud - Request 1549
Submit
Actions
Done
Corporate ...
ALL CORPO...
Select Viewpoint

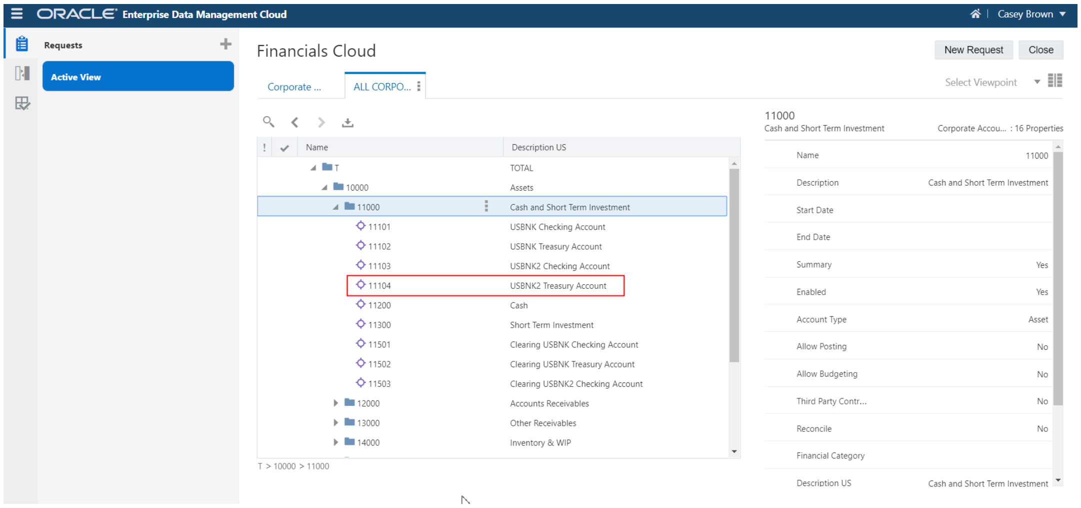
Name	Description US
T	TOTAL
10000	Assets
11000	Cash and Short Term Investment
11101	USBNK Checking Account
11102	USBNK Treasury Account
11103	USBNK2 Checking Account
11104	
11200	Cash
11300	Short Term Investment
11501	Clearing USBNK Checking Account
11502	Clearing USBNK Treasury Account
11503	Clearing USBNK2 Checking Account
12000	Accounts Receivables
13000	Other Receivables
14000	Inventory & WIP

T > 10000 > 11000 > 11104

11104 Corporate Accou... : 16 Properties

Name	11104
Description	
Start Date	
End Date	
Summary	No
Enabled	Yes
Account Type	Asset
Allow Posting	Yes
Allow Budget...	Yes
Third Party C...	No
Financial Cat...	Cash
Description US	USBNK2 Treasury Account

5. Verify that the node was added in the viewpoint.



Name	Description US
T	TOTAL
10000	Assets
11000	Cash and Short Term Investment
11101	USBNK Checking Account
11102	USBNK Treasury Account
11103	USBNK2 Checking Account
11104	USBNK2 Treasury Account
11200	Cash
11300	Short Term Investment
11501	Clearing USBNK Checking Account
11502	Clearing USBNK Treasury Account
11503	Clearing USBNK2 Checking Account
12000	Accounts Receivables
13000	Other Receivables
14000	Inventory & WIP

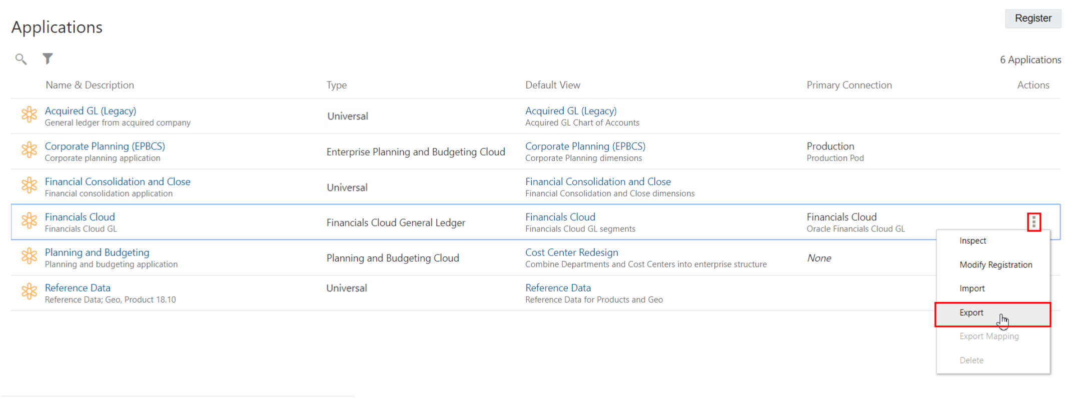
T > 10000 > 11000

11000
Cash and Short Term Investment Corporate Accou... : 16 Properties

Name	11000
Description	Cash and Short Term Investment
Start Date	
End Date	
Summary	Yes
Enabled	Yes
Account Type	Asset
Allow Posting	No
Allow Budgeting	No
Third Party Contr...	No
Reconcile	No
Financial Category	
Description US	Cash and Short Term Investment

To export the dimension:

1. From **Applications**, scroll to Financials Cloud, click ⋮, and then select **Export**.

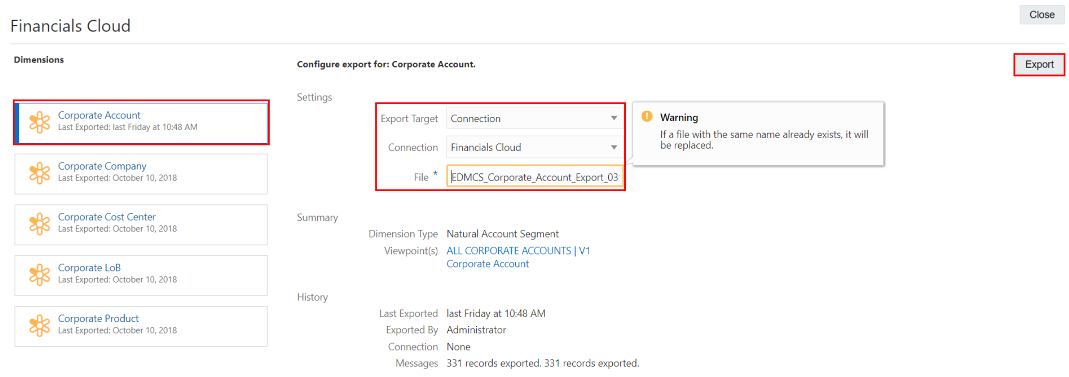


Applications

6 Applications

Name & Description	Type	Default View	Primary Connection	Actions
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts		
Corporate Planning (EPBCS) Corporate planning application	Enterprise Planning and Budgeting Cloud	Corporate Planning (EPBCS) Corporate Planning dimensions	Production Production Pod	
Financial Consolidation and Close Financial consolidation application	Universal	Financial Consolidation and Close Financial Consolidation and Close dimensions		
Financials Cloud Financials Cloud GL	Financials Cloud General Ledger	Financials Cloud Financials Cloud GL segments	Financials Cloud Oracle Financials Cloud GL	⋮
Planning and Budgeting Planning and budgeting application	Planning and Budgeting Cloud	Cost Center Redesign Combine Departments and Cost Centers into enterprise structure	*None*	
Reference Data Reference Data; Geo, Product 18.10	Universal	Reference Data Reference Data for Products and Geo		

2. On the export screen, perform the following actions:
   a. Select the **Corporate Account** dimension
   b. For **Export Target**, select **Connection**.
   c. Select the **Financials Cloud** connection.
   d. Enter a file name, such as `EDMCS_Corporate_Account_Export_031819.zip`.
   e. Click **Export**.



Financials Cloud

Dimensions

Corporate Account
Last Exported: last Friday at 10:48 AM

Corporate Company
Last Exported: October 10, 2018

Corporate Cost Center
Last Exported: October 10, 2018

Corporate LoB
Last Exported: October 10, 2018

Corporate Product
Last Exported: October 10, 2018

Configure export for: Corporate Account.

Settings

Export Target: Connection
Connection: Financials Cloud
File *: EDMCS_Corporate_Account_Export_03

Warning
If a file with the same name already exists, it will be replaced.

Summary

Dimension Type: Natural Account Segment
Viewpoint(s): ALL CORPORATE ACCOUNTS | V1
Corporate Account

History

Last Exported: last Friday at 10:48 AM
Exported By: Administrator
Connection: None
Messages: 331 records exported. 331 records exported.

Oracle Financials Cloud General Ledger User Steps

Oracle Financials Cloud General Ledger User Prerequisites

You must have the Financial Application Administrator job role to import and load interface files to update segment values and hierarchies and to publish account hierarchies.

Oracle Financials Cloud General Ledger User Process

1. Importing and loading the modified chart of accounts data in Oracle Financials Cloud General Ledger. See External Data Integration Services for Oracle Cloud in *Oracle Financials Cloud Implementing Common Features for Financials*.

2. Verifying the modified chart of accounts data in Oracle Financials Cloud General Ledger. See Manage Trees and Tree Versions in *Implementing Enterprise Structures and General Ledger*
3. Publishing the modified chart of accounts data in Oracle Financials Cloud General Ledger. See Publishing an Account Hierarchy: Example in *Implementing Enterprise Structures and General Ledger*.

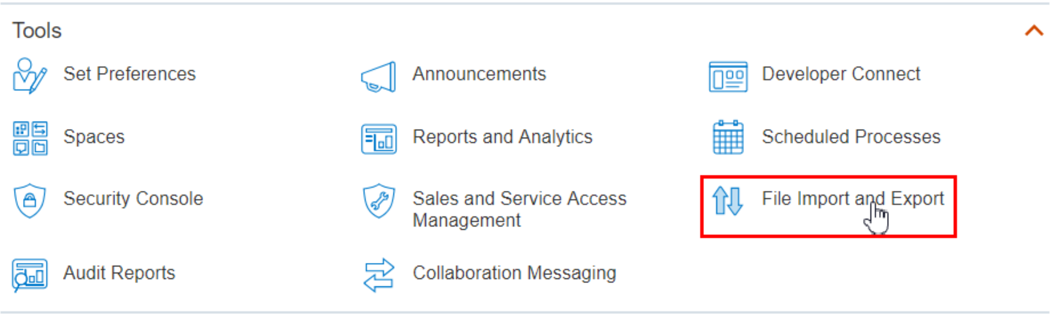
Detailed Steps

Related Topics

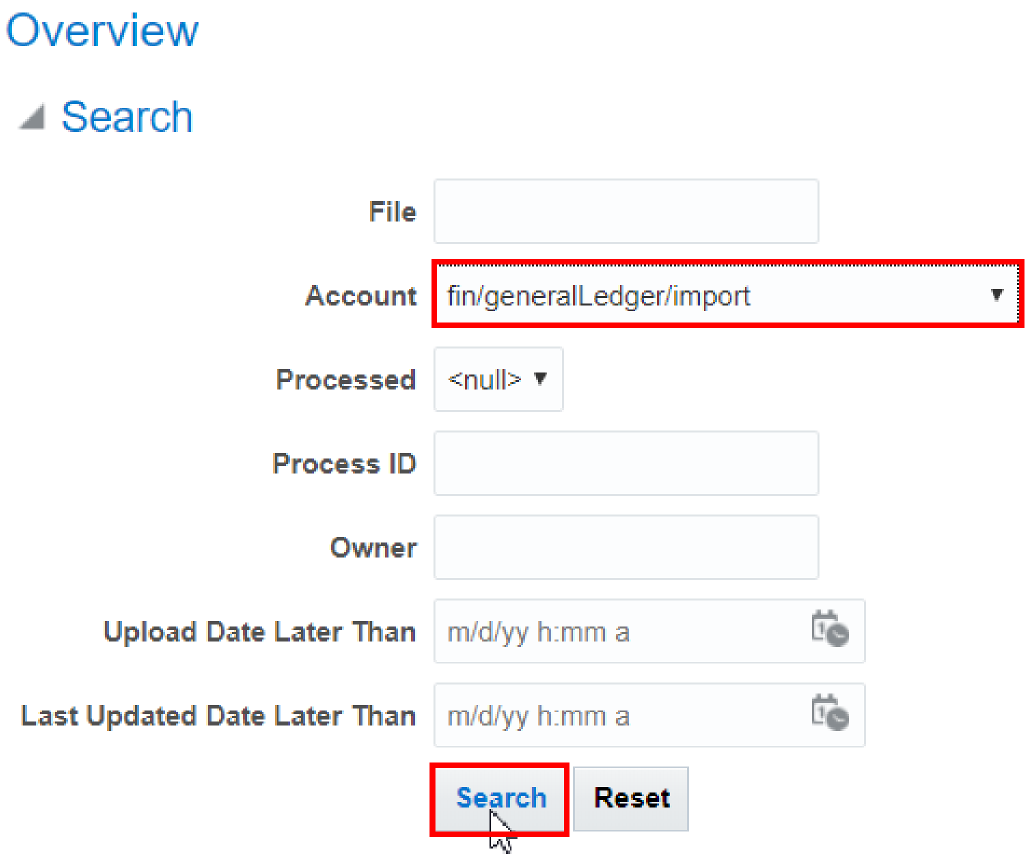
- Step 1: Importing and Loading Chart of Account Data in Oracle Financials Cloud
- Step 2: Verifying the Exported Data in Oracle Financials Cloud General Ledger
- Step 3: Publishing the Updated Oracle Financials Cloud General Ledger Hierarchy

Step 1: Importing and Loading Chart of Account Data in Oracle Financials Cloud

1. In Oracle Financials Cloud, click ☰, and then under **Tools**, select **File Import and Export**.



2. From the **Account** drop down menu, select **fin/generalLedger/import**, and then click **Search**.



Overview

Search

File:

Account: fin/generalLedger/import

Processed: <null>

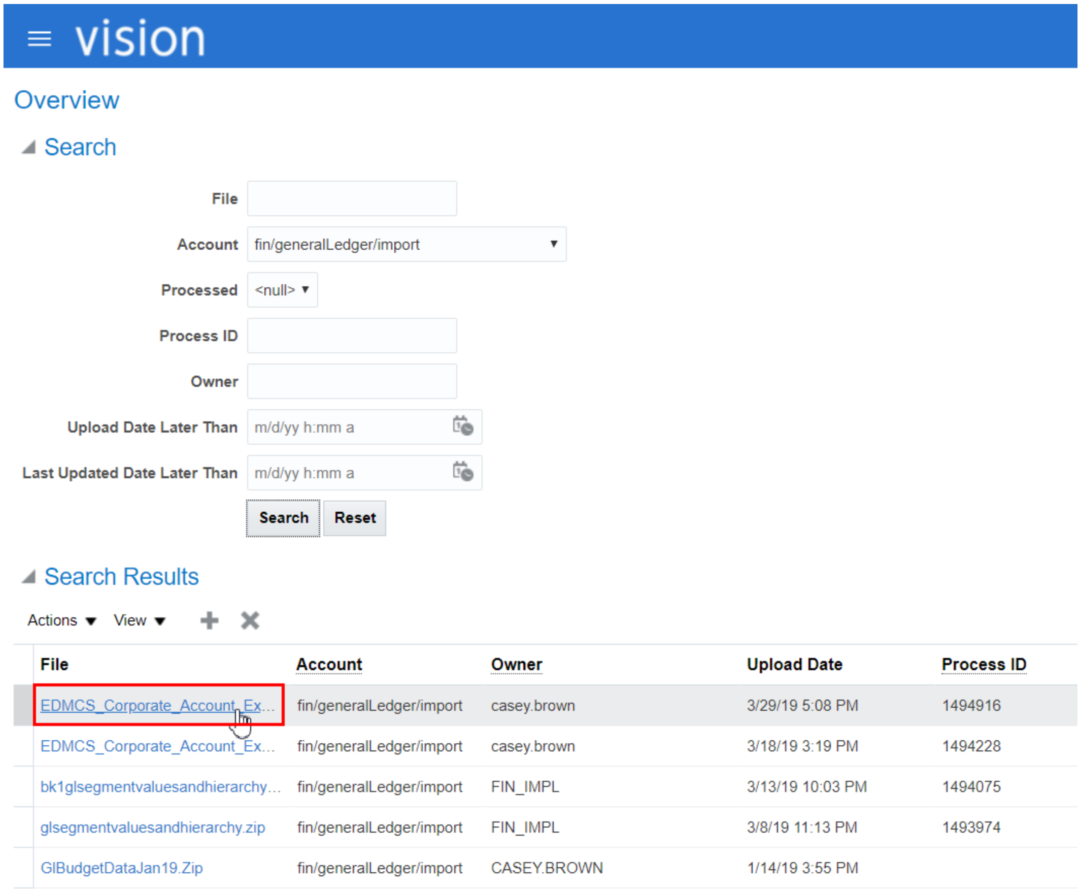
Process ID:

Owner:

Upload Date Later Than: m/d/yy h:mm a

Last Updated Date Later Than: m/d/yy h:mm a

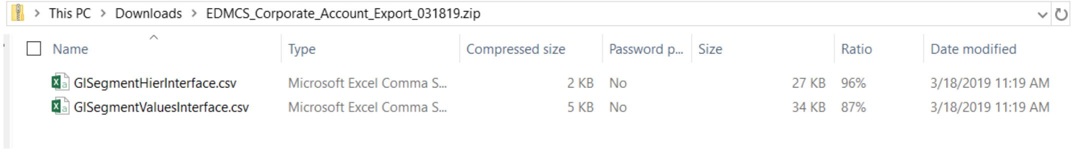
3. Click the name of the export file from Oracle Fusion Cloud Enterprise Data Management to download the CSV files.



File
Account fin/generalLedger/import
Processed <null>
Process ID
Owner
Upload Date Later Than m/d/yy h:mm a
Last Updated Date Later Than m/d/yy h:mm a

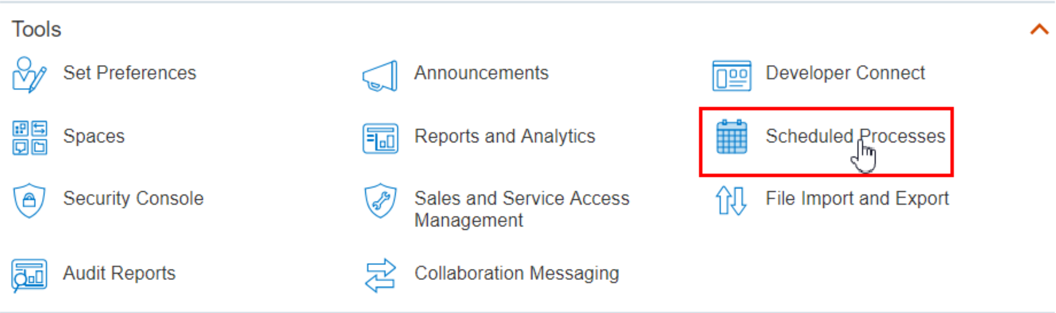
File	Account	Owner	Upload Date	Process ID
EDMCS_Corporate_Account_Ex...	fin/generalLedger/import	casey.brown	3/29/19 5:08 PM	1494916
EDMCS_Corporate_Account_Ex...	fin/generalLedger/import	casey.brown	3/18/19 3:19 PM	1494228
bk1glsegmentvaluesandhierarchy...	fin/generalLedger/import	FIN_IMPL	3/13/19 10:03 PM	1494075
glsegmentvaluesandhierarchy.zip	fin/generalLedger/import	FIN_IMPL	3/8/19 11:13 PM	1493974
GlBudgetDataJan19.Zip	fin/generalLedger/import	CASEY.BROWN	1/14/19 3:55 PM	

4. Open and verify the contents of each CSV file to ensure that the correct data will be loaded.

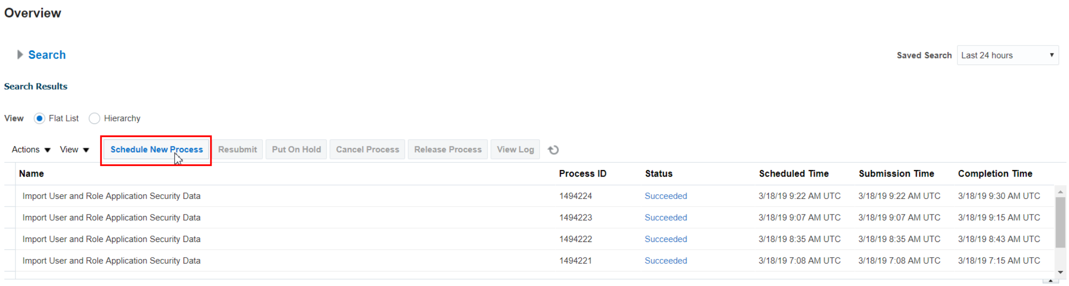

> This PC > Downloads > EDMCS_Corporate_Account_Export_031819.zip

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
GlSegmentHierInterface.csv	Microsoft Excel Comma S...	2 KB	No	27 KB	96%	3/18/2019 11:19 AM
GlSegmentValuesInterface.csv	Microsoft Excel Comma S...	5 KB	No	34 KB	87%	3/18/2019 11:19 AM

5. In Oracle Financials Cloud, click ☰, and then under **Tools**, select **Scheduled Processes**.



6. Click Schedule New Process.



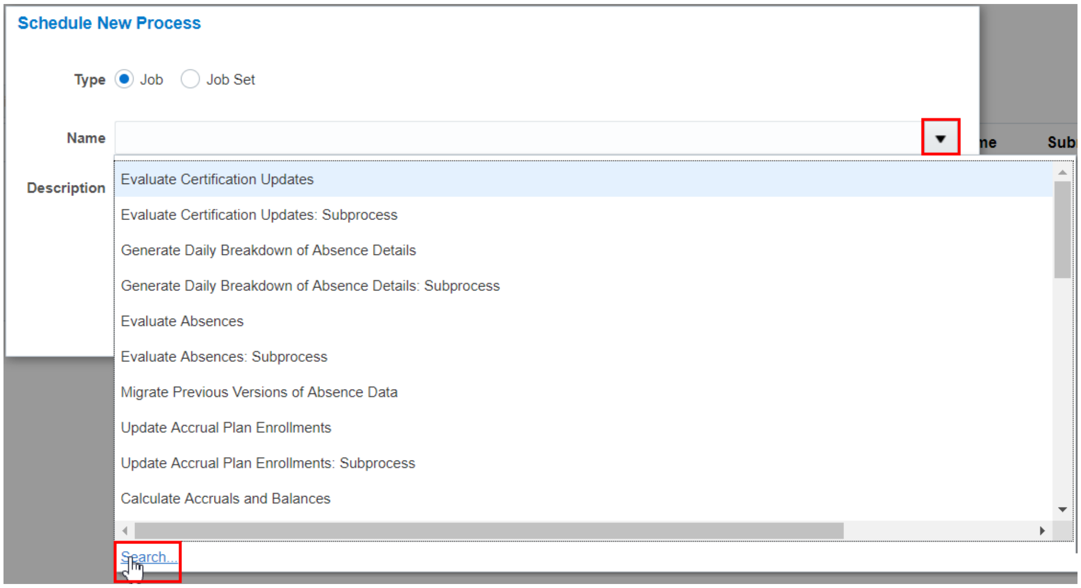
Overview

Search Results

View ● Flat List ○ Hierarchy

Name	Process ID	Status	Scheduled Time	Submission Time	Completion Time
Import User and Role Application Security Data	1494224	Succeeded	3/18/19 9:22 AM UTC	3/18/19 9:22 AM UTC	3/18/19 9:30 AM UTC
Import User and Role Application Security Data	1494223	Succeeded	3/18/19 9:07 AM UTC	3/18/19 9:07 AM UTC	3/18/19 9:15 AM UTC
Import User and Role Application Security Data	1494222	Succeeded	3/18/19 8:35 AM UTC	3/18/19 8:35 AM UTC	3/18/19 8:43 AM UTC
Import User and Role Application Security Data	1494221	Succeeded	3/18/19 7:08 AM UTC	3/18/19 7:08 AM UTC	3/18/19 7:15 AM UTC

7. Click the down arrow to the right of the **Name** field, and then click **Search**.



Schedule New Process

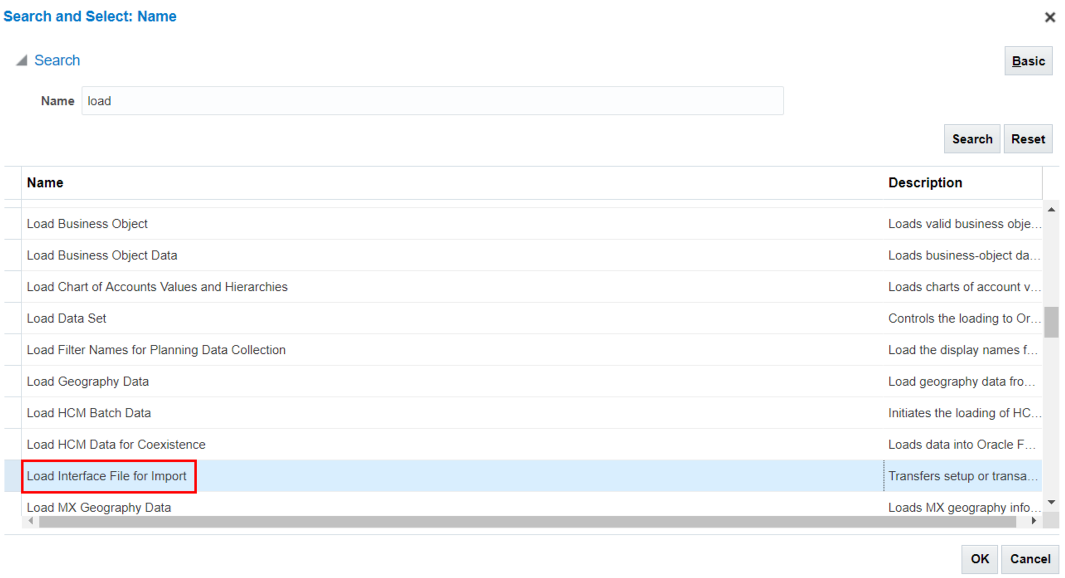
Type ● Job ○ Job Set

Name

Description

Evaluate Certification Updates
Evaluate Certification Updates: Subprocess
Generate Daily Breakdown of Absence Details
Generate Daily Breakdown of Absence Details: Subprocess
Evaluate Absences
Evaluate Absences: Subprocess
Migrate Previous Versions of Absence Data
Update Accrual Plan Enrollments
Update Accrual Plan Enrollments: Subprocess
Calculate Accruals and Balances

8. Type `load` in the **Name** field, click **Search**, and then select **Load Interface File for Import** and click **OK**.

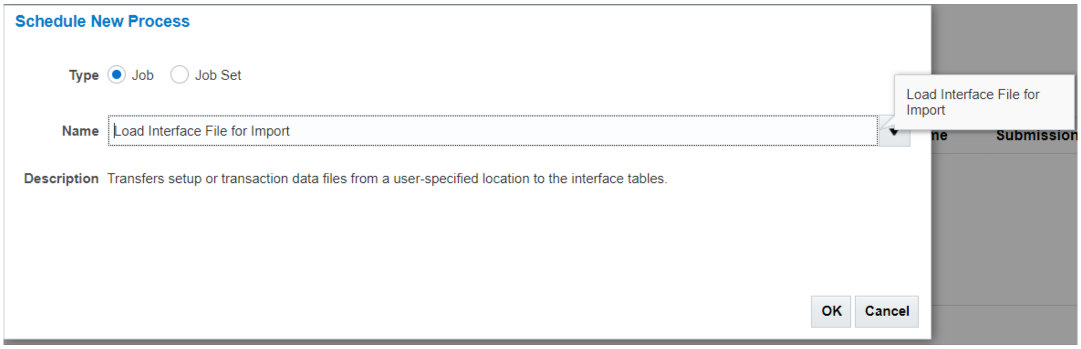


Search and Select: Name

Name: load

Name	Description
Load Business Object	Loads valid business obje...
Load Business Object Data	Loads business-object da...
Load Chart of Accounts Values and Hierarchies	Loads charts of account v...
Load Data Set	Controls the loading to Or...
Load Filter Names for Planning Data Collection	Load the display names f...
Load Geography Data	Load geography data fro...
Load HCM Batch Data	Initiates the loading of HC...
Load HCM Data for Coexistence	Loads data into Oracle F...
Load Interface File for Import	Transfers setup or transa...
Load MX Geography Data	Loads MX geography info...

9. Click **OK**.



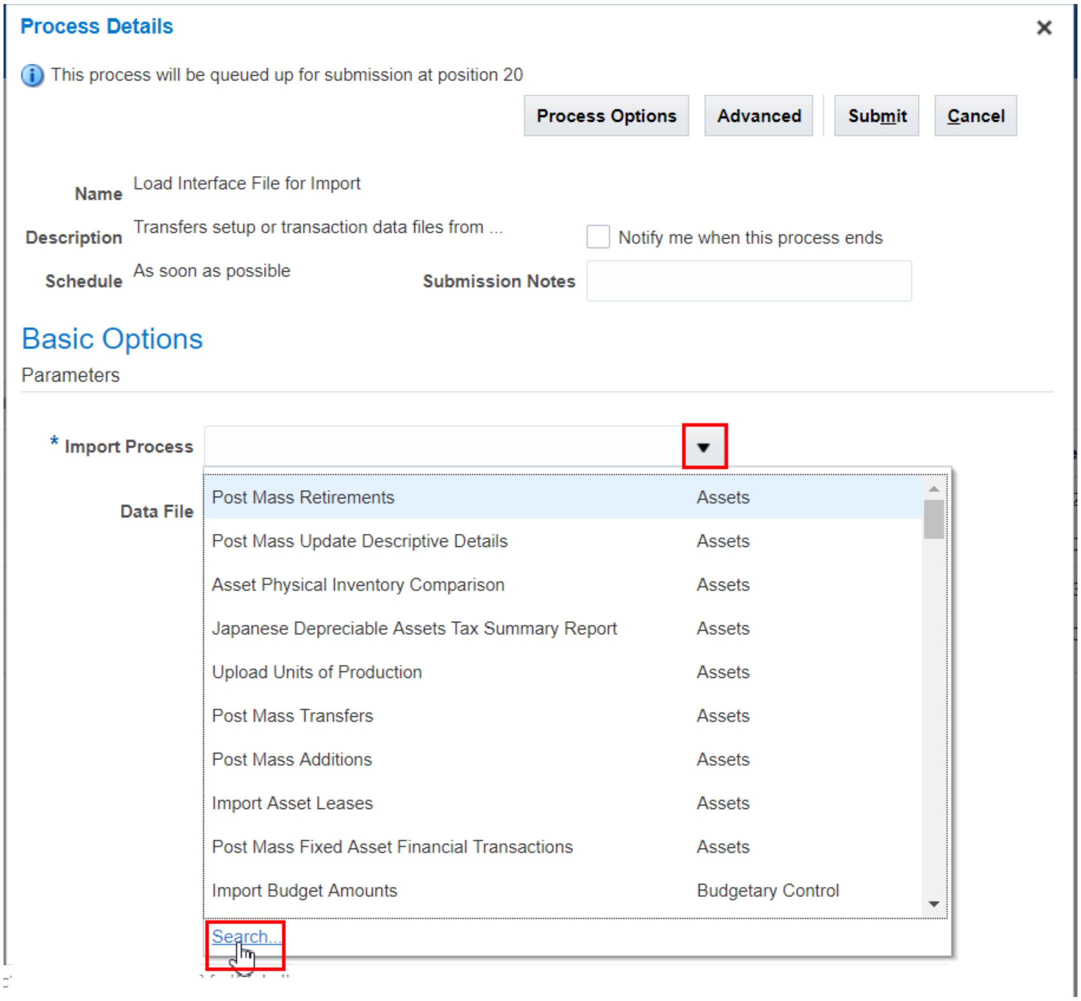
Schedule New Process

Type ● Job ○ Job Set

Name Load Interface File for Import

Description Transfers setup or transaction data files from a user-specified location to the interface tables.

10. In Process Details, click the down arrow next to **Import Process**, and then click **Search**.



Process Details

This process will be queued up for submission at position 20

Name Load Interface File for Import

Description Transfers setup or transaction data files from ...

□ Notify me when this process ends

Schedule As soon as possible

Submission Notes

Basic Options

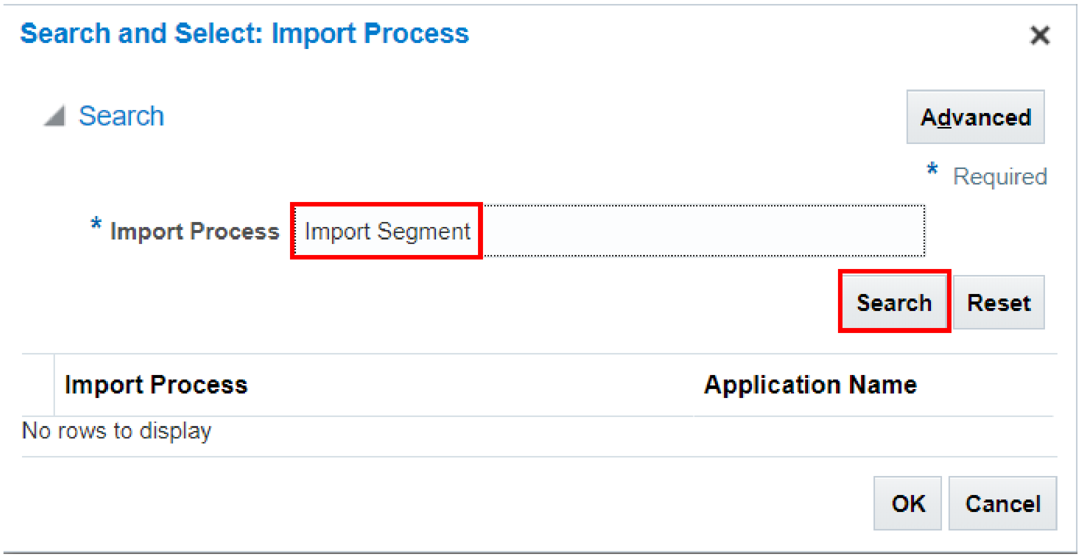
Parameters

* Import Process

Data File

Post Mass Retirements	Assets
Post Mass Update Descriptive Details	Assets
Asset Physical Inventory Comparison	Assets
Japanese Depreciable Assets Tax Summary Report	Assets
Upload Units of Production	Assets
Post Mass Transfers	Assets
Post Mass Additions	Assets
Import Asset Leases	Assets
Post Mass Fixed Asset Financial Transactions	Assets
Import Budget Amounts	Budgetary Control

11. In **Import Process**, type `Import Segment` and then click **Search**.

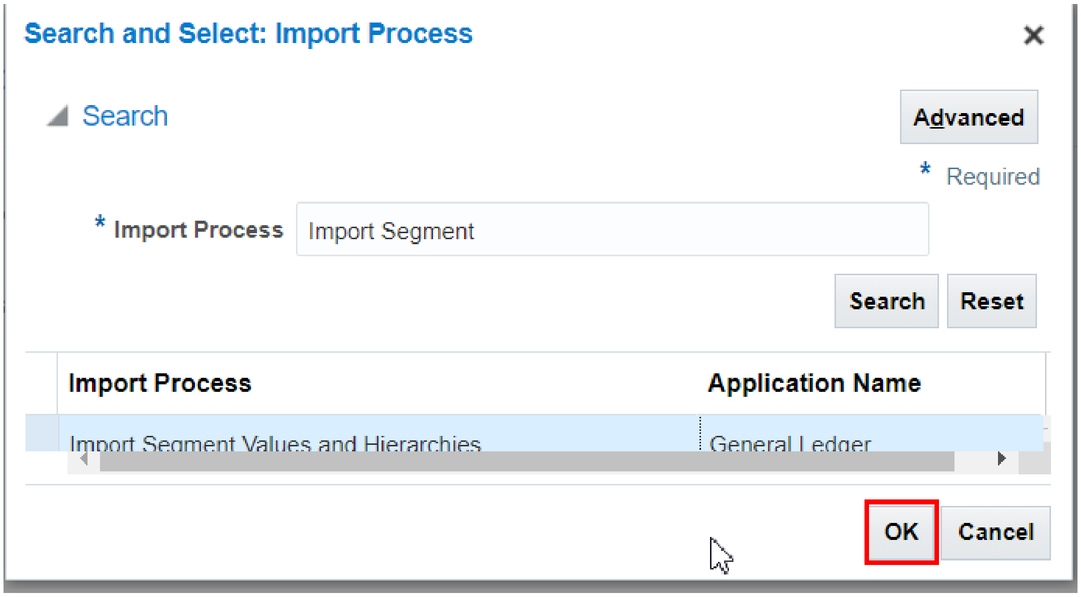


Search and Select: Import Process

* Import Process: Import Segment

Import Process	Application Name
No rows to display	

12. Select **Import Segment Values and Hierarchies**, and then click **OK**.

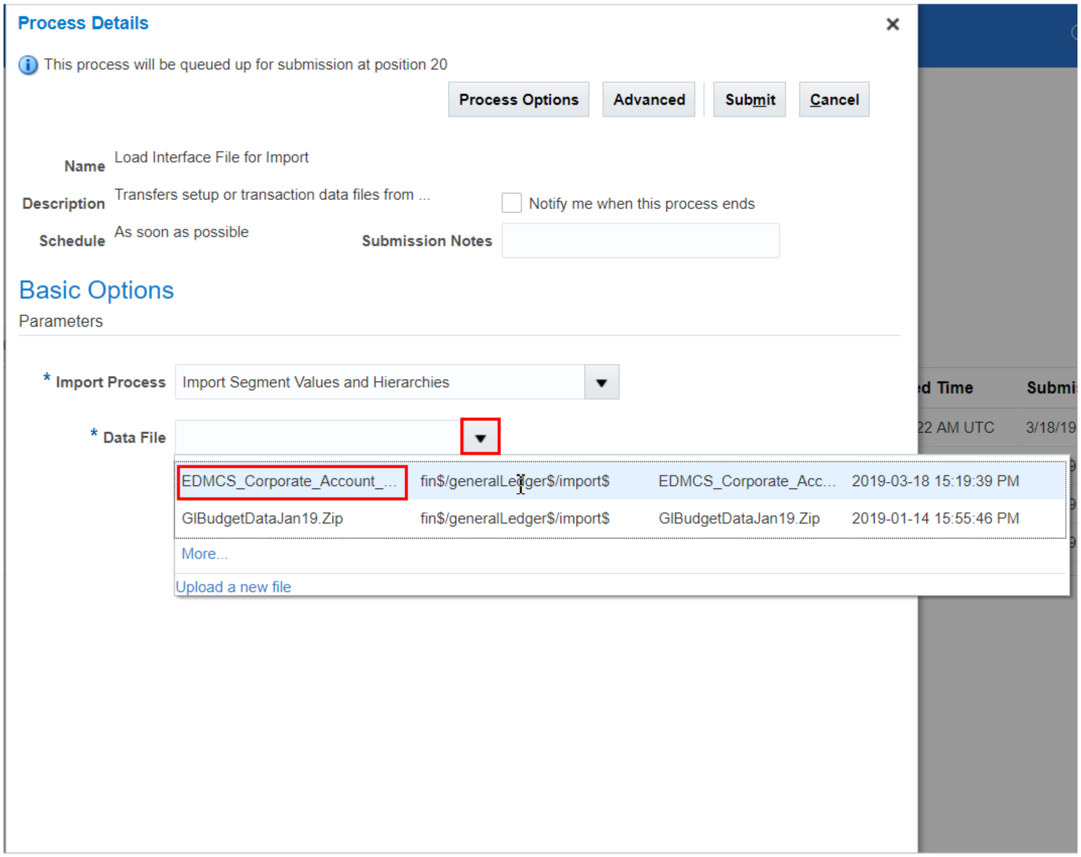


Search and Select: Import Process

* Import Process: Import Segment

Import Process	Application Name
Import Segment Values and Hierarchies	General Ledger

13. Click the down arrow next to **Data File**, and then select `EDMCS_Corporate_Account_Export_031819.zip`.



Process Details

This process will be queued up for submission at position 20

Name: Load Interface File for Import

Description: Transfers setup or transaction data files from ...

Schedule: As soon as possible

□ Notify me when this process ends

Submission Notes:

Basic Options

Parameters

* Import Process: Import Segment Values and Hierarchies

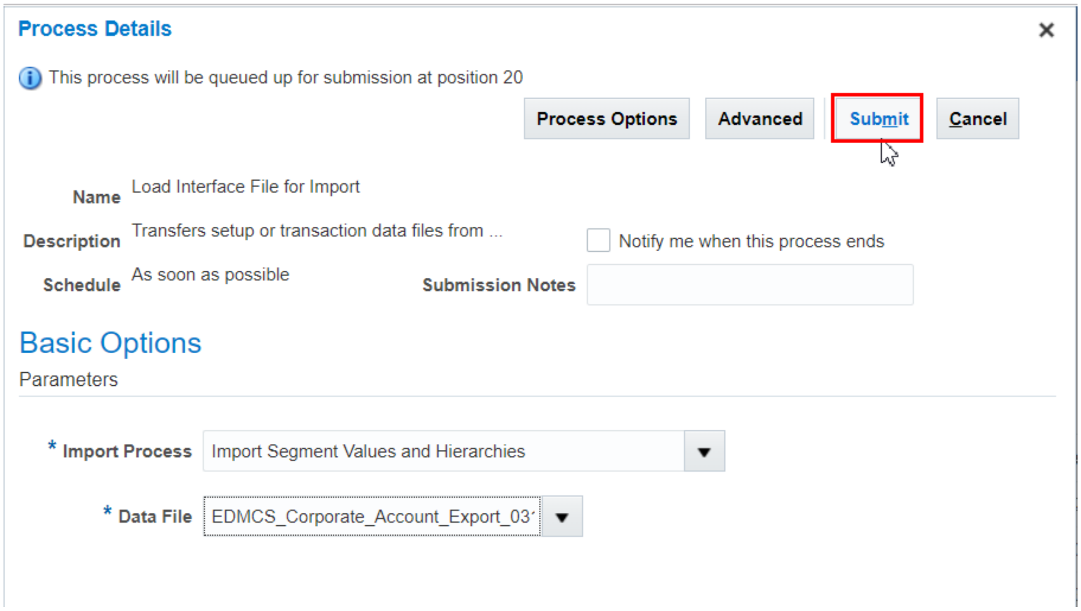
* Data File:

EDMCS_Corporate_Account_...	fin$/generalLedger$/import$	EDMCS_Corporate_Acc...	2019-03-18 15:19:39 PM
GlBudgetDataJan19.Zip	fin$/generalLedger$/import$	GlBudgetDataJan19.Zip	2019-01-14 15:55:46 PM

More...

Upload a new file

14. Click **Submit**.



Process Details

This process will be queued up for submission at position 20

Name: Load Interface File for Import

Description: Transfers setup or transaction data files from ...

Schedule: As soon as possible

□ Notify me when this process ends

Submission Notes:

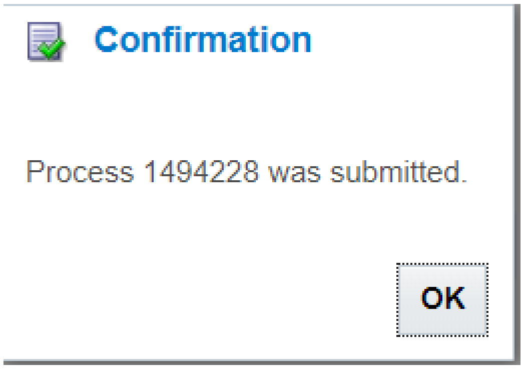
Basic Options

Parameters

* Import Process: Import Segment Values and Hierarchies

* Data File: EDMCS_Corporate_Account_Export_03'

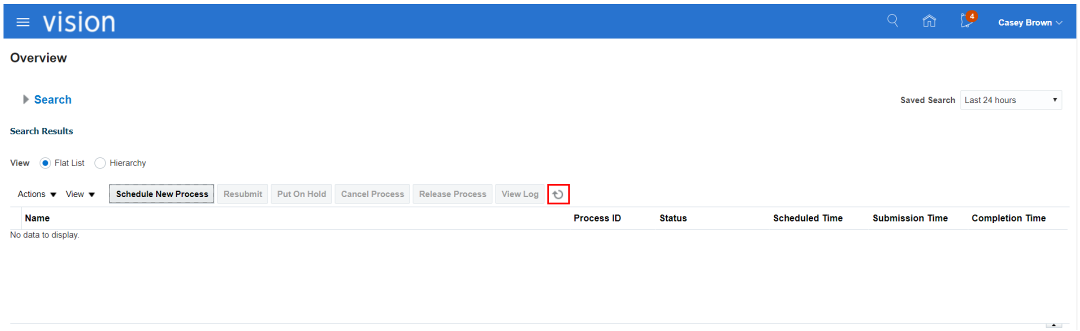
15. Click **OK** on the confirmation dialog box.



Confirmation

Process 1494228 was submitted.

16. On the **Overview** screen, click ↻ to refresh the scheduled processes.



Overview

Saved Search: Last 24 hours

Search Results

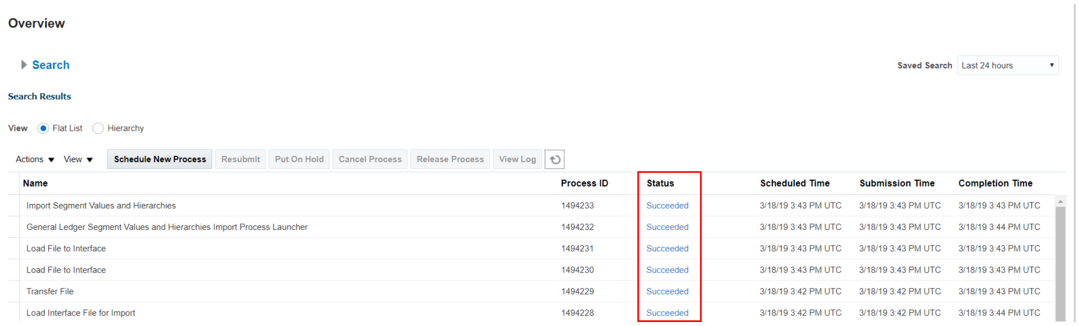
View ● Flat List ○ Hierarchy

Name	Process ID	Status	Scheduled Time	Submission Time	Completion Time

No data to display.

17. Verify that all scheduled processes for the import completed successfully. There are six processes in total:

- Load Interface File for Import
- Transfer File
- Load File to Interface
- Load File to Interface
- General Ledger Segment Values and Hierarchies Import Process Launcher
- Import Segment Values and Hierarchies



Overview

Saved Search: Last 24 hours

Search Results

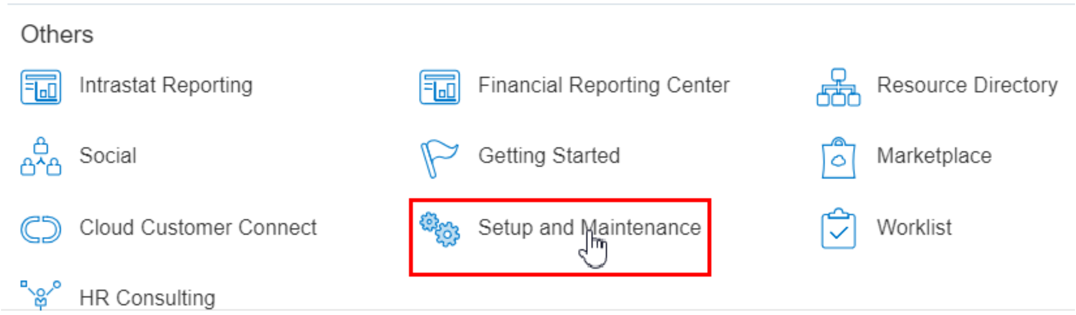
View ● Flat List ○ Hierarchy

Name	Process ID	Status	Scheduled Time	Submission Time	Completion Time
Import Segment Values and Hierarchies	1494233	Succeeded	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC
General Ledger Segment Values and Hierarchies Import Process Launcher	1494232	Succeeded	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC	3/18/19 3:44 PM UTC
Load File to Interface	1494231	Succeeded	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC
Load File to Interface	1494230	Succeeded	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC	3/18/19 3:43 PM UTC
Transfer File	1494229	Succeeded	3/18/19 3:42 PM UTC	3/18/19 3:42 PM UTC	3/18/19 3:43 PM UTC
Load Interface File for Import	1494228	Succeeded	3/18/19 3:42 PM UTC	3/18/19 3:42 PM UTC	3/18/19 3:44 PM UTC

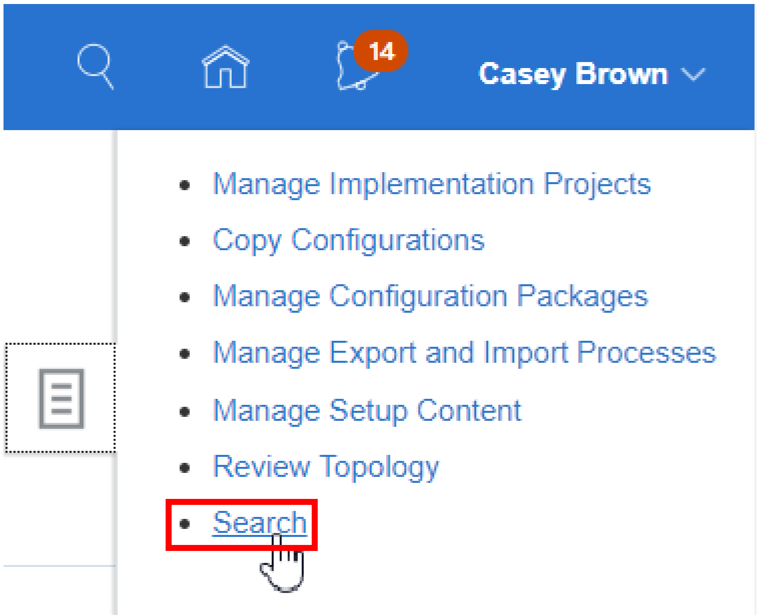
Step 2: Verifying the Exported Data in Oracle Financials Cloud General Ledger

Next, you browse the hierarchy in Oracle Financials Cloud General Ledger to verify that the node that was added in Oracle Fusion Cloud Enterprise Data Management was loaded correctly.

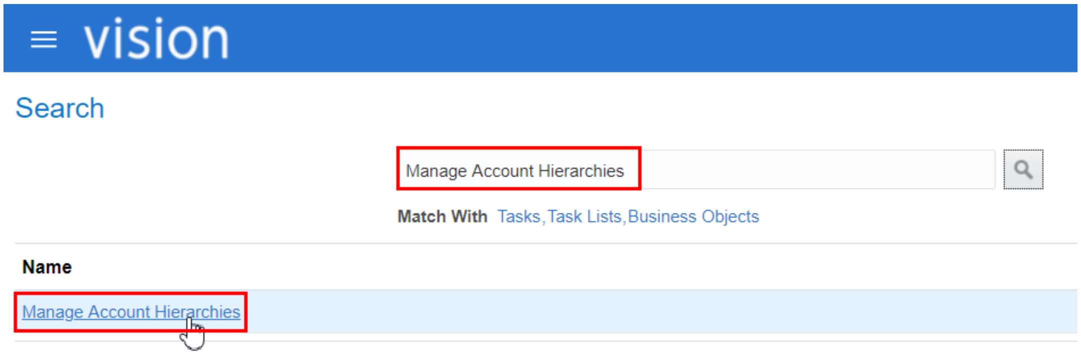
1. In Oracle Financials Cloud, click , and then under **Others**, select **Setup and Maintenance**.



2. Click , and then click **Search**.


14
Casey Brown
Manage Implementation Projects
Copy Configurations
Manage Configuration Packages
Manage Export and Import Processes
Manage Setup Content
Review Topology
Search

3. Search for and then select `Manage Account Hierarchies`.



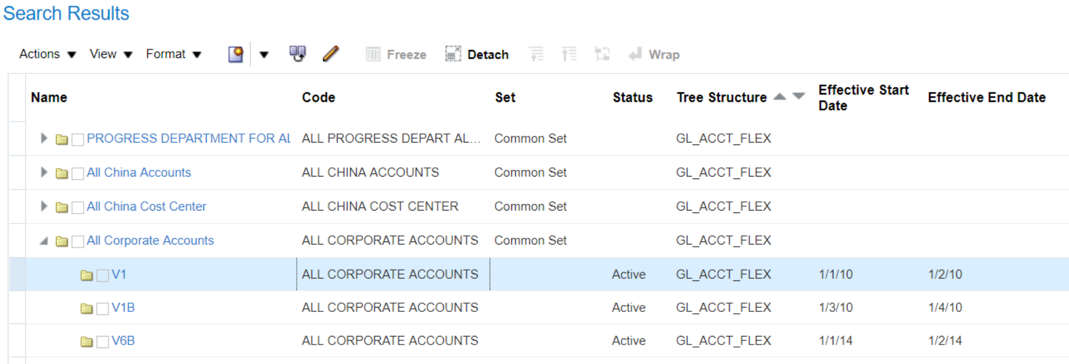
Search

Manage Account Hierarchies

Match With Tasks, Task Lists, Business Objects

Name
Manage Account Hierarchies

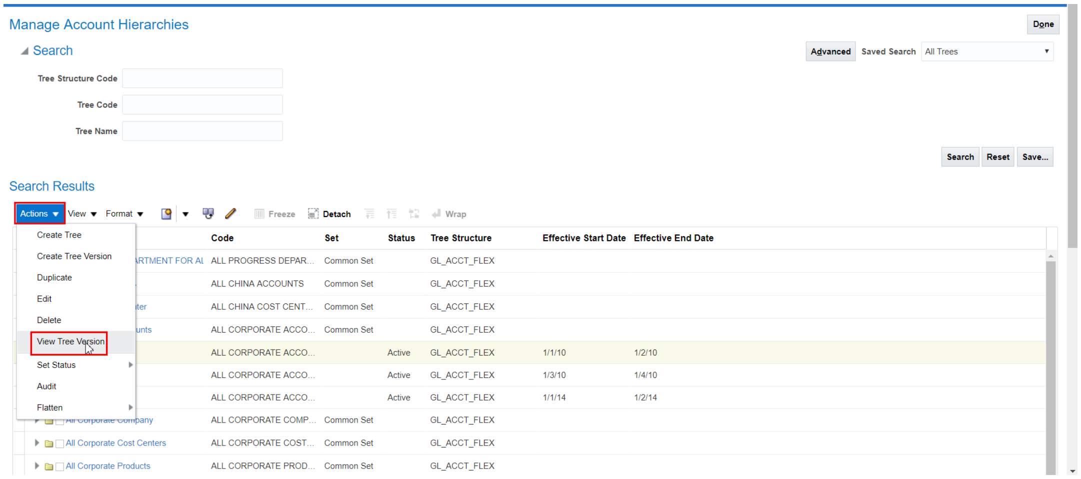
4. In the search results, expand **All Corporate Accounts**, and then highlight the line for **V1**. Do not click on **V1**, as that will put the tree version in draft mode.



Search Results

Name	Code	Set	Status	Tree Structure	Effective Start Date	Effective End Date
PROGRESS DEPARTMENT FOR AI	ALL PROGRESS DEPART AL...	Common Set		GL_ACCT_FLEX		
All China Accounts	ALL CHINA ACCOUNTS	Common Set		GL_ACCT_FLEX		
All China Cost Center	ALL CHINA COST CENTER	Common Set		GL_ACCT_FLEX		
All Corporate Accounts	ALL CORPORATE ACCOUNTS	Common Set		GL_ACCT_FLEX		
V1	ALL CORPORATE ACCOUNTS		Active	GL_ACCT_FLEX	1/1/10	1/2/10
V1B	ALL CORPORATE ACCOUNTS		Active	GL_ACCT_FLEX	1/3/10	1/4/10
V6B	ALL CORPORATE ACCOUNTS		Active	GL_ACCT_FLEX	1/1/14	1/2/14

5. Select **Actions**, and then **View Tree Version**.



Manage Account Hierarchies

Search

Saved Search All Trees

Tree Structure Code

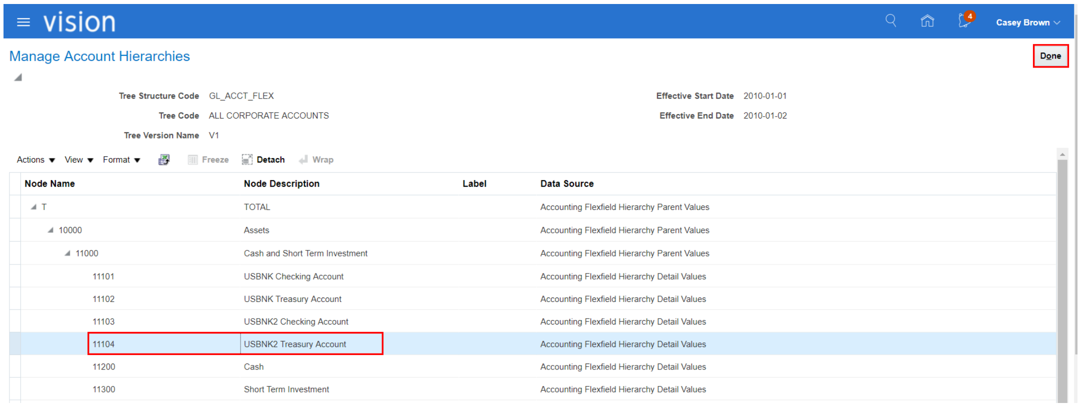
Tree Code

Tree Name

Search Results

	Code	Set	Status	Tree Structure	Effective Start Date	Effective End Date
ARTMENT FOR AI	ALL PROGRESS DEPAR...	Common Set		GL_ACCT_FLEX		
	ALL CHINA ACCOUNTS	Common Set		GL_ACCT_FLEX		
ter	ALL CHINA COST CENT...	Common Set		GL_ACCT_FLEX		
unts	ALL CORPORATE ACCO...	Common Set		GL_ACCT_FLEX		
	ALL CORPORATE ACCO...		Active	GL_ACCT_FLEX	1/1/10	1/2/10
	ALL CORPORATE ACCO...		Active	GL_ACCT_FLEX	1/3/10	1/4/10
	ALL CORPORATE ACCO...		Active	GL_ACCT_FLEX	1/1/14	1/2/14
All Corporate Company	ALL CORPORATE COMP...	Common Set		GL_ACCT_FLEX		
All Corporate Cost Centers	ALL CORPORATE COST...	Common Set		GL_ACCT_FLEX		
All Corporate Products	ALL CORPORATE PROD...	Common Set		GL_ACCT_FLEX		

6. Expand the hierarchy to verify that node `11104` was added, and then click **Done**.



Manage Account Hierarchies

Tree Structure Code	GL_ACCT_FLEX	Effective Start Date	2010-01-01
Tree Code	ALL CORPORATE ACCOUNTS	Effective End Date	2010-01-02
Tree Version Name	V1		

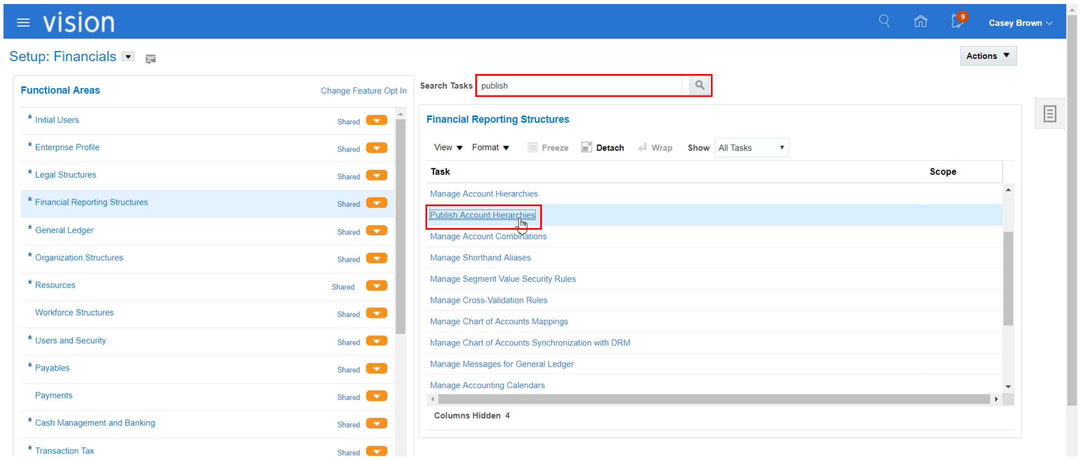
Node Name	Node Description	Label	Data Source
T	TOTAL		Accounting Flexfield Hierarchy Parent Values
10000	Assets		Accounting Flexfield Hierarchy Parent Values
11000	Cash and Short Term Investment		Accounting Flexfield Hierarchy Parent Values
11101	USBNK Checking Account		Accounting Flexfield Hierarchy Detail Values
11102	USBNK Treasury Account		Accounting Flexfield Hierarchy Detail Values
11103	USBNK2 Checking Account		Accounting Flexfield Hierarchy Detail Values
11104	USBNK2 Treasury Account		Accounting Flexfield Hierarchy Detail Values
11200	Cash		Accounting Flexfield Hierarchy Detail Values
11300	Short Term Investment		Accounting Flexfield Hierarchy Detail Values

Step 3: Publishing the Updated Oracle Financials Cloud General Ledger Hierarchy

After the data has been loaded and verified, the final step is to publish the updated hierarchy.

To publish the account hierarchy:

1. In Search Tasks, search for `publish`, and then click **Publish Account Hierarchies**.



Setup: Financials

Search Tasks: publish

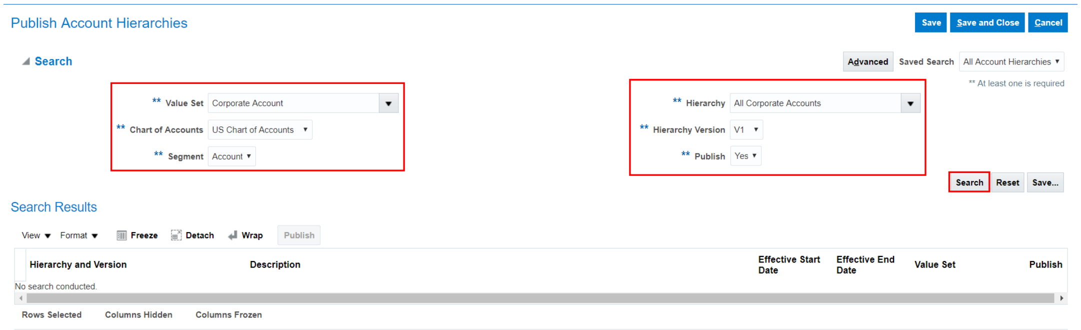
Functional Areas	
* Initial Users	Shared
* Enterprise Profile	Shared
* Legal Structures	Shared
* Financial Reporting Structures	Shared
* General Ledger	Shared
* Organization Structures	Shared
* Resources	Shared
Workforce Structures	Shared
* Users and Security	Shared
* Payables	Shared
Payments	Shared
* Cash Management and Banking	Shared
* Transaction Tax	Shared

Financial Reporting Structures

Task	Scope
Manage Account Hierarchies	
Publish Account Hierarchies	
Manage Account Combinations	
Manage Shorthand Aliases	
Manage Segment Value Security Rules	
Manage Cross-Validation Rules	
Manage Chart of Accounts Mappings	
Manage Chart of Accounts Synchronization with DRM	
Manage Messages for General Ledger	
Manage Accounting Calendars	

Columns Hidden 4

2. Enter the following search criteria, and then click **Search**.
   - **Value Set**: Corporate Account
   - **Chart of Accounts**: US Chart of Accounts
   - **Segment**: Account
   - **Hierarchy**: All Corporate Accounts
   - **Hierarchy Version**: V1
   - **Publish**: Yes



** At least one is required

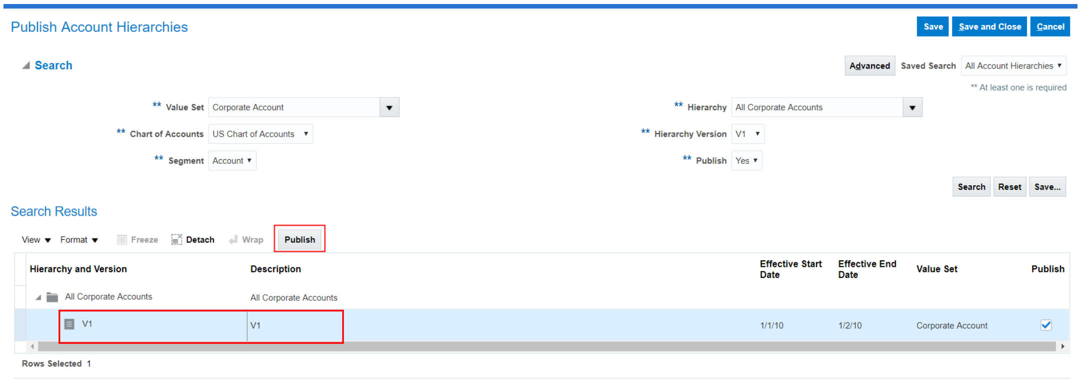
** Value Set: Corporate Account
** Chart of Accounts: US Chart of Accounts
** Segment: Account
** Hierarchy: All Corporate Accounts
** Hierarchy Version: V1
** Publish: Yes

Hierarchy and Version	Description	Effective Start Date	Effective End Date	Value Set	Publish

No search conducted.

Rows Selected Columns Hidden Columns Frozen

3. Highlight the **V1** row, and then click **Publish**.



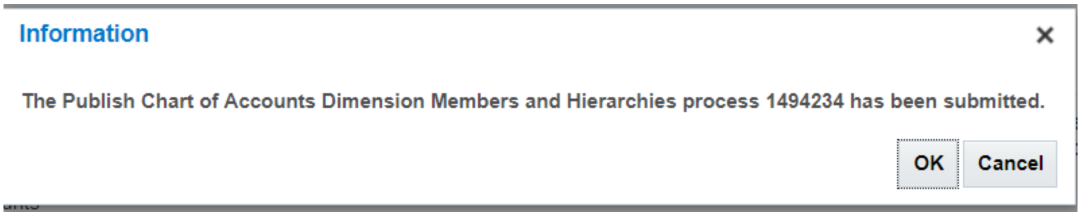
** At least one is required

** Value Set: Corporate Account
** Chart of Accounts: US Chart of Accounts
** Segment: Account
** Hierarchy: All Corporate Accounts
** Hierarchy Version: V1
** Publish: Yes

Hierarchy and Version	Description	Effective Start Date	Effective End Date	Value Set	Publish
All Corporate Accounts	All Corporate Accounts				
V1	V1	1/1/10	1/2/10	Corporate Account	☑

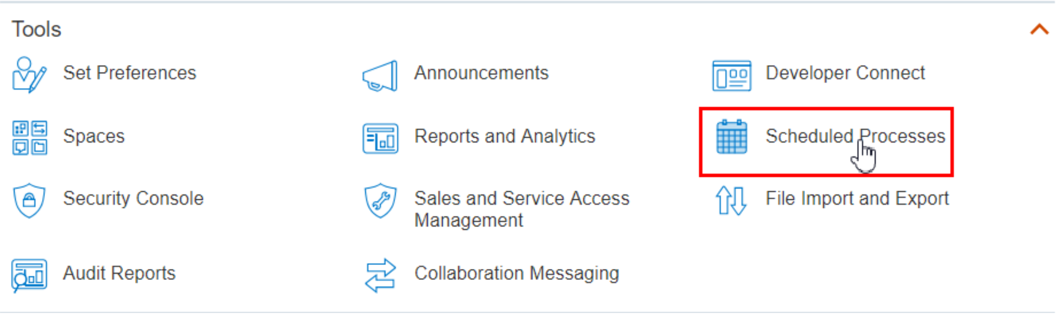
Rows Selected 1

4. Review the confirmation dialog box, and then click **OK**.

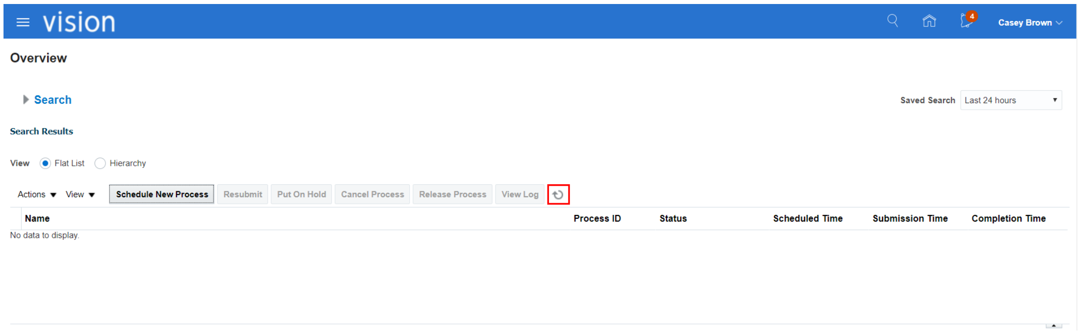


The Publish Chart of Accounts Dimension Members and Hierarchies process 1494234 has been submitted.

5. Click , and then under **Tools**, select **Scheduled Processes**.



6. On the **Overview** screen, click to refresh the scheduled processes.



Overview

Search

Saved Search Last 24 hours

Search Results

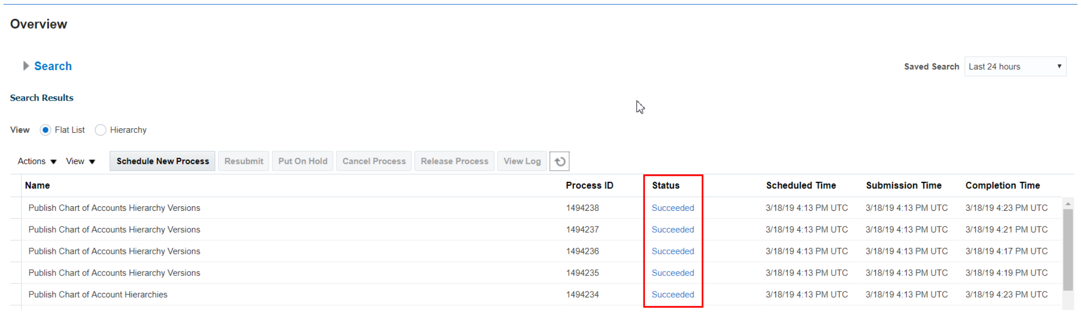
View ● Flat List ○ Hierarchy

Actions ▾ View ▾ Schedule New Process Resubmit Put On Hold Cancel Process Release Process View Log

Name	Process ID	Status	Scheduled Time	Submission Time	Completion Time

No data to display.

7. Verify that all scheduled processes for the publish action completed successfully.



Overview

Search

Saved Search Last 24 hours

Search Results

View ● Flat List ○ Hierarchy

Actions ▾ View ▾ Schedule New Process Resubmit Put On Hold Cancel Process Release Process View Log

Name	Process ID	Status	Scheduled Time	Submission Time	Completion Time
Publish Chart of Accounts Hierarchy Versions	1494238	Succeeded	3/18/19 4:13 PM UTC	3/18/19 4:13 PM UTC	3/18/19 4:23 PM UTC
Publish Chart of Accounts Hierarchy Versions	1494237	Succeeded	3/18/19 4:13 PM UTC	3/18/19 4:13 PM UTC	3/18/19 4:21 PM UTC
Publish Chart of Accounts Hierarchy Versions	1494236	Succeeded	3/18/19 4:13 PM UTC	3/18/19 4:13 PM UTC	3/18/19 4:17 PM UTC
Publish Chart of Accounts Hierarchy Versions	1494235	Succeeded	3/18/19 4:13 PM UTC	3/18/19 4:13 PM UTC	3/18/19 4:19 PM UTC
Publish Chart of Account Hierarchies	1494234	Succeeded	3/18/19 4:13 PM UTC	3/18/19 4:13 PM UTC	3/18/19 4:23 PM UTC

42

Integrating with External Applications Using a Universal Application

This scenario demonstrates how to manage dimensions from an external application using a Universal application. The steps walks you through registering your Universal application and importing your dimensions, managing your enterprise data, and exporting changes back to your application.

The steps below are focused on the administrative tasks required to complete the register to export process within Oracle Fusion Cloud Enterprise Data Management. Only a general overview of data management tasks are provided with this scenario.

Administrator Steps

Administrator Prerequisites

You need the Application - Create role to register the application. After you register an application you are automatically assigned the Owner permission for the application and the default view.

A dimension export from your external application is required when using the Universal application wizard. You must prepare a dimension export that includes the data for the dimension you want to import and manage in Oracle Fusion Cloud Enterprise Data Management.

1. Export dimension data from your external application.
2. Prepare the data into a .CSV file format for import into a Universal application in Cloud EDM, see Import and Export File Format for User Defined Dimensions.

Administrator Process

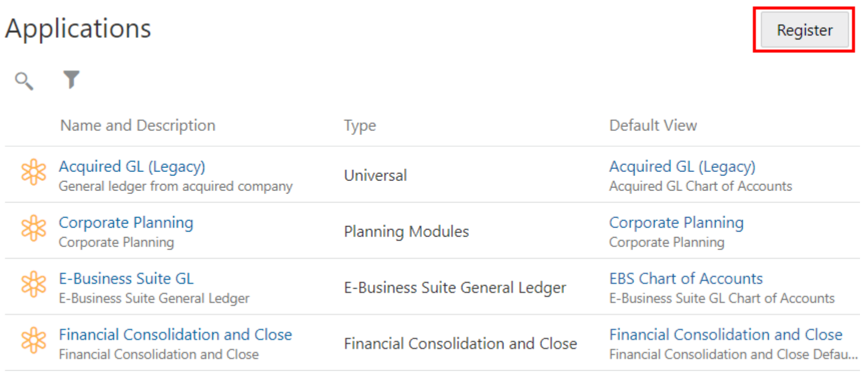
1. Register your Universal application, see Registering a Universal Application.
2. Import the dimensions, see Importing a User Defined Dimension.
3. Provide your users with the appropriate access to the data chain elements needed to manage the dimensions for the Universal application, see Working with Roles and Permissions.
4. Export dimensions to files that can be imported into your external application, see Exporting a User Defined Dimension.

Administrator Detailed Steps

Step 1: Register Your Universal Application

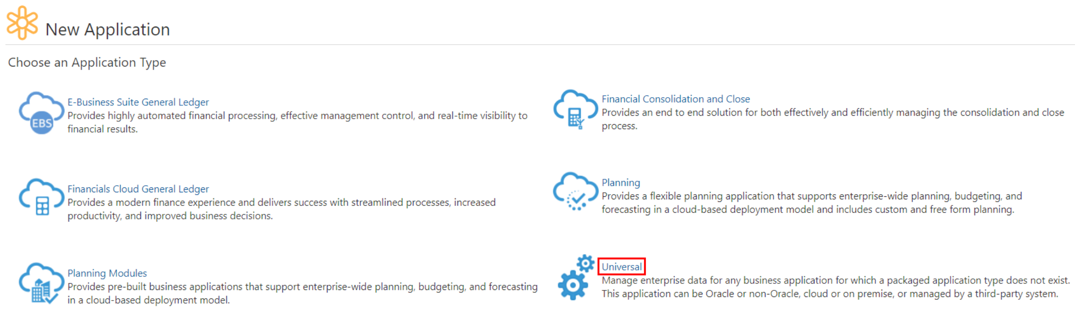
To register your Universal application:

1. In Applications, click **Register**.



Name and Description	Type	Default View
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning
E-Business Suite GL E-Business Suite General Ledger	E-Business Suite General Ledger	EBS Chart of Accounts E-Business Suite GL Chart of Accounts
Financial Consolidation and Close Financial Consolidation and Close	Financial Consolidation and Close	Financial Consolidation and Close Financial Consolidation and Close Defau...

2. Select the Universal application type.



Choose an Application Type

E-Business Suite General Ledger
Provides highly automated financial processing, effective management control, and real-time visibility to financial results.

Financial Consolidation and Close
Provides an end to end solution for both effectively and efficiently managing the consolidation and close process.

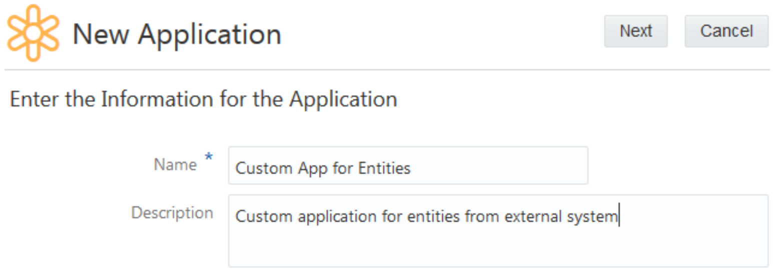
Financials Cloud General Ledger
Provides a modern finance experience and delivers success with streamlined processes, increased productivity, and improved business decisions.

Planning
Provides a flexible planning application that supports enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model and includes custom and free form planning.

Planning Modules
Provides pre-built business applications that support enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model.

Universal
Manage enterprise data for any business application for which a packaged application type does not exist. This application can be Oracle or non-Oracle, cloud or on premise, or managed by a third-party system.

3. Enter a name and description for the application and click **Next**.

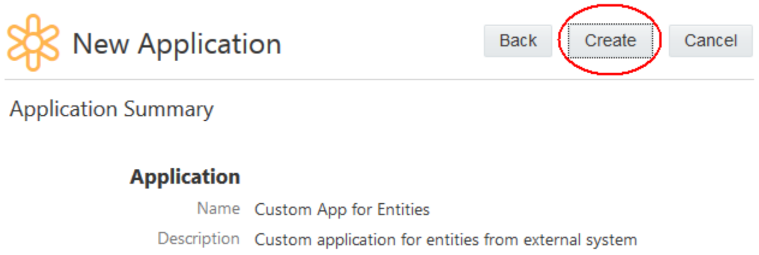


Enter the Information for the Application

Name * Custom App for Entities

Description Custom application for entities from external system

4. Review the application summary and click **Create**.



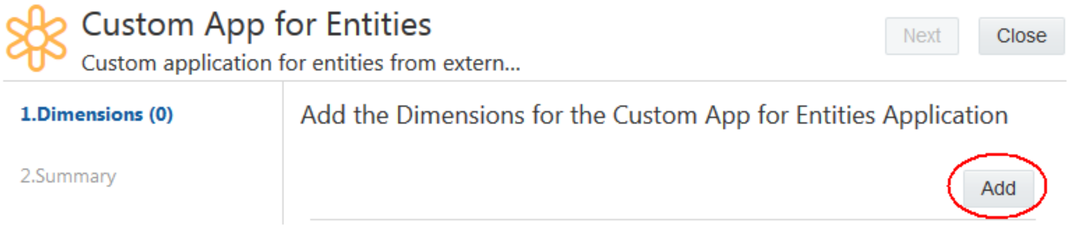
Application Summary

Application

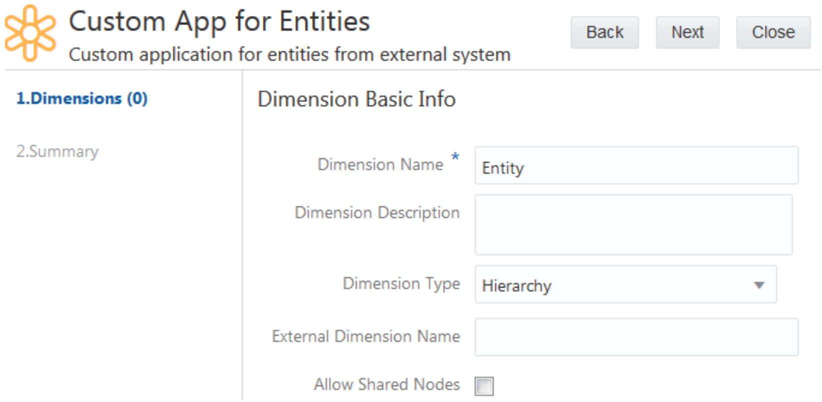
Name Custom App for Entities

Description Custom application for entities from external system

5. Click **Add** to define a dimension for the application, see Adding, Removing, or Modifying a User Defined Dimension.


Custom App for Entities
Custom application for entities from extern...
Next
Close
1.Dimensions (0)
2.Summary
Add the Dimensions for the Custom App for Entities Application
Add

6. Enter dimension information, such as whether the dimension type is a hierarchy or list, and then click **Next**.



Dimension Name * Entity

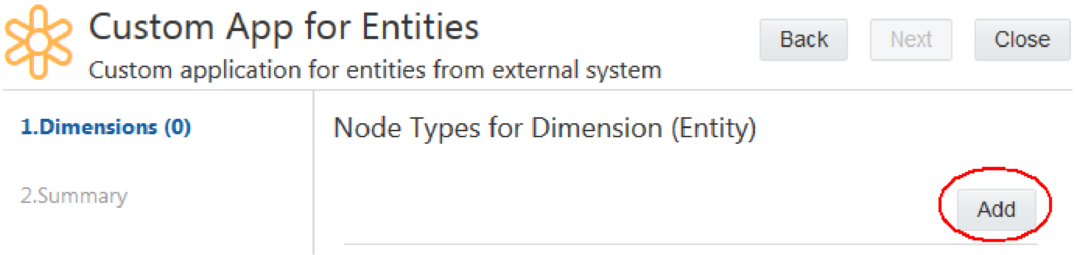
Dimension Description

Dimension Type Hierarchy

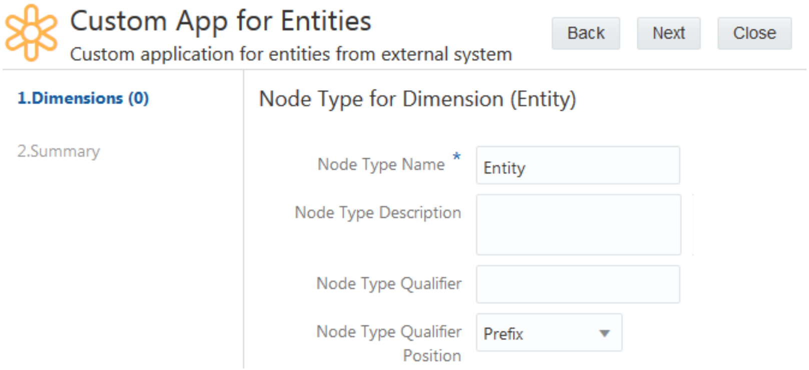
External Dimension Name

Allow Shared Nodes □

7. Click **Add** to define node types for the dimension, see Adding or Modifying a Node Type for a User Defined Dimension.


Custom App for Entities
Custom application for entities from external system
Back
Next
Close
1.Dimensions (0)
2.Summary
Node Types for Dimension (Entity)
Add

8. Enter node type information and click **Next**.


Custom App for Entities
Custom application for entities from external system
Back
Next
Close
1.Dimensions (0)
2.Summary
Node Type for Dimension (Entity)

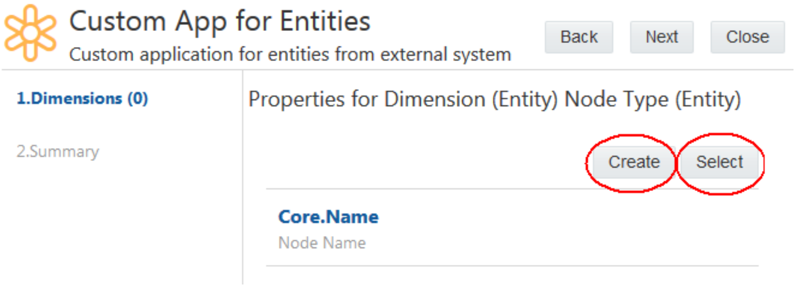
Node Type Name * Entity

Node Type Description

Node Type Qualifier

Node Type Qualifier Position Prefix

9. Click **Create** to define a new property or click **Select** to select an existing property for a node type. After all properties have been configured for the node type, click **Next**. See Adding an Existing Property to a Node Type for a User Defined Dimension and Creating a Custom Property.



Custom App for Entities

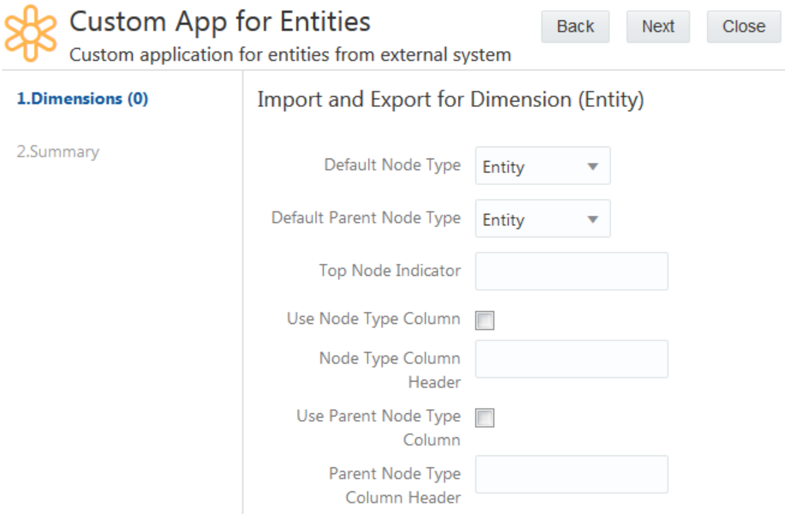
Custom application for entities from external system

Properties for Dimension (Entity) Node Type (Entity)

Core.Name
Node Name

10. Define more node types for the dimension or click **Next** to continue.

11. You can optionally define import and export settings for the dimension based on node types configured and the expected contents of import and export files. Click **Next** to continue.



Custom App for Entities

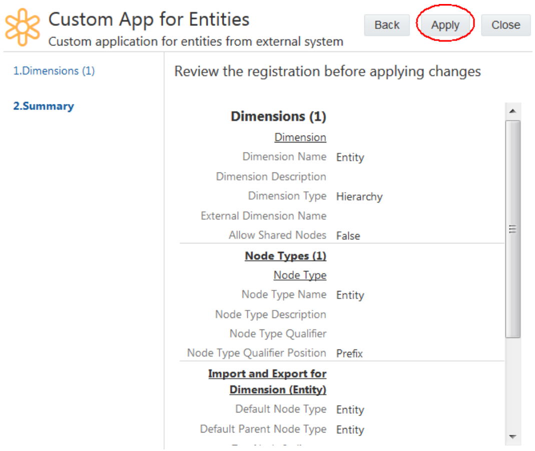
Custom application for entities from external system

Import and Export for Dimension (Entity)

Default Node Type: Entity
Default Parent Node Type: Entity
Top Node Indicator:
Use Node Type Column: □
Node Type Column Header:
Use Parent Node Type Column: □
Parent Node Type Column Header:

12. Repeat steps 5-10 to add more dimensions to the application and then click **Next**.

13. Review all the registration information you have entered and click **Apply** to complete registration of your Universal application.



Custom App for Entities

Custom application for entities from external system

Review the registration before applying changes

Dimensions (1)

Dimension
Dimension Name: Entity
Dimension Description:
Dimension Type: Hierarchy
External Dimension Name:
Allow Shared Nodes: False

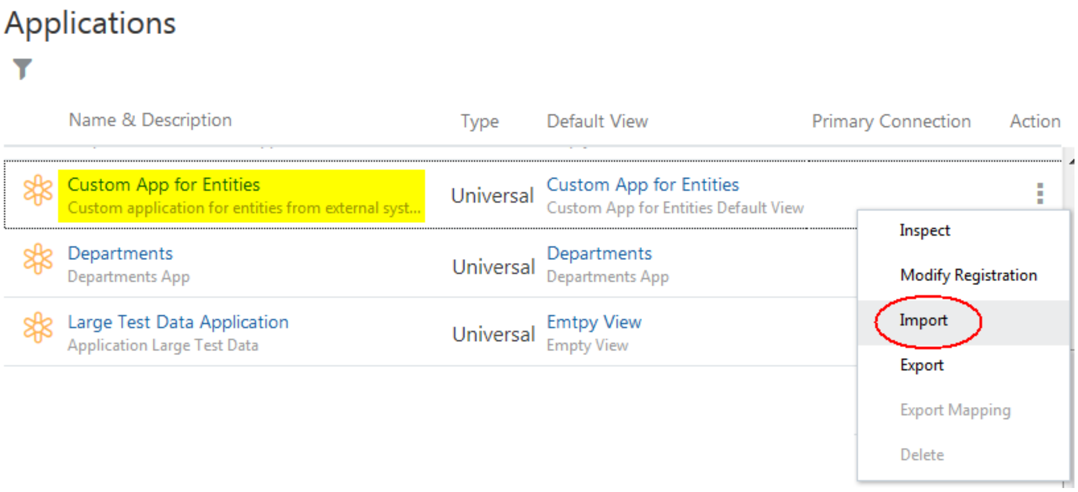
Node Types (1)

Node Type
Node Type Name: Entity
Node Type Description:
Node Type Qualifier:
Node Type Qualifier Position: Prefix

Import and Export for Dimension (Entity)
Default Node Type: Entity
Default Parent Node Type: Entity

Step 2: Import Data for Registered Dimensions

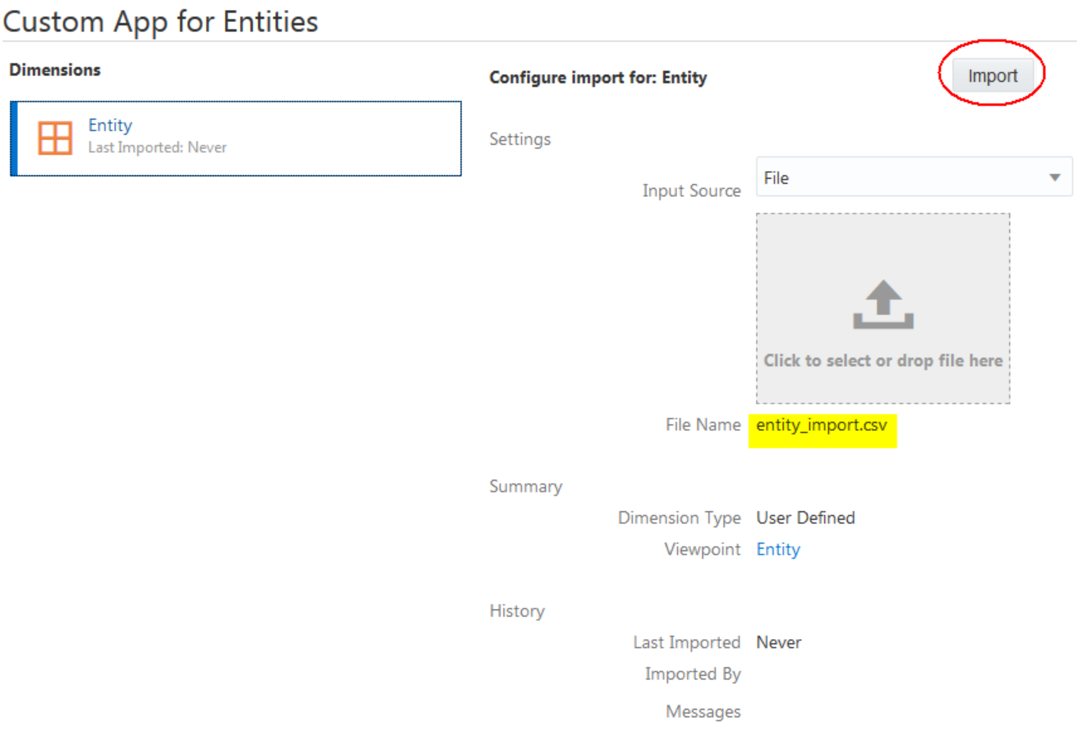
1. From **Applications**, find the application that you registered, click ⋮ and then select **Import**.



Applications

Name & Description	Type	Default View	Primary Connection	Action
Custom App for Entities Custom application for entities from external syst...	Universal	Custom App for Entities Custom App for Entities Default View		
Departments Departments App	Universal	Departments Departments App		
Large Test Data Application Application Large Test Data	Universal	Emtpy View Empty View		

2. Browse and select the file that you created earlier and then click **Import**.



Custom App for Entities

Dimensions

Entity
Last Imported: Never

Configure import for: Entity

Settings

Input Source: File

File Name: entity_import.csv

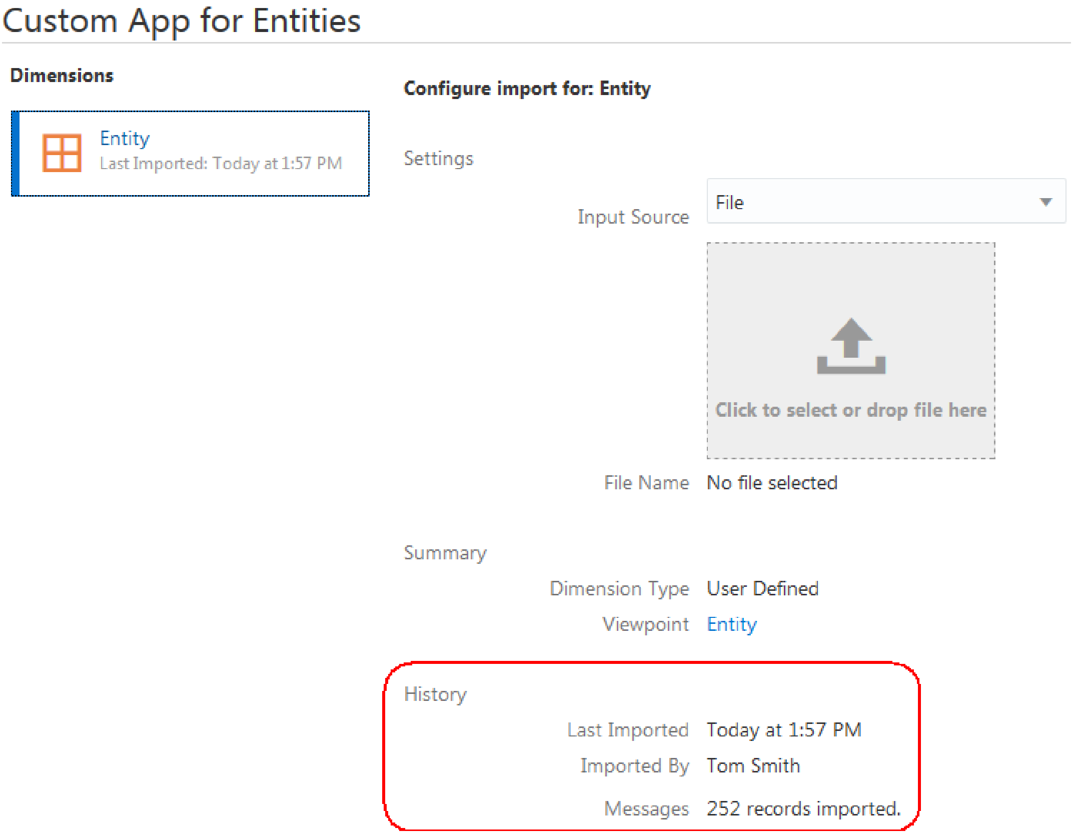
Summary

Dimension Type: User Defined
Viewpoint: Entity

History

Last Imported: Never
Imported By:
Messages:

3. After the import, you can view the number of records that were imported.



Custom App for Entities

Dimensions

Entity
Last Imported: Today at 1:57 PM

Configure import for: Entity

Settings

Input Source: File

File Name: No file selected

Summary

Dimension Type: User Defined

Viewpoint: Entity

History

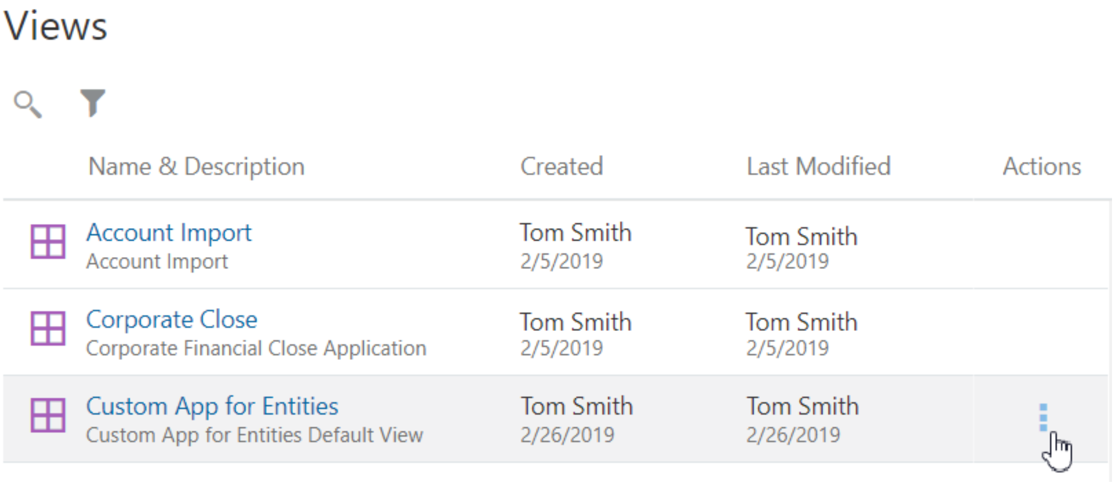
Last Imported: Today at 1:57 PM

Imported By: Tom Smith

Messages: 252 records imported.

Step 3: Make Changes to Enterprise Data

In this step, your users make changes to the enterprise data that was imported. Provide the users with the name of the default view that was created for the Universal application during registration.



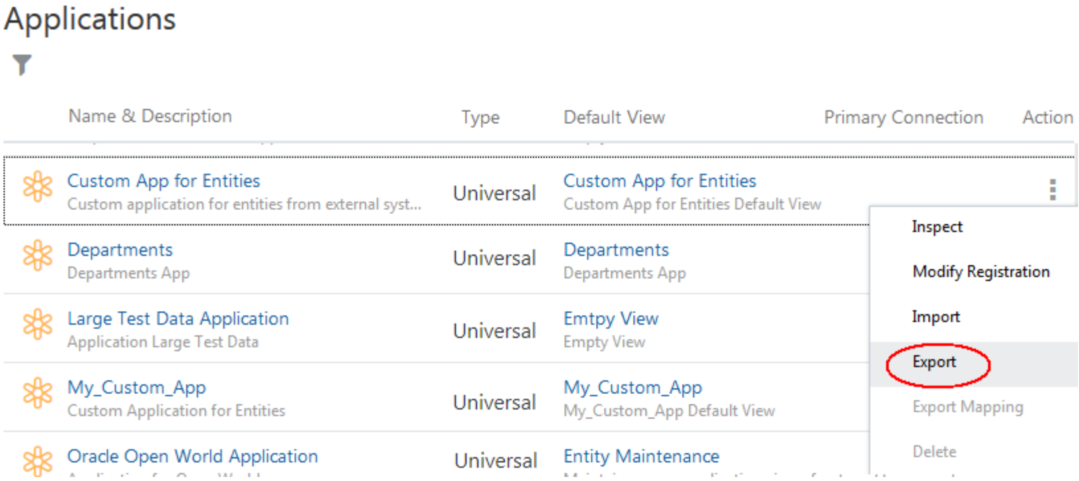
Views

Name & Description	Created	Last Modified	Actions
Account Import Account Import	Tom Smith 2/5/2019	Tom Smith 2/5/2019	
Corporate Close Corporate Financial Close Application	Tom Smith 2/5/2019	Tom Smith 2/5/2019	
Custom App for Entities Custom App for Entities Default View	Tom Smith 2/26/2019	Tom Smith 2/26/2019	⋮

Step 4: Export Dimensions Back to Your External Application

After your users are done making changes, you can export the enterprise data to files which can then be imported into your external application.

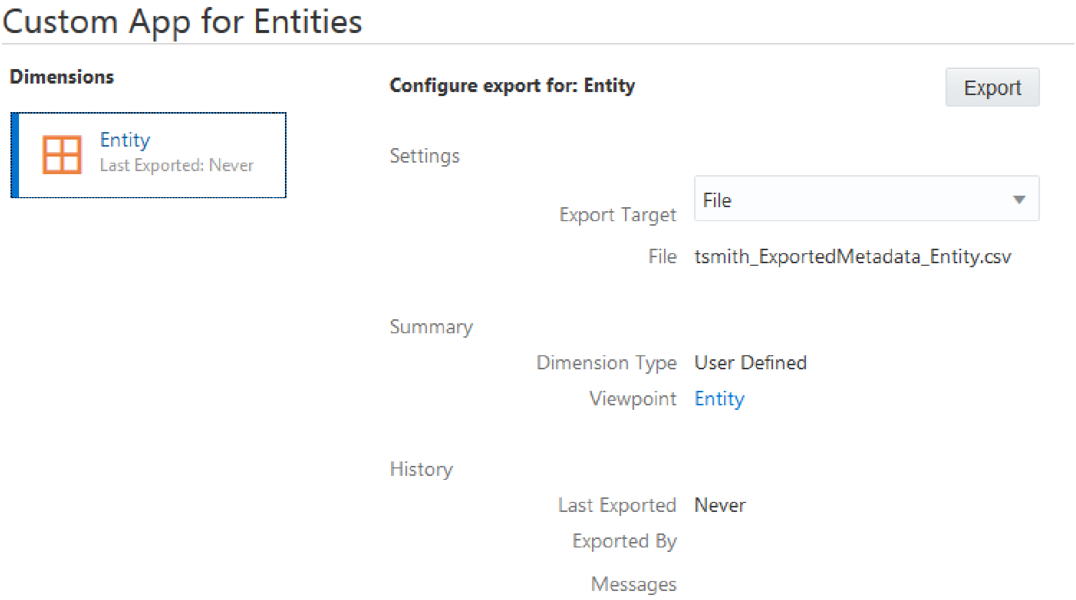
1. In the **Applications** worklist, find the application, click ⋮ and then select **Export**.



Applications

Name & Description	Type	Default View	Primary Connection	Action
Custom App for Entities Custom application for entities from external syst...	Universal	Custom App for Entities Custom App for Entities Default View		
Departments Departments App	Universal	Departments Departments App		
Large Test Data Application Application Large Test Data	Universal	Emtpy View Empty View		
My_Custom_App Custom Application for Entities	Universal	My_Custom_App My_Custom_App Default View		
Oracle Open World Application	Universal	Entity Maintenance		

2. Select the dimension to export and then click **Export**.



Custom App for Entities

Dimensions

Entity
Last Exported: Never

Configure export for: Entity

Settings

Export Target: File

File: tsmith_ExportedMetadata_Entity.csv

Summary

Dimension Type: User Defined

Viewpoint: Entity

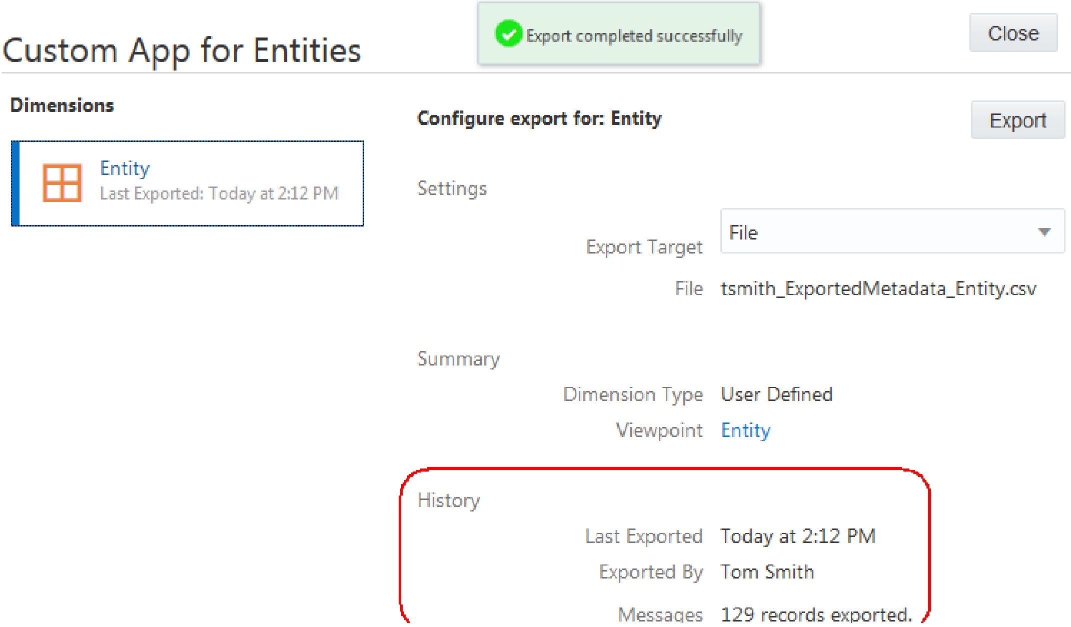
History

Last Exported: Never

Exported By

Messages

The export file is downloaded and saved to a local file system. After the export, you can view the number of records that were exported.


Custom App for Entities
Export completed successfully
Close
Dimensions
Entity
Last Exported: Today at 2:12 PM
Configure export for: Entity
Export
Settings
Summary
History

Export Target: File

File: tsmith_ExportedMetadata_Entity.csv

Dimension Type: User Defined

Viewpoint: Entity

Last Exported: Today at 2:12 PM

Exported By: Tom Smith

Messages: 129 records exported.

User Steps

User Prerequisites

Your administrator must follow the steps in this scenario to successfully register your Universal application and import enterprise data elements. Then you can manage your enterprise data based on your security privileges.

User Process

This is a high level summary of the tasks you can perform:

- Make maintenance views and viewpoints, see Working with Viewpoints
- Use side by side layout and compare to rationalize nodes across viewpoints within a view, see Comparing Viewpoints
- Update nodes and properties and update alternate hierarchies for what if analysis, see Making Changes Interactively
- Map nodes from one hierarchy to another, see Creating Mapping Viewpoints
- Load enterprise data such as accounts, see Making Changes Using a Load File
- Browse draft and completed requests in the Requests list, see Working with Request Activity

Outcome

You now have registered your Universal application so you can proactively manage changes across this application. You have imported the dimensions that you want to manage and have successfully exported updated data out of Oracle Fusion Cloud Enterprise Data Management into files that you can load into your external system. You have full request activity audit information to know what data was changed, who made those changes, and when they were made.

43 Sharing Enterprise Data Across Applications

This scenario demonstrates how to make hierarchical changes and maintain hierarchical synchronization across multiple external applications. The steps walk you through adding a new node to a single dimension hierarchy and maintaining the change in another application. You can repeat the steps for additional dimensional hierarchies. Similar steps can also be used for other hierarchy and property changes.

Suppose your General Ledger application stores companies in company segments and your Planning application stores companies in an Entity dimension. To add a new company to both applications the Administrator needs to create a maintenance view with both applications and the user can add nodes to both applications.

Administrator Steps

Administrator Prerequisites

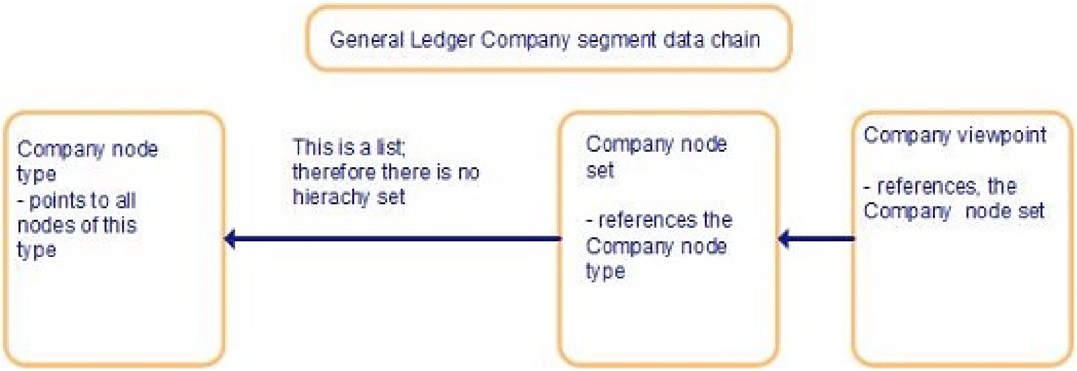
As an administrator you need the Application - Create role to register the two applications, import the dimensions, and create a node type converter. You need the Views - Create role to create the maintenance view.

Administrator Process

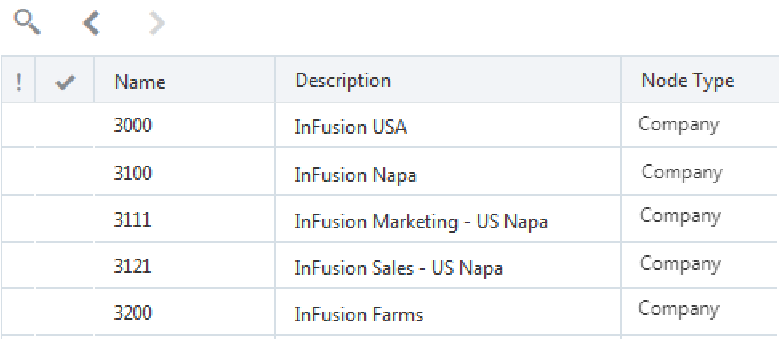
1. Register and import the General Ledger application and Planning application.
2. Create a node type converter to convert the General Ledger Company node type to the Planning Entity node type.

Administrator Detailed Steps

1. Register and import the General Ledger application and include the Company segment. The following diagram shows the data chain:

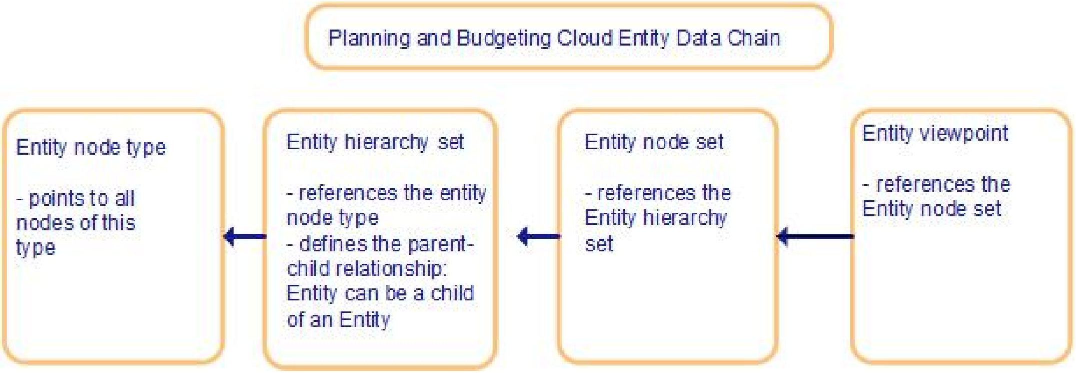

General Ledger Company segment data chain
Company node type
- points to all nodes of this type
This is a list; therefore there is no hierachy set
Company node set
- references the Company node type
Company viewpoint
- references, the Company node set

The following diagram shows the Company viewpoint nodes:

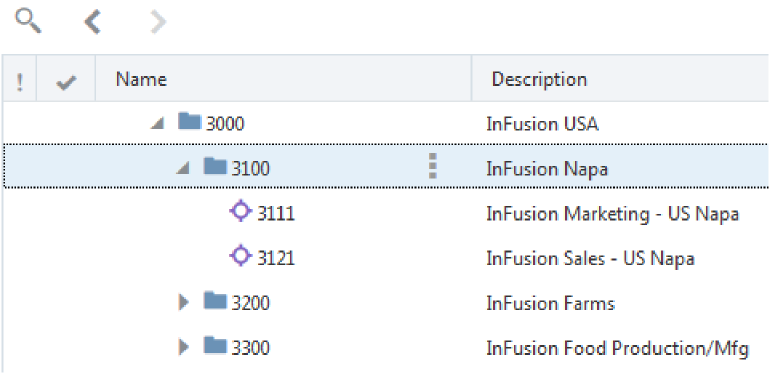


!	✓	Name	Description	Node Type
		3000	InFusion USA	Company
		3100	InFusion Napa	Company
		3111	InFusion Marketing - US Napa	Company
		3121	InFusion Sales - US Napa	Company
		3200	InFusion Farms	Company

2. Register and import the Planning application and the Entity dimension. The following diagram shows the imported Entity data chain:

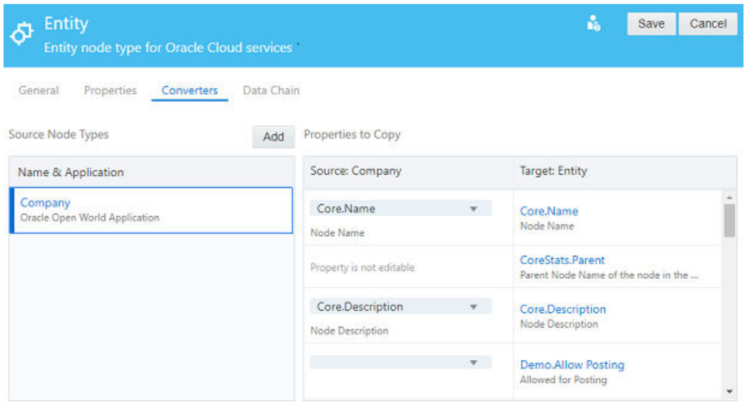

Planning and Budgeting Cloud Entity Data Chain
Entity node type
- points to all nodes of this type
Entity hierarchy set
- references the entity node type
- defines the parent-child relationship: Entity can be a child of an Entity
Entity node set
- references the Entity hierarchy set
Entity viewpoint
- references the Entity node set

The following diagram shows the Entity viewpoint nodes:



!	✓	Name	Description
		◢ 3000	InFusion USA
		◢ 3100	InFusion Napa
		3111	InFusion Marketing - US Napa
		3121	InFusion Sales - US Napa
		▶ 3200	InFusion Farms
		▶ 3300	InFusion Food Production/Mfg

3. Create a node type converter to convert the General Ledger Company node type to the Planning Entity node type as shown below.



Entity
Entity node type for Oracle Cloud services

Name & Application
Company Oracle Open World Application

Source: Company	Target: Entity
Core.Name Node Name	Core.Name Node Name
Property is not editable	CoreStats.Parent Parent Node Name of the node in the ...
Core.Description Node Description	Core.Description Node Description
	Demo.Allow Posting Allowed for Posting

4. Create a maintenance view and create two viewpoints:
   - Create an Entity viewpoint using the Entity node set.
   - Create a Company viewpoint using the Company node set.
5. Give the user the name of the maintenance view and *Participant(Write)* permission to both applications.

Related Topics:

- Working with Applications
- Working with Node Type Converters
- Security Examples
- Exporting Dimensions

User Steps

User Prerequisites

Your administrator needs to give you the maintenance view with at least *Participant(Write)* permission to the two applications or dimensions. This enables you to update nodes in both applications or dimensions.

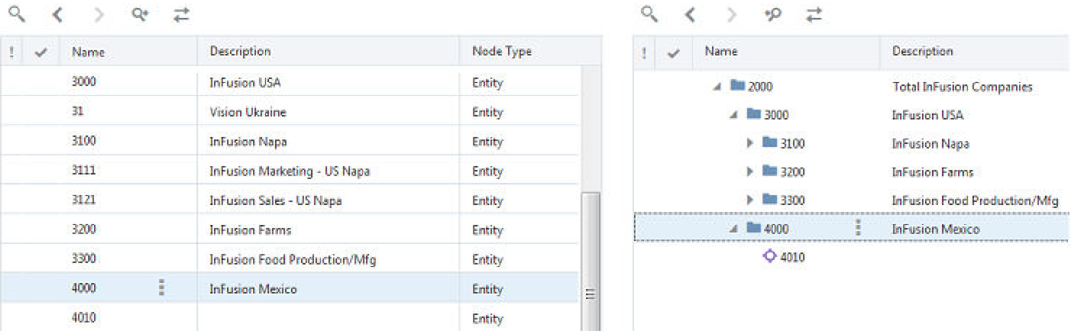
User Process

Using a side-by-side layout, create a new node and place it in both applications.

User Detailed Steps

1. Open the maintenance view and click the **Side by Side** button and display the General Ledger viewpoint and the Planning viewpoint.
2. Create or open a request.
3. Add the Infusion Mexico node to the General Ledger list.
4. Drag and the drop the Infusion Mexico node into the Planning viewpoint; it is converted to an Entity node type using the node type converter. The Infusion Mexico node is a Company node type in the General Ledger viewpoint on the left and an Entity node type in

the Planning viewpoint on the right.



!	✓	Name	Description	Node Type
		3000	InFusion USA	Entity
		31	Vision Ukraine	Entity
		3100	InFusion Napa	Entity
		3111	InFusion Marketing - US Napa	Entity
		3121	InFusion Sales - US Napa	Entity
		3200	InFusion Farms	Entity
		3300	InFusion Food Production/Mfg	Entity
		4000	InFusion Mexico	Entity
		4010		Entity

!	✓	Name	Description
		2000	Total InFusion Companies
		3000	InFusion USA
		3100	InFusion Napa
		3200	InFusion Farms
		3300	InFusion Food Production/Mfg
		4000	InFusion Mexico
		4010	

5. Submit the request.
6. You can continue to add nodes and make updates to both applications.

Outcome

You now have a process where you can manage all hierarchies, from different external applications, in one maintenance view. You can quickly identify differences from one hierarchy to another and take corrective action to ensure that your hierarchies meet your business need.

44 Comparing Enterprise Data Across Applications

You can use Oracle Fusion Cloud Enterprise Data Management to ensure that a hierarchy contains all the leaf nodes that are contained by another hierarchy. To do so, an administrator sets up a node type converter and a user uses the comparison and request features.

> **Note:**
>
> The hierarchy that contains all the leaf nodes is called the "source hierarchy". The hierarchy you are checking for missing leaf nodes is called the "target hierarchy".

This scenario also shows the role of node type qualifiers. The nodes in the target hierarchy are all prefixed with "cc_". For example, if the source hierarchy contains a node named 100, the corresponding node in the target hierarchy is named cc_100. The node type qualifier will be applied to leaf nodes when a user copies nodes from the source to the target hierarchy. For more information, see Working with Node Type Qualifiers.

Administrator Steps

Administrator Prerequisites

You must have the *Application - Create* role to register the applications. After you register the applications, you are assigned the *Owner* permission to the application and default view. Users must have the *Views - Create* role in order for them to create a view, and the *Participant(Write)* permission on the application to enable them to make changes to data.

Administrator Process

An administrator:

1. Registers two Universal applications.
2. Imports data.
3. Creates a node type converter, see Working with Node Type Converters.

Administrator Detailed Steps

To perform these steps, you must be familiar with Registering a Universal Application.

1. Create a Universal application named Source Hierarchy, then add a dimension by taking the following steps:
   a. Enter Cost Center as the **Dimension Name** and then click **Next**.
   b. Click **Add**.
   c. Enter Cost Center as the **Node Type Name** and then click **Next**.
   d. Click **Select**.

e. Select the `Core.Description` property and then click **OK**.

f. Enter Description as the **Column Header** and then click **Next** a few times until the Import and Export for Dimension page displays.

g. Perform the following steps:

   - Select **Use Node Type Column**.
   - Enter Node Type as the **Node Type Column Header**.
   - Select **Use Parent Node Type Column**.
   - Enter Parent Node Type as the **Parent Node Type Column Header**.

h. Finish registering the Source Hierarchy application.

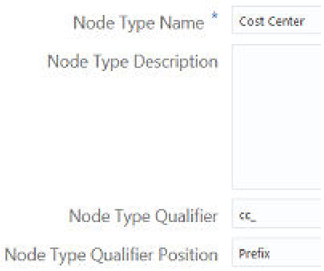
2. Create a Universal application named Target Hierarchy, then add a dimension by taking the following steps:

   a. Enter Cost Center as the **Dimension Name** and then click **Next**.

   b. Click **Add**.

   c. Enter Cost Center as the **Node Type Name**.

   d. Specify cc_ as the **Node Type Qualifier**, as shown in the following example:



Node Type Name * Cost Center
Node Type Description
Node Type Qualifier cc_
Node Type Qualifier Position Prefix

   e. Click **Next**.

   f. Click **Select**.

   g. Select the `Core.Description` property and then click **OK**.

   h. Enter Description as the **Column Header**, then click **Next** a few times until the Import and Export for Dimension page displays.

   i. Perform the following steps:

      - Select **Use Node Type Column**.
      - Enter Node Type as the **Node Type Column Header**.
      - Select **Use Parent Node Type Column**.
      - Enter Parent Node Type as the **Parent Node Type Column Header**.

   j. Finish registering the Target Hierarchy application.

3. Import a comma-delimited (`.csv`) file containing the following fields into the Source Hierarchy application:

```
Name,Description,Node Type,Parent,Parent Node Type
1000,Total Cost Centers,Cost Center, ,
100,Research and Development,Cost Center,1000,Cost Center
110,R and D US,Cost Center,1000,Cost Center
```

```
111,R and D Growth,Cost Center,1000,Cost Center
112,R and D Mfg,Cost Center,1000,Cost Center
120,R and D Canada,Cost Center,1000,Cost Center
```

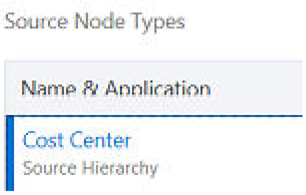
4. Import a comma-delimited (`.csv`) file containing the following fields into the Target Hierarchy application:

```
Name,Description,Node Type,Parent,Parent Node Type
cc_1000,Cost Centers,Cost Center, ,
cc_100,Research and Development,Cost Center,cc_1000,Cost Center
cc_110,R and D US,Cost Center,cc_1000,Cost Center
cc_111,R and D Growth,Cost Center,cc_1000,Cost Center
```

Create a node type converter by performing the following steps.

5. Click **Node Types**.
6. Click the **Cost Center** link for the Target Hierarchy application.
7. Click **Converters**.
8. Click **Edit** and then click **Add**.
9. Select the Cost Center node type for the Source Hierarchy application and then click **OK**.

   The node type converter should look like the following example:

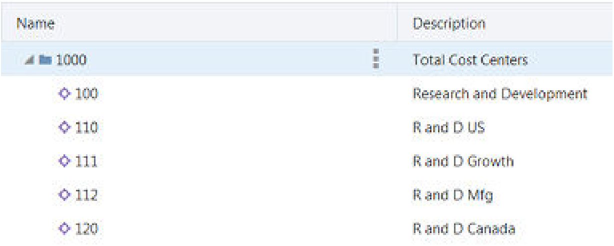


Source Node Types

Name & Application
Cost Center Source Hierarchy

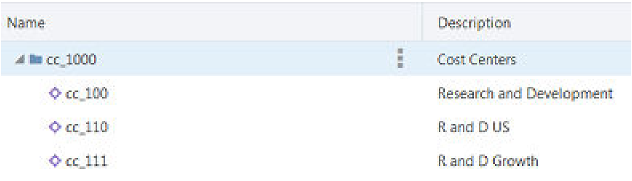
10. Click **Save** and then click **Close**.
11. Check the default views of both applications to verify that the nodes were correctly imported.

    The source hierarchy should look like the following example:



Name	Description
1000	Total Cost Centers
100	Research and Development
110	R and D US
111	R and D Growth
112	R and D Mfg
120	R and D Canada

The target hierarchy should look like the following example:



Name	Description
cc_1000	Cost Centers
cc_100	Research and Development
cc_110	R and D US
cc_111	R and D Growth

User Steps

User Prerequisites

You must have the *Views - Create* role in order to create views, and the *Participant(Write)* permission on the application in order to make changes to application data.

User Process

To resolve differences between the source and target hierarchies, a user:

1. Creates a view containing viewpoints for both the source and target hierarchies. The new viewpoints should use the same node sets used by the viewpoints in the default views.
2. Opens the view and compares the two viewpoints, see Running Viewpoint Comparisons.
3. Creates a request if the comparison indicates that the source hierarchy contains nodes that the target hierarchy does not contain, see Making Changes Using Requests.
4. Uses the comparison results to drag and drop the missing nodes from the source to the target hierarchy, see Correcting Node Differences From Compare Results.
5. Submits the request.

User Detailed Steps

To perform this procedure, an administrator must have performed the procedure in Administrator Steps.

1. Create a view, then add viewpoints for both hierarchies by performing the following steps:
   a. Click **Definition**.
   b. Click **Create**.
   c. From **Application Dimension**, select the Cost Center dimension for the Source Hierarchy application.
   d. Enter Source Hierarchy, and then click **Create**.
   e. Click **Edit**.
   f. Select Cost Center from **Node Set**.
   g. Click **Save**.
   h. On the bottom of the page, click the tab containing the name of the view.
   i. Click **Definition**.
   j. Click **Create**.
   k. From **Application Dimension**, select the Cost Center dimension for the Target Hierarchy application.
   l. Enter Target Hierarchy, and then click **Create**.
   m. Click **Edit**.
   n. Select Cost Center from **Node Set**.
   o. Click **Save**, and then click **Close**.
2. Open the view.

3. From the left pane, select the **Side by Side** button .

   The two viewpoints display side by side. The **Source Hierarchy** viewpoint is on the left.

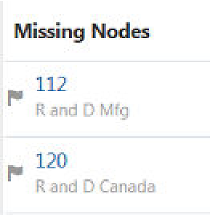
4. Perform the following steps in the **Compare Viewpoints** section:

   a. Select Left to Right from the first dropdown list.

   b. Select Bottom Nodes from the second dropdown list.

5. Click ▶ to run the comparison.

   The **Missing Nodes** section lists nodes named 112 and 120, as shown in the following example:



Missing Nodes

112
R and D Mfg

120
R and D Canada

6. Add the missing nodes to the target hierarchy by taking the following steps:

   a. Click **New Request**.

   b. In the **Missing Nodes** section, select the node named 112.

      The node is highlighted in the source hierarchy.

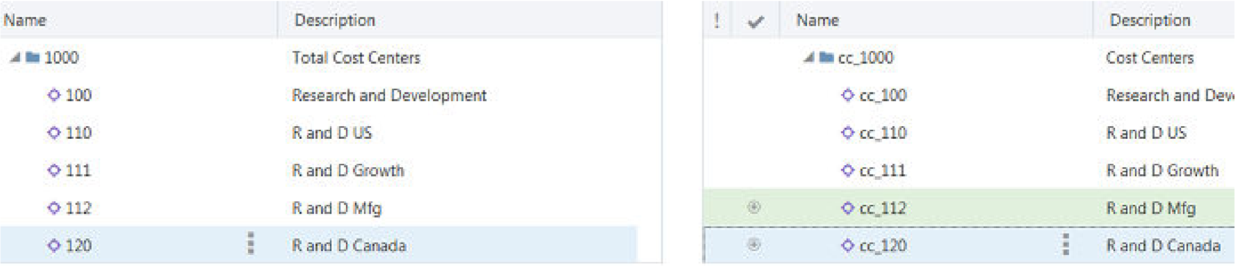
   c. Drag and drop the 112 node from the source hierarchy onto the top node of the target hierarchy.

   d. In the **Missing Nodes** section, select the node named 120.

      The node is highlighted in the source hierarchy.

   e. Drag and drop the 120 node from the source hierarchy onto the top node of the target hierarchy.

      The two hierarchies now have identical leaf nodes, as shown in the following example:



Name	Description
1000	Total Cost Centers
100	Research and Development
110	R and D US
111	R and D Growth
112	R and D Mfg
120	R and D Canada

!	✓	Name	Description
		cc_1000	Cost Centers
		cc_100	Research and Dev
		cc_110	R and D US
		cc_111	R and D Growth
	⊕	cc_112	R and D Mfg
	⊕	cc_120	R and D Canada

Note:

The prefix cc_ was automatically added to the nodes of the target hierarchy. This is because the node type for the Target Hierarchy application has a node type qualifier of cc_.

7. Click ⟳ to rerun the comparison to verify that all **Source Hierarchy** leaf nodes are in the **Target Hierarchy** viewpoint.

   The **Missing Nodes** section should display a message that says **No Differences to Display**.

8. Click **Submit**.

Outcome

You have created node type converters, compared nodes in two viewpoints to find the differences, and created and submitted a request to add any missing nodes.

45
Managing Enterprise Data as Lists

This scenario describes how to synchronize segment values from transactional systems such as general ledger accounts, cost centers, and departments.

It can also be used for systems that do not require a hierarchy, but use the lowest level of a domain, such as reference data lists and enterprise data like employees.

Administrator Steps

Administrator Prerequisites

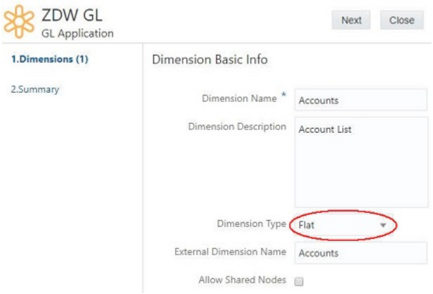
You must do an extract from the system that includes all of the segment values or reference data that you want to import and manage in Oracle Fusion Cloud Enterprise Data Management, see Import and Export File Format for User Defined Dimensions.

Administrator Process

1. From your external application, extract segment values or reference data into a file.
2. Register the external application using the Universal application wizard, see Working with Universal Applications.
3. Import the file using the import wizard or a request to load to the viewpoint.
4. Give your user the *Data Manager* permission to the Cloud EDM application.

Administrator Detailed Steps

1. Register your dimension and select **Flat** as the dimension type.



Dimension Basic Info

Dimension Name * Accounts

Dimension Description Account List

Dimension Type Flat

External Dimension Name Accounts

Allow Shared Nodes □

2. After the default view is created, do one of the following:
   - Use the Import wizard if no data currently exists for the dimension, see Importing Dimensions
   - Open the viewpoint in the default view, create a request and then load the file to the viewpoint if you are adding data to an existing dimension, see Making Changes Using a Load File
3. Give your user the name of the default view and *Data Manager* permission to the application, see User Steps

User Steps

User Prerequisites

You need *Data Manager* permissions to the application.

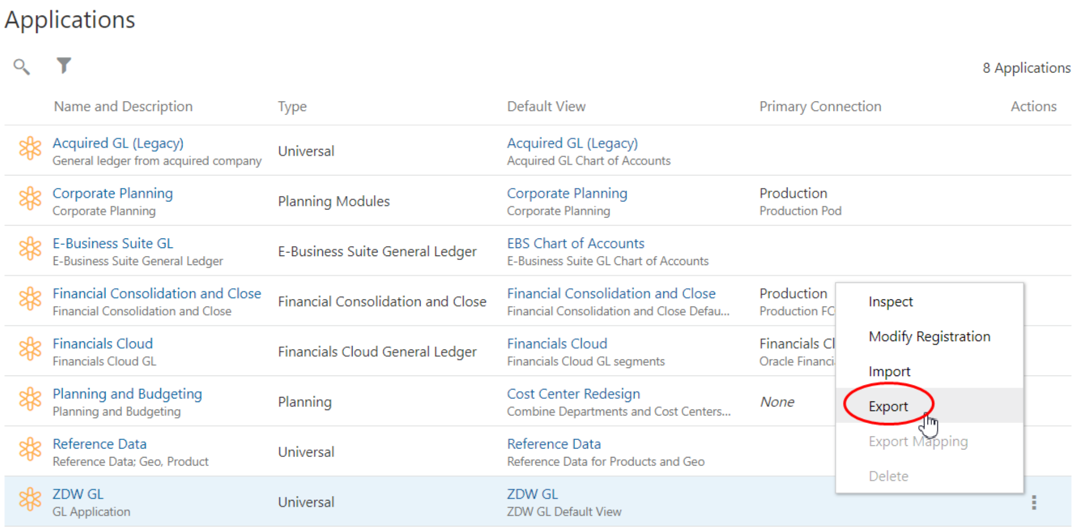
User Process

The user manages data and exports data to external applications.

1. Update nodes and properties as necessary, see Making Changes Interactively.
2. Export the bottom level nodes, see Exporting Dimensions.
3. Import the bottom level nodes into your external application.

User Detailed Steps

1. Update the data in Oracle Fusion Cloud Enterprise Data Management using requests.
2. Before you export, do the following:
   - Ensure that the dimension is bound, see Understanding Binding Rules.
   - Validate the viewpoint, see Validating a Viewpoint, Resolving Validation Issues, and Understanding Validations and Constraints.
3. Export the information to a file for use with target applications such as a general ledger application.
   To do this, on the **Applications** screen, click the Action menu next to the application, and then click **Export**.

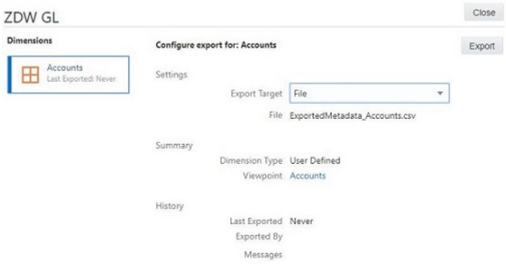


Applications

8 Applications

Name and Description	Type	Default View	Primary Connection	Actions
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts		
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning	Production Production Pod	
E-Business Suite GL E-Business Suite General Ledger	E-Business Suite General Ledger	EBS Chart of Accounts E-Business Suite GL Chart of Accounts		
Financial Consolidation and Close Financial Consolidation and Close	Financial Consolidation and Close	Financial Consolidation and Close Financial Consolidation and Close Defau...	Production Production FC	
Financials Cloud Financials Cloud GL	Financials Cloud General Ledger	Financials Cloud Financials Cloud GL segments	Financials Cl Oracle Financi	
Planning and Budgeting Planning and Budgeting	Planning	Cost Center Redesign Combine Departments and Cost Centers...	*None*	
Reference Data Reference Data: Geo, Product	Universal	Reference Data Reference Data for Products and Geo		
ZDW GL GL Application	Universal	ZDW GL ZDW GL Default View		⋮

4. In **Export Target**, select **File**.



Configure export for: Accounts

Export Target: File
File: ExportedMetadata_Accounts.csv

Dimension Type: User Defined
Viewpoint: Accounts

Last Exported: Never
Exported By:
Messages:

5. After the file has been exported, you can import it into your external application.

Outcome

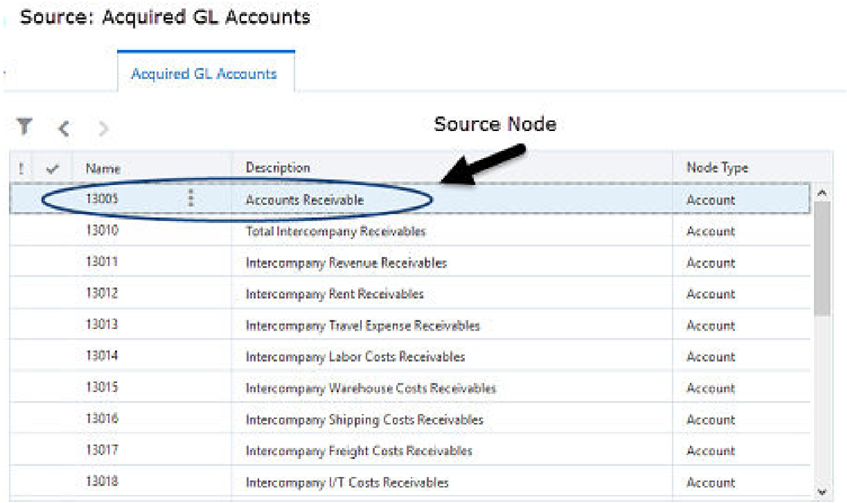
You now have an export of just base-level nodes from your Oracle Fusion Cloud Enterprise Data Management application which can be imported into target applications to fulfill use cases for reference data management and management of flat lists such as segment values for general ledgers.

46 Mapping Source Dimensions to Target Dimensions

Suppose you have an acquired company with a general ledger, and you want to map the general ledger accounts to planning accounts. Creating a maintenance view enables you to define the mapped data and import it back to Planning using Data Management. In this scenario you use three viewpoints, one for general ledger which is the source, one for Planning which is the target, and one for mapping the general ledger accounts to the Planning accounts.

> **Note:**
>
> You can map bottom level source nodes to bottom level target nodes only.

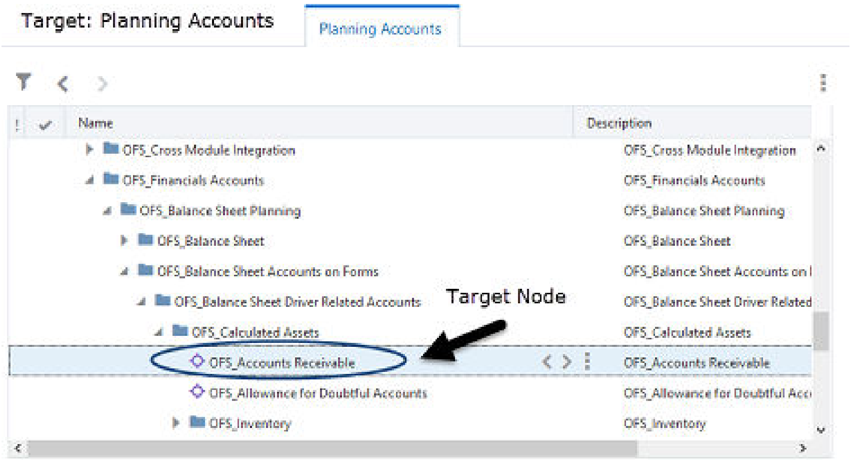
Your source is the Acquired GL application and the Account dimension shown below:



Source: Acquired GL Accounts

!	✓	Name	Description	Node Type
		13005	Accounts Receivable	Account
		13010	Total Intercompany Receivables	Account
		13011	Intercompany Revenue Receivables	Account
		13012	Intercompany Rent Receivables	Account
		13013	Intercompany Travel Expense Receivables	Account
		13014	Intercompany Labor Costs Receivables	Account
		13015	Intercompany Warehouse Costs Receivables	Account
		13016	Intercompany Shipping Costs Receivables	Account
		13017	Intercompany Freight Costs Receivables	Account
		13018	Intercompany I/T Costs Receivables	Account

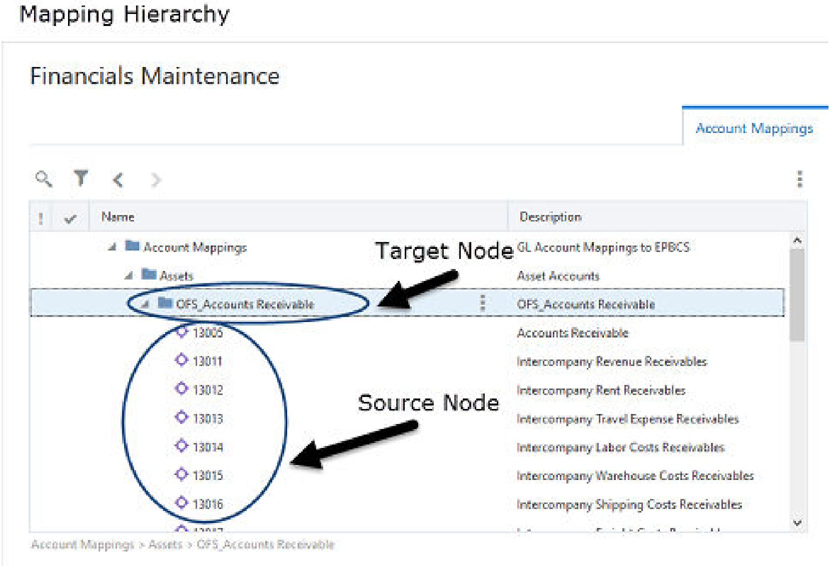
Your target is the Planning application and the Account dimension shown below:



Target: Planning Accounts

Name	Description
OFS_Cross Module Integration	OFS_Cross Module Integration
OFS_Financials Accounts	OFS_Financials Accounts
OFS_Balance Sheet Planning	OFS_Balance Sheet Planning
OFS_Balance Sheet	OFS_Balance Sheet
OFS_Balance Sheet Accounts on Forms	OFS_Balance Sheet Accounts on I
OFS_Balance Sheet Driver Related Accounts	OFS_Balance Sheet Driver Related
OFS_Calculated Assets	OFS_Calculated Assets
OFS_Accounts Receivable	OFS_Accounts Receivable
OFS_Allowance for Doubtful Accounts	OFS_Allowance for Doubtful Acc
OFS_Inventory	OFS_Inventory

Within the mapping viewpoint or hierarchy, add target nodes which are planning accounts. Then add source nodes which are general ledger accounts as shown below.



Mapping Hierarchy

Financials Maintenance

Name	Description
Account Mappings	GL Account Mappings to EPBCS
Assets	Asset Accounts
OFS_Accounts Receivable	OFS_Accounts Receivable
13005	Accounts Receivable
13011	Intercompany Revenue Receivables
13012	Intercompany Rent Receivables
13013	Intercompany Travel Expense Receivables
13014	Intercompany Labor Costs Receivables
13015	Intercompany Warehouse Costs Receivables
13016	Intercompany Shipping Costs Receivables

Account Mappings > Assets > OFS_Accounts Receivable

Process

- These are the administrator steps:
  - Register the source and target applications.

    > **Note:**
    >
    > You need the Application - Create role to register applications.

  - Import the source and target dimensions.
  - Create a map binding to map the source and target dimensions. When you create the map binding, a new hierarchy set, node set, and mapping viewpoint are created.
  - Define a mapping key.

  - Create a maintenance view that contains viewpoints for the source dimension, target dimension, and the mapping viewpoint that was created when you created the map binding.

    > **Note:**
    >
    > You need the Views - Create role to create the view.

  - For the user
    - Grant the user *Data Manager* permission to the target application, so the user can update the mapping viewpoint.
    - Give the user the mapping key location name.
- These are the user steps:
  - You need *Data Manager* permission to the target application, to update the mapping viewpoint.
  - Use the maintenance view your administrator sets up.
  - Map nodes using the mapping viewpoint. Add parent nodes from the target application and child nodes from the source application.
  - Export the mapping data using the mapping key location name.
  - Open Planning, and then use the Data Management module to import the mapping data. Use the location name your administrator defined for the mapping key.

Administrator Steps

As an administrator, set up a view with the source, target and mapping viewpoint for the user to work on as they map data.

Administrator Prerequisites

You must have the *Application - Create* role to register the applications. After you register the applications, you are assigned the *Owner* permission to the application and default view. You must also have the *Views - Create* role to create a maintenance view and viewpoints.

Administrator Process

1. Register and import the source and target applications.
2. Create a map binding. This also creates a hierarchy set, node set, and mapping viewpoint.
3. Create a maintenance view with viewpoints containing the source dimension, target dimension, and mapping viewpoint.
4. Define a mapping key.
5. Grant the *Data Manager* permission to the target application, and give the user the mapping key location name.

Administrator Detailed Steps

Register and import applications and dimensions

To register and import applications and dimensions:

1. Register the general ledger application, and import the general ledger account dimension, which is the source dimension, see Working with Universal Applications.

   > **Note:**
   >
   > A default view, viewpoint, node set, hierarchy set, and node type are created with the name GL Account. The GL Account viewpoint, node set, hierarchy set, and node type are bound to the GL Account dimension.

2. Register the Planning application and import the planning account dimension, which is the target dimension, see Working with Planning and FreeForm Applications.

   > **Note:**
   >
   > A default view, viewpoint, node set, hierarchy set, and node type are created with the name Planning Account. The Planning Account viewpoint, node set, hierarchy set, and node type are bound to the Planning Account dimension.

Create source node types for the map binding.

> **Note:**
>
> You must create new source node types because you cannot use the source node types from the source applications. Node types are specific to dimension and applications.

1. Create node types, using the target dimension, to hold the source nodes in the mapping viewpoint, see Creating a Node Type.
2. Create node type converters to convert the node types from each source application to the node types you created in step 1 above. See Working with Node Type Converters.
3. Optionally, add the `Core.Change.Sign` property to the source node types if you are mapping nodes which require a sign change when transferring numerical data across applications (for example, if you are mapping an asset to a liability).

Create a map binding

Create the map binding to bind source nodes to target nodes. See Creating a Map Binding.

When you add node types to the map binding, add all of the target node types that you want to map to as well as the source node types that you created for the mapping viewpoint.

Set up a maintenance view

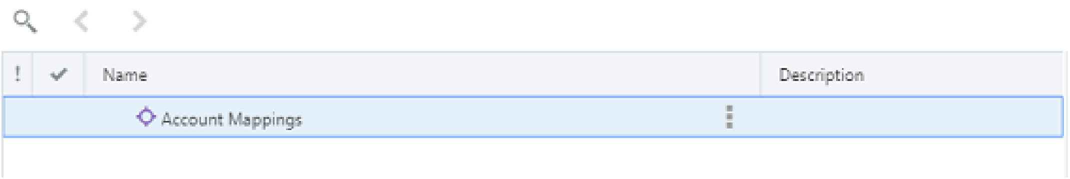
Set up a maintenance view to contain the source and target viewpoints.

1. Create the maintenance view, see Creating a View.
2. Add the source viewpoint with the GL Account node set.
3. Add the target viewpoint with the Planning Account node set.

> **Note:**
>
> See Creating a Viewpoint

4. Add the mapping viewpoint that was created when you created the map binding.
5. Create a top node for the mapping viewpoint using the target node type.
   This keeps the number of top nodes at a minimum. For example name it Account Mappings, then you can drag and drop target accounts under it.

   This image shows the mapping viewpoint with the top mode Account Mappings.



!	✓	Name	Description
		Account Mappings	

Bind the target viewpoint to the target dimension

Before you can create the mapping key and export mapping data using the maintenance view, it needs to contain both the mapping viewpoint and a viewpoint bound to the target dimension. In the maintenance view, the target viewpoint was created outside of application registration and by default is not bound to the target dimension. To bind the target viewpoint, edit the target dimension's existing binding and change it to point to the target viewpoint. See Identifying Objects Bound to Dimensions.

Create the mapping key

The mapping key defines which source nodes types are mapped to target node types. To define the mapping key:

1. Create a Mapping Key, to define which nodes types are mapped to each other. Within the mapping viewpoint specify the source node type: Acquired GL Account and the target node type: Planning Account, see Defining Mapping Keys.
2. Set the **Location Name** to `MapGLtoPlanning`.

Give Information to Your User

Give the following information to your user:

- The name of the maintenance view to work in.
- *Participant(Write)* permission to the target application, see Assigning Application Permissions.
- The mapping key location name, `MapGLtoPlanning`, to use when exporting.

> **Note:**
>
> For user steps, see User Steps.

User Steps

As a user, within the mapping viewpoint add target nodes and then map source nodes to the target nodes. Then you can export the mapping viewpoint back to Planning using Data Management.

User Prerequisites

The administrator registers and imports the applications; creates a maintenance view with three viewpoints for the source account, target account, and mapping viewpoint; and defines the mapping key. Your administrator needs to give you the mapping key location name, `MapGLtoPlanning`, to export the mapping data. Then from Planning you can import the mapping data.

You must have *Data Manager* permission to the target application in order to export the mapping data.

User Process

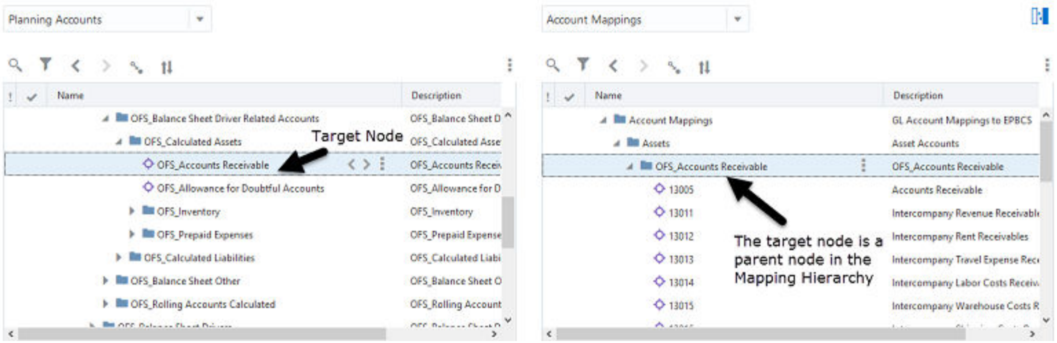
1. Update the mapping viewpoint with parents from the target application and children from the source application.
2. Export the mapping data.
3. Import the mapping data to Planning.

User Detailed Steps

Map bottom level nodes

To map bottom level nodes:

1. Review and decide which source Acquired General Ledger account nodes you want to map to planning account nodes.
2. Use the side-by-side layout with the planning accounts on the left and the mapping viewpoint on the right.
3. Drag and drop planning nodes as parents into the mapping viewpoint.



Planning Accounts

Name	Description
OFS_Balance Sheet Driver Related Accounts	OFS_Balance Sheet D
OFS_Calculated Assets	OFS_Calculated Asse
OFS_Accounts Receivable	OFS_Accounts Recei
OFS_Allowance for Doubtful Accounts	OFS_Allowance for D
OFS_Inventory	OFS_Inventory
OFS_Prepaid Expenses	OFS_Prepaid Expense
OFS_Calculated Liabilities	OFS_Calculated Liabi
OFS_Balance Sheet Other	OFS_Balance Sheet O
OFS_Rolling Accounts Calculated	OFS_Rolling Account

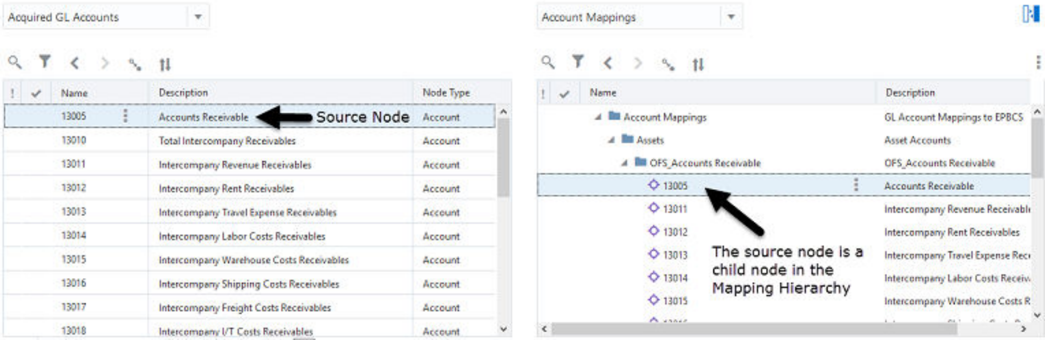
Account Mappings

Name	Description
Account Mappings	GL Account Mappings to EPBCS
Assets	Asset Accounts
OFS_Accounts Receivable	OFS_Accounts Receivable
13005	Accounts Receivable
13011	Intercompany Revenue Receivabl
13012	Intercompany Rent Receivables
13013	Intercompany Travel Expense Rec
13014	Intercompany Labor Costs Receiv
13015	Intercompany Warehouse Costs R

Target Node

The target node is a parent node in the Mapping Hierarchy

4. Use the side-by-side layout with the Acquired GL Accounts on the left and the mapping viewpoint on the right.
5. Drag and drop Acquired General Ledger nodes as children into the mapping viewpoint.



Acquired GL Accounts

!	✓	Name	Description	Node Type
		13005	Accounts Receivable	Account
		13010	Total Intercompany Receivables	Account
		13011	Intercompany Revenue Receivables	Account
		13012	Intercompany Rent Receivables	Account
		13013	Intercompany Travel Expense Receivables	Account
		13014	Intercompany Labor Costs Receivables	Account
		13015	Intercompany Warehouse Costs Receivables	Account
		13016	Intercompany Shipping Costs Receivables	Account
		13017	Intercompany Freight Costs Receivables	Account
		13018	Intercompany I/T Costs Receivables	Account

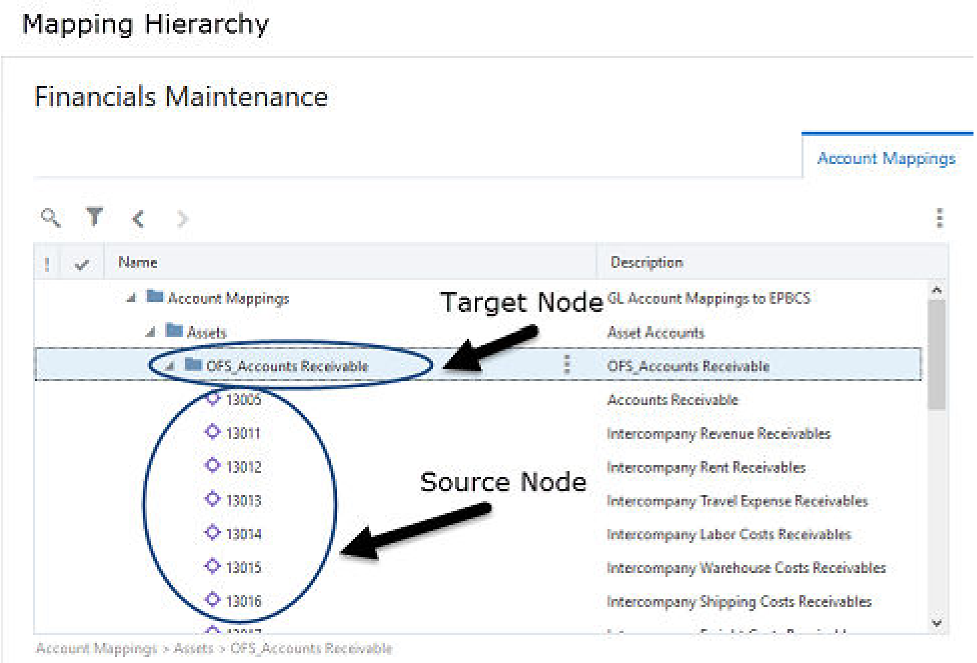
Source Node

Account Mappings

!	✓	Name	Description
		Account Mappings	GL Account Mappings to EPBCS
		Assets	Asset Accounts
		OFS_Accounts Receivable	OFS_Accounts Receivable
		13005	Accounts Receivable
		13011	Intercompany Revenue Receivabl
		13012	Intercompany Rent Receivables
		13013	Intercompany Travel Expense Rec
		13014	Intercompany Labor Costs Receiv
		13015	Intercompany Warehouse Costs R

The source node is a child node in the Mapping Hierarchy

6. Map as many nodes as desired.



Mapping Hierarchy

Financials Maintenance

Account Mappings

!	✓	Name	Description
		Account Mappings	GL Account Mappings to EPBCS
		Assets	Asset Accounts
		OFS_Accounts Receivable	OFS_Accounts Receivable
		13005	Accounts Receivable
		13011	Intercompany Revenue Receivables
		13012	Intercompany Rent Receivables
		13013	Intercompany Travel Expense Receivables
		13014	Intercompany Labor Costs Receivables
		13015	Intercompany Warehouse Costs Receivables
		13016	Intercompany Shipping Costs Receivables

Target Node

Source Node

Account Mappings > Assets > OFS_Accounts Receivable

Export the mapping data

To export mapping data:

1. Validate the mapping viewpoint, see Validating a Viewpoint.
2. Export the mapping data to a local file or to the Data Management inbox, see Exporting Mapping Data.

> **Note:**
>
> When you export, use the location name, `MapGLtoPlanning` your administrator defined for the mapping key.

Import the mapping data using the Data Management inbox

Within Planning, open the Data Management module and navigate to the inbox, see Importing Member Mappings in *Administering Data Management*

During the import the negative sign is applied to values in the target node for the selected mapping key which is the location in Data Management

.
To import mapping data:

1. Open your external application and navigate to the Data Management inbox.
2. Click **Data Load Mapping**.
3. At the bottom of the screen, click **DelimitedLocation**.
4. In the **Location** field, enter `MapGLtoPlanning`, which is the Location Name your administrator set up for the mapping key in Oracle Fusion Cloud Enterprise Data Management.
5. Click **Import > Current Dimension**.
6. Select the mapping data file you exported.

47

Migrating Enterprise Data from Data Relationship Management to Cloud EDM

This scenario demonstrates how to migrate your Enterprise Data from an Oracle Data Relationship Management application to Oracle Fusion Cloud Enterprise Data Management using the Universal application wizard. The steps walk you through registering your Data Relationship Management application as a Universal application and importing your dimensional structures into the application.

While the *Data Manager* role plays an integral part during the management of the enterprise data assets, the steps in this scenario are focused on the administrative tasks involved to complete the register to export process within Cloud EDM.

> **Note:**
>
> Even though this scenario is written for the Data Relationship Management application, you can use the same steps to connect to other types of applications using the Universal application wizard.
>
> Data Relationship Management 11.1.2.4.330 or higher is supported for on-premise to cloud migration.

Administrator Steps

Administrator Prerequisites

A dimensional export from your Oracle Data Relationship Management application is required when using the Universal application wizard. You must prepare a Data Relationship Management dimension export that includes all the dimensionality you want to import and manage in Oracle Fusion Cloud Enterprise Data Management. Create this export and run the job so that the resulting .CSV file is generated and placed into a location you can connect to with Cloud EDM. You'll import this file into Cloud EDM.

You need the Application - Create role to register the application. After you register an application you are automatically assigned the Owner permission for the application and the default view.

Administrator Process

1. Register your Data Relationship Management application, see Working with Universal Applications.
2. Create an export in Data Relationship Management.
3. Import the registered dimensions, see Understanding Validations and Constraints.
4. Provide your users with the appropriate access to the data chain elements needed to manage the dimensionality from the Data Relationship Management application, see Working with Roles and Permissions.

5. Export dimensionality after changes have been made by your users, see Exporting a User Defined Dimension.

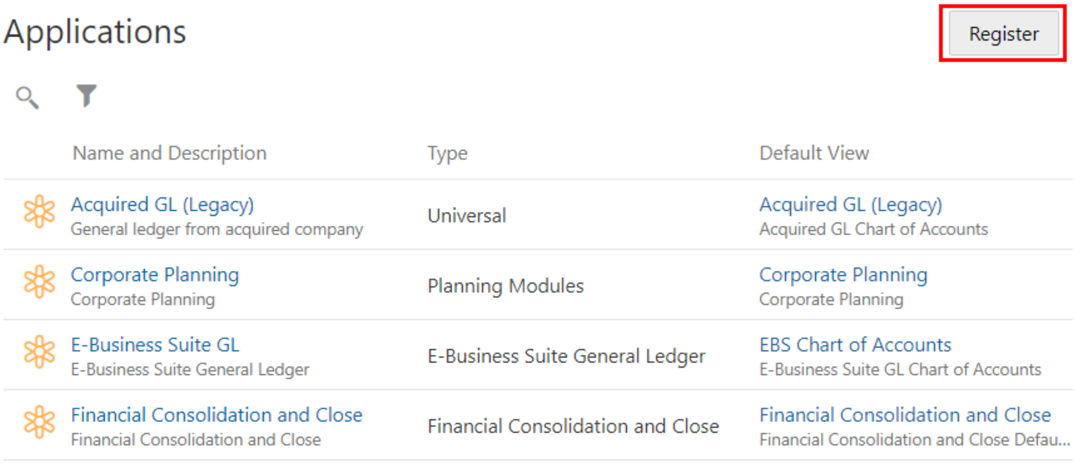
Detailed Steps

Related Topics

- Step 1: Register your Data Relationship Management Application
- Step 2: Create an Export File in Data Relationship Management
- Step 3: Import Dimensions

Step 1: Register your Data Relationship Management Application

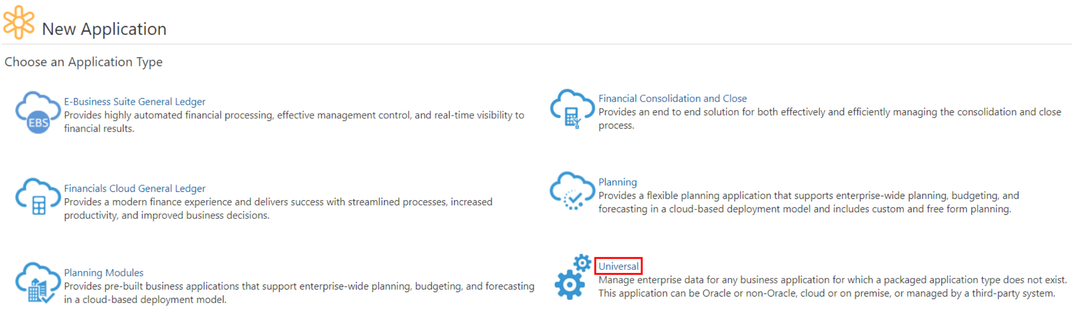
1. From Applications, click **Register**.



Applications

Name and Description	Type	Default View
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning
E-Business Suite GL E-Business Suite General Ledger	E-Business Suite General Ledger	EBS Chart of Accounts E-Business Suite GL Chart of Accounts
Financial Consolidation and Close Financial Consolidation and Close	Financial Consolidation and Close	Financial Consolidation and Close Financial Consolidation and Close Defau...

2. Select the **Universal** application type.



New Application

Choose an Application Type

E-Business Suite General Ledger
Provides highly automated financial processing, effective management control, and real-time visibility to financial results.

Financial Consolidation and Close
Provides an end to end solution for both effectively and efficiently managing the consolidation and close process.

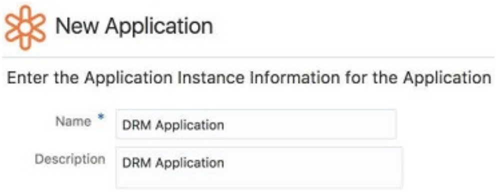
Financials Cloud General Ledger
Provides a modern finance experience and delivers success with streamlined processes, increased productivity, and improved business decisions.

Planning
Provides a flexible planning application that supports enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model and includes custom and free form planning.

Planning Modules
Provides pre-built business applications that support enterprise-wide planning, budgeting, and forecasting in a cloud-based deployment model.

Universal
Manage enterprise data for any business application for which a packaged application type does not exist. This application can be Oracle or non-Oracle, cloud or on premise, or managed by a third-party system.

3. Provide a name and description for the application.



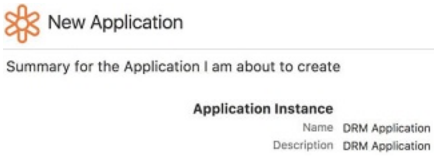
New Application

Enter the Application Instance Information for the Application

Name * DRM Application

Description DRM Application

4. Create your Oracle Data Relationship Management application.



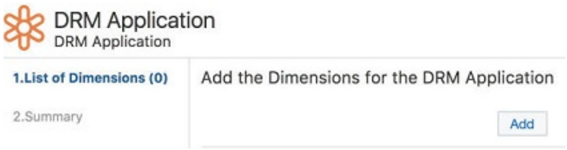
New Application

Summary for the Application I am about to create

Application Instance

Name DRM Application
Description DRM Application

5. Click **Add** to define the dimensions you want to import from your Data Relationship Management application.

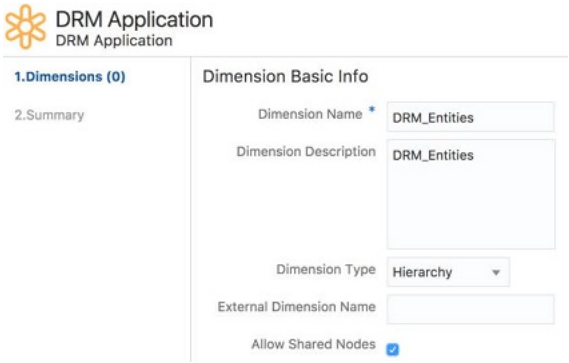


DRM Application
DRM Application

1.List of Dimensions (0)
2.Summary

Add the Dimensions for the DRM Application

6. Enter dimension information. For each dimension, determine if it will be a hierarchy or flat list, note the external dimension name in Data Relationship Management, and whether to allow shared nodes.



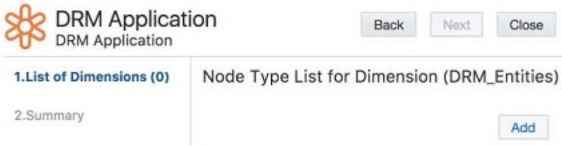
DRM Application
DRM Application

1.Dimensions (0)
2.Summary

Dimension Basic Info

Dimension Name * DRM_Entities
Dimension Description DRM_Entities
Dimension Type Hierarchy
External Dimension Name
Allow Shared Nodes ☑

7. For each dimension, click **Add** and define a node type list.

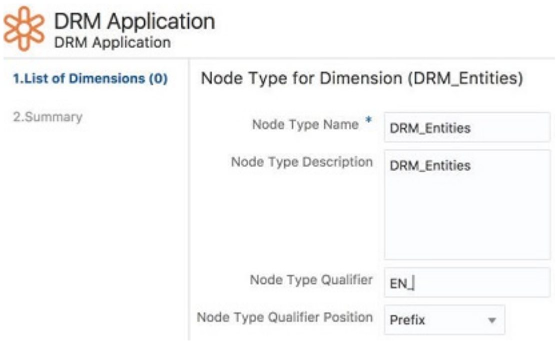


DRM Application
DRM Application

1.List of Dimensions (0)
2.Summary

Node Type List for Dimension (DRM_Entities)

8. Enter a name for each node type and a qualifier if needed.



Node Type for Dimension (DRM_Entities)

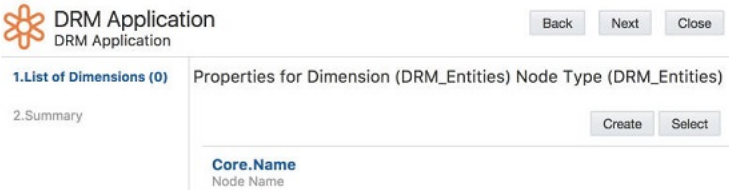
Node Type Name * DRM_Entities

Node Type Description DRM_Entities

Node Type Qualifier EN_

Node Type Qualifier Position Prefix

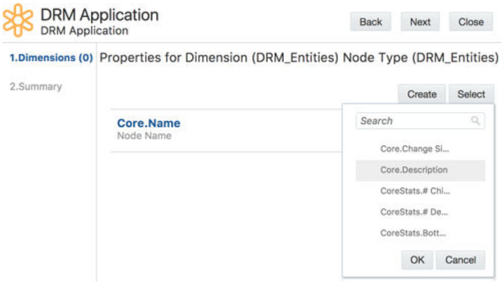
9. For each dimension, create properties for each node type.



Properties for Dimension (DRM_Entities) Node Type (DRM_Entities)

Core.Name
Node Name

10. Define which properties should be available to nodes of that node type. Either create new properties or select existing properties. Click **Next**.



Properties for Dimension (DRM_Entities) Node Type (DRM_Entities)

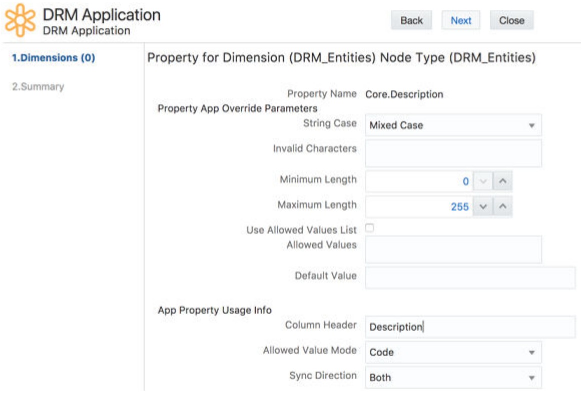
Core.Name
Node Name

Core.Change Si...
Core.Description
CoreStats.# Chi...
CoreStats.# De...
CoreStats.Bott...

11. Update property information and then click **Next**.

Note:

Column header is required.



Property for Dimension (DRM_Entities) Node Type (DRM_Entities)

Property Name Core.Description

Property App Override Parameters

String Case Mixed Case

Invalid Characters

Minimum Length 0

Maximum Length 255

Use Allowed Values List □

Allowed Values

Default Value

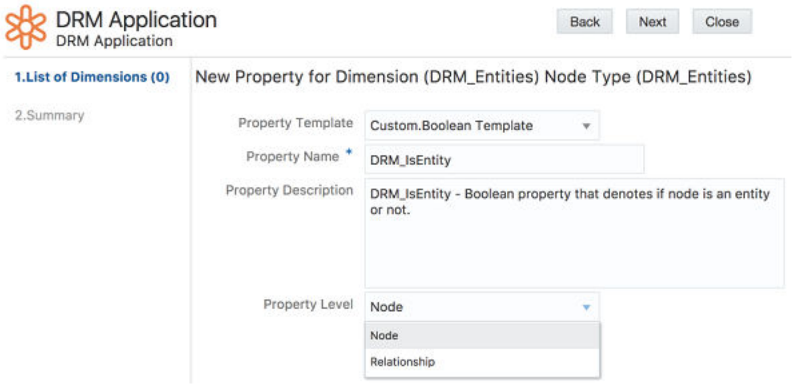
App Property Usage Info

Column Header Description

Allowed Value Mode Code

Sync Direction Both

12. Define the property name, description, valid values, and whether the property is at the node level or relationship level. Click **Next**.



New Property for Dimension (DRM_Entities) Node Type (DRM_Entities)

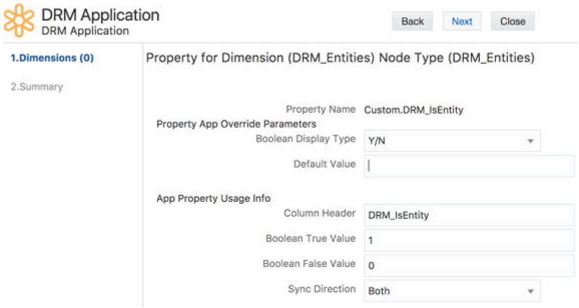
Property Template: Custom.Boolean Template

Property Name *: DRM_IsEntity

Property Description: DRM_IsEntity - Boolean property that denotes if node is an entity or not.

Property Level: Node

13. Define property override parameters and property usage information.



Property for Dimension (DRM_Entities) Node Type (DRM_Entities)

Property Name: Custom.DRM_IsEntity

Property App Override Parameters

Boolean Display Type: Y/N

Default Value:

App Property Usage Info

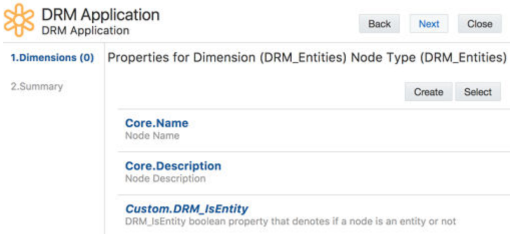
Column Header: DRM_IsEntity

Boolean True Value: 1

Boolean False Value: 0

Sync Direction: Both

14. Review the summary of properties for the dimension.



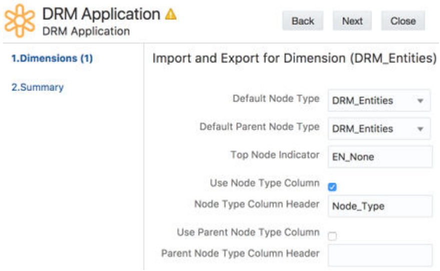
Properties for Dimension (DRM_Entities) Node Type (DRM_Entities)

Core.Name
Node Name

Core.Description
Node Description

Custom.DRM_IsEntity
DRM_IsEntity boolean property that denotes if a node is an entity or not

15. Include top nodes for each dimension and column headers for Node Type fields that will be in import and export file.



Import and Export for Dimension (DRM_Entities)

Default Node Type: DRM_Entities

Default Parent Node Type: DRM_Entities

Top Node Indicator: EN_None

Use Node Type Column: ☑

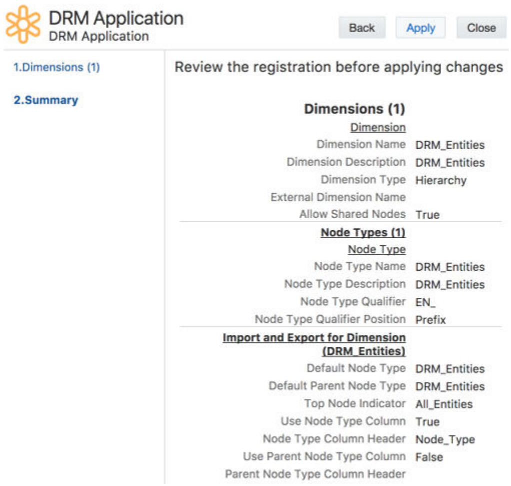
Node Type Column Header: Node_Type

Use Parent Node Type Column: ☐

Parent Node Type Column Header:

16. Repeat steps 5-15 to add any other dimensions to the registration, then click **Next**.

17. Review all the registration information and click Apply to register your Data Relationship Management application.



DRM Application
DRM Application

Review the registration before applying changes

Dimensions (1)

Dimension

Dimension Name	DRM_Entities
Dimension Description	DRM_Entities
Dimension Type	Hierarchy
External Dimension Name	
Allow Shared Nodes	True

Node Types (1)

Node Type

Node Type Name	DRM_Entities
Node Type Description	DRM_Entities
Node Type Qualifier	EN_
Node Type Qualifier Position	Prefix

Import and Export for Dimension (DRM_Entities)

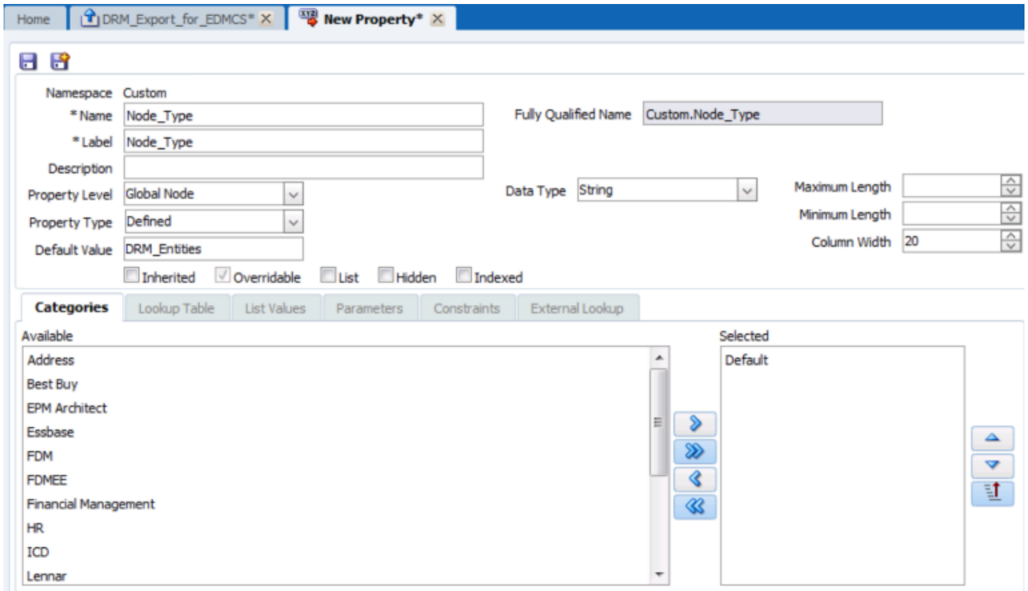
Default Node Type	DRM_Entities
Default Parent Node Type	DRM_Entities
Top Node Indicator	All_Entities
Use Node Type Column	True
Node Type Column Header	Node_Type
Use Parent Node Type Column	False
Parent Node Type Column Header	

Step 2: Create an Export File in Data Relationship Management

> **Note:**
>
> This step is performed in Oracle Data Relationship Management.

Prerequisites

If needed, create new properties, such as Oracle Fusion Cloud Enterprise Data Management node types in Data Relationship Management.



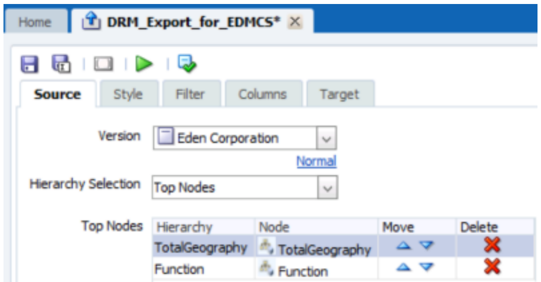
Namespace	Custom		
*Name	Node_Type	Fully Qualified Name	Custom.Node_Type
*Label	Node_Type		
Description			
Property Level	Global Node	Data Type	String
Property Type	Defined	Maximum Length	
Default Value	DRM_Entities	Minimum Length	
		Column Width	20

□ Inherited ☑ Overridable □ List □ Hidden □ Indexed

Available: Address, Best Buy, EPM Architect, Essbase, FDM, FDMEE, Financial Management, HR, ICD, Lennar

Selected: Default

1. In Data Relationship Management, create a new export.



Version: Eden Corporation

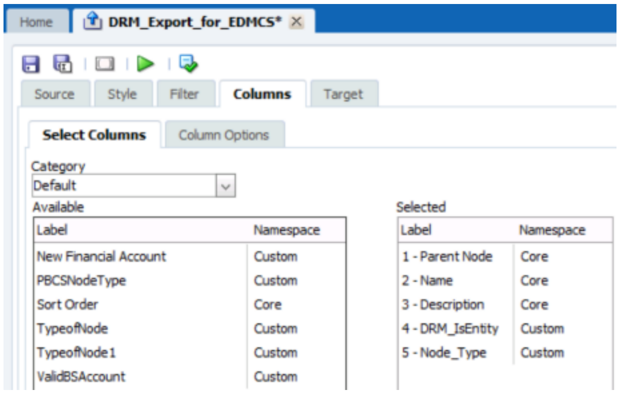
Normal

Hierarchy Selection: Top Nodes

Top Nodes

Hierarchy	Node	Move	Delete
TotalGeography	TotalGeography	△ ▽	✖
Function	Function	△ ▽	✖

2. Select the columns and fields required (based on what you set up for this dimension in the Universal application wizard).



Category: Default

Available

Label	Namespace
New Financial Account	Custom
PBCSNodeType	Custom
Sort Order	Core
TypeofNode	Custom
TypeofNode1	Custom
ValidBSAccount	Custom

Selected

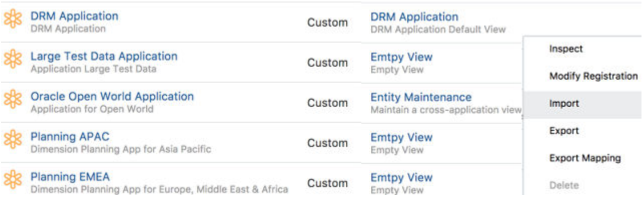
Label	Namespace
1 - Parent Node	Core
2 - Name	Core
3 - Description	Core
4 - DRM_IsEntity	Custom
5 - Node_Type	Custom

3. Export with column headings and comma delimited if importing into Cloud EDM with those selections.

Step 3: Import Dimensions

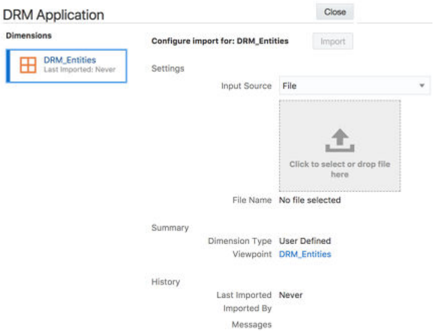
> **Note:**
>
> This step is performed in Oracle Fusion Cloud Enterprise Data Management.

1. From **Applications**, locate your new application and from the Action menu, select **Import**.



DRM Application DRM Application	Custom	DRM Application DRM Application Default View
Large Test Data Application Application Large Test Data	Custom	Emtpy View Empty View
Oracle Open World Application Application for Open World	Custom	Entity Maintenance Maintain a cross-application view
Planning APAC Dimension Planning App for Asia Pacific	Custom	Emtpy View Empty View
Planning EMEA Dimension Planning App for Europe, Middle East & Africa	Custom	Emtpy View Empty View

2. From **Import Source**, select **File** and then browse and select the file that you exported from Oracle Data Relationship Management.



DRM Application

Dimensions

DRM_Entities
Last Imported: Never

Configure import for: DRM_Entities

Settings

Input Source: File

File Name: No file selected

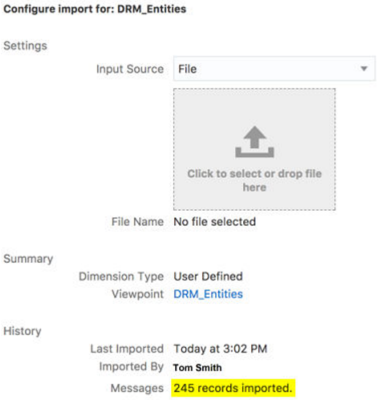
Summary

Dimension Type: User Defined
Viewpoint: DRM_Entities

History

Last Imported: Never
Imported By
Messages

A successful import will display the number of records imported.



Configure import for: DRM_Entities

Settings

Input Source: File

File Name: No file selected

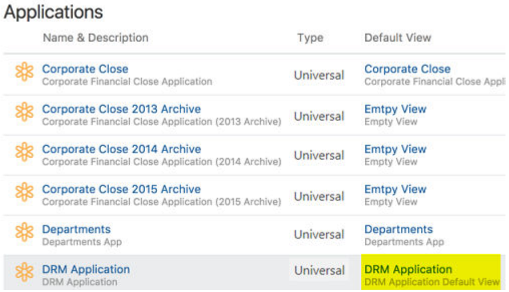
Summary

Dimension Type: User Defined
Viewpoint: DRM_Entities

History

Last Imported: Today at 3:02 PM
Imported By: Tom Smith
Messages: 245 records imported.

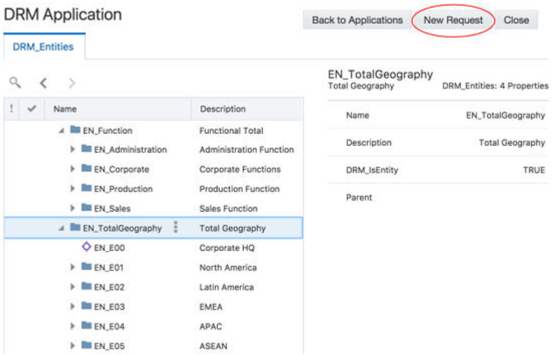
3. Close the import wizard and go to **Applications**, and select the default view for the application.



Applications

Name & Description	Type	Default View
Corporate Close Corporate Financial Close Application	Universal	Corporate Close Corporate Financial Close Appli
Corporate Close 2013 Archive Corporate Financial Close Application (2013 Archive)	Universal	Emtpy View Empty View
Corporate Close 2014 Archive Corporate Financial Close Application (2014 Archive)	Universal	Emtpy View Empty View
Corporate Close 2015 Archive Corporate Financial Close Application (2015 Archive)	Universal	Emtpy View Empty View
Departments Departments App	Universal	Departments Departments App
DRM Application DRM Application	Universal	DRM Application DRM Application Default View

4. Expand hierarchies to navigate to imported nodes and view property values. Give the view name to your users so they can begin maintenance actions, see User Steps.



DRM Application

DRM_Entities

!	✓	Name	Description
		EN_Function	Functional Total
		EN_Administration	Administration Function
		EN_Corporate	Corporate Functions
		EN_Production	Production Function
		EN_Sales	Sales Function
		EN_TotalGeography	Total Geography
		EN_E00	Corporate HQ
		EN_E01	North America
		EN_E02	Latin America
		EN_E03	EMEA
		EN_E04	APAC
		EN_E05	ASEAN

EN_TotalGeography
Total Geography
DRM_Entities: 4 Properties

Name	EN_TotalGeography
Description	Total Geography
DRM_IsEntity	TRUE
Parent	

Outcome

You have registered your Oracle Data Relationship Management application with Oracle Fusion Cloud Enterprise Data Management and can now migrate dimensions from Data

Relationship Management for management within Cloud EDM. You have imported the dimensionality you want to manage and can also export changes out of Cloud EDM for the Data Relationship Management application or other applications. You have full request activity audit information to know where, when, and by whom changes were made.

User Steps

User Prerequisites

Your administrator must follow the steps in this scenario to successfully register your Oracle Data Relationship Management application and import the desired enterprise data elements. After this is complete, you can proactively manage your enterprise data pursuant to your security privileges.

User Process

This is a high level summary of the tasks you can perform:

- Make maintenance views and viewpoints, see Working with Viewpoints.
- Use side-by-side layout and compare to rationalize nodes across applications, see Comparing Viewpoints.
- Update nodes and properties and update alternate hierarchies for "what if " analysis, see Making Changes Interactively.
- Map nodes from one hierarchy to another, see Creating Mapping Viewpoints.
- Load data to add new dimensions, such as accounts, see Making Changes Using a Load File.
- Browse draft and completed requests in the Requests list, see Working with Request Activity.

Outcome

You now have a central point of change management for your enterprise data that will feed from your Data Relationship Management and potentially back to the Data Relationship Management application as well as other external applications if desired.

48 Changing the Bound Viewpoint to Export an Alternate Viewpoint

This use case describes changing the viewpoint bound to a dimension so that you can export an alternate viewpoint.

Each dimension has one viewpoint that is bound to it. When you import a dimension, the objects in the viewpoint's bound data chain are updated. When you export to a dimension, the objects in your bound viewpoint are exported.

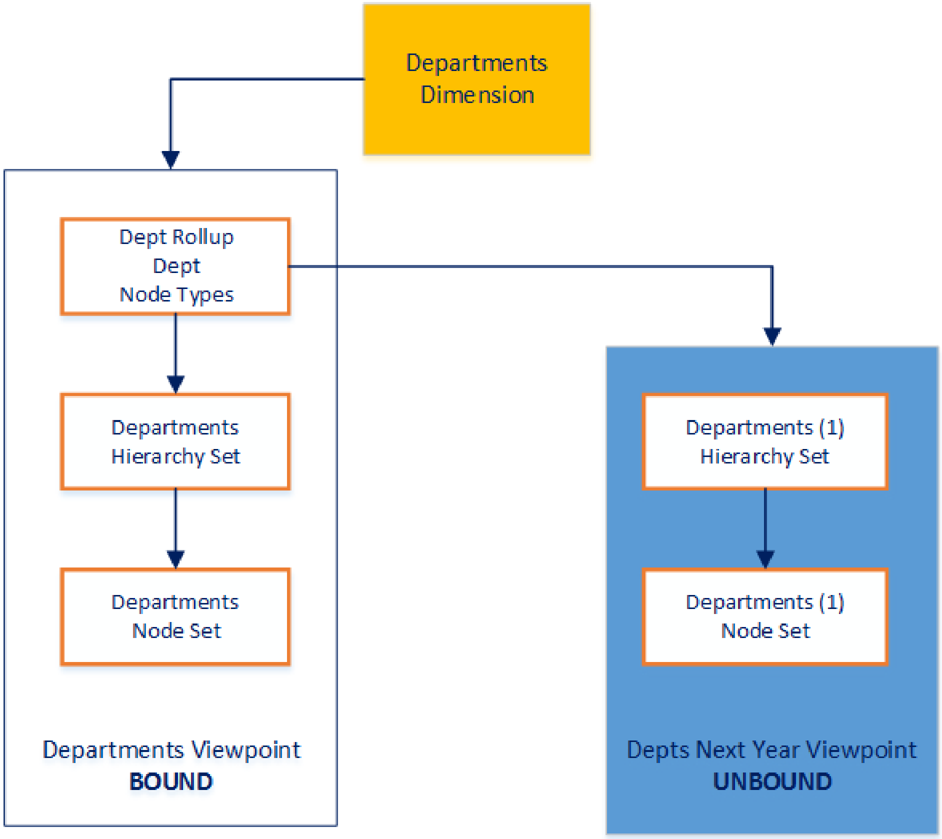
For more information see:

- Understanding Bindings and Bound Data Objects
- Defining Alternate Views and Viewpoints
- Copying a Viewpoint
- Changing the Viewpoint Bound to a User Defined Dimension in Universal Applications

Scenario

- Our company will be adding operations in Germany next year. We need to start planning for that change.
- We have an existing Departments viewpoint. This viewpoint is bound to the Departments dimension.
- We'll create a copy of the Departments viewpoint to use as a starting point for our Depts Next Year viewpoint.
- We'll make changes to the Depts Next Year viewpoint – adding the new Germany parent and its children.
- When we're ready to include the Depts Next Year viewpoint in our application, we'll change the viewpoint that's bound to the Departments dimension and export the dimension.

This diagram shows the data objects for the bound viewpoint and the new objects that will be created from copying the existing viewpoint and copying relationships only. We'll reuse the same node types that are used in the copied viewpoint.



Prerequisites

This use case assumes that:

- An application has been registered.
- The Departments dimension has been imported.
- A viewpoint named Departments is bound to the Departments dimension.

User Steps

User Prerequisites

You must have the *Owner* permission on both the application and the view in order to be able to copy viewpoints in that view.

User Process

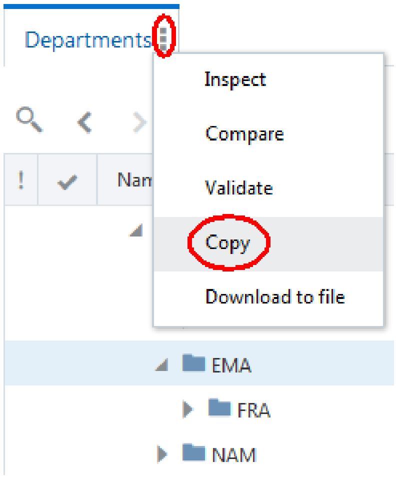
1. Create a new viewpoint by copying an existing viewpoint.
2. Use a request to make changes to the new viewpoint.
3. Bind the new viewpoint to the dimension.
4. Export the dimension to the external application.

Detailed Steps

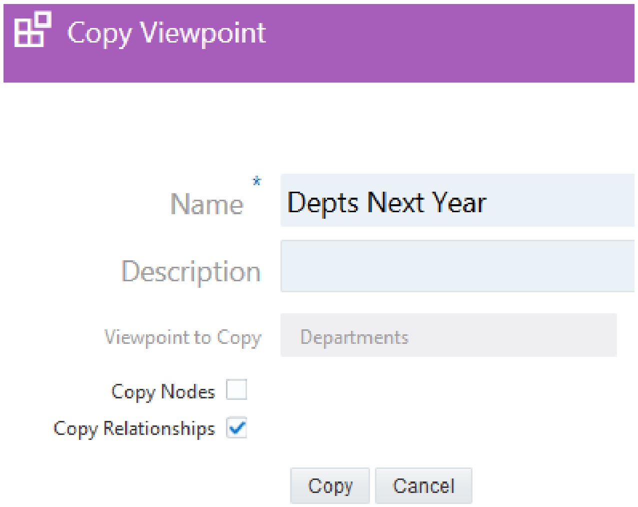
1. Create a new viewpoint by copying an existing viewpoint.

a. Open the Departments view and select Departments.

b. Click ⋮ and then select **Copy**.



c. Enter the name for the new viewpoint (**Depts Next Year)**, clear **Copy Nodes**, make sure that **Copy Relationships** is selected, and then click **Copy**.



Copy Viewpoint

Name * Depts Next Year

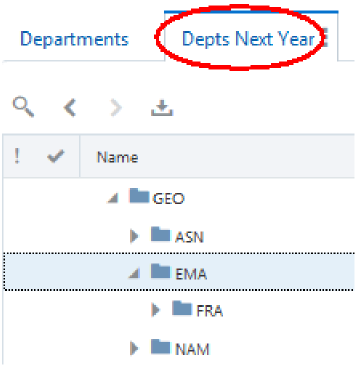
Description

Viewpoint to Copy Departments

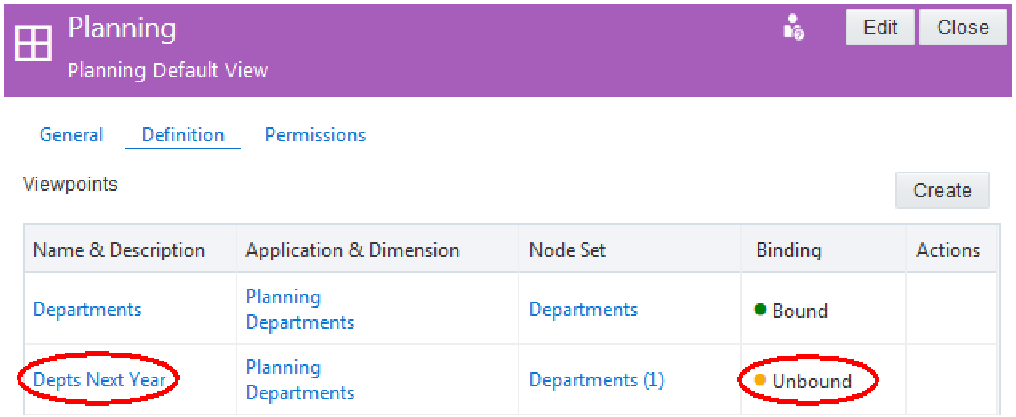
Copy Nodes □

Copy Relationships ☑

The new viewpoint is immediately available:



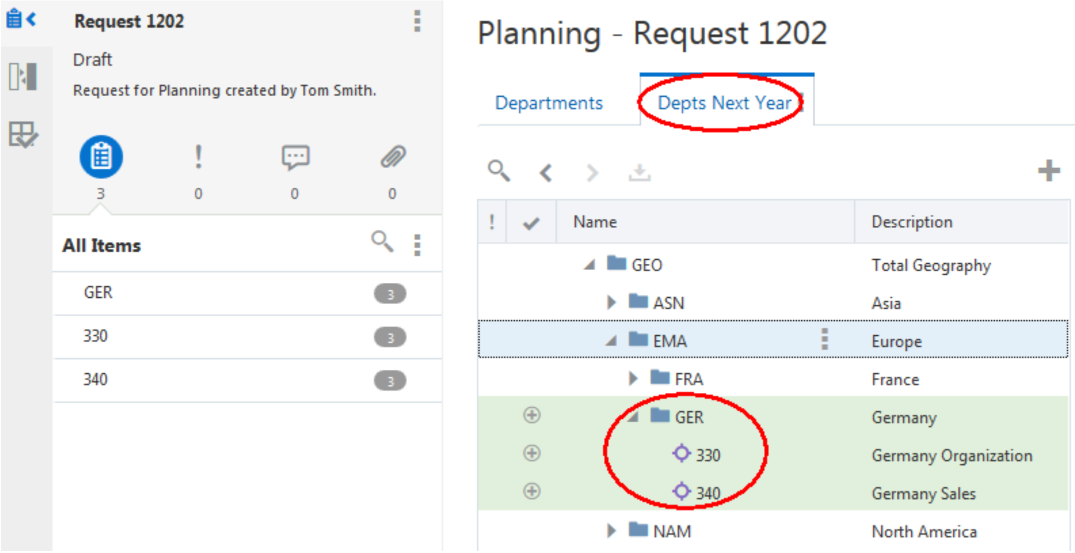
If you inspect the view where you created the new viewpoint, you'll see that the Depts Next Year viewpoint has been created and is not bound to the Departments dimension.



Name & Description	Application & Dimension	Node Set	Binding	Actions
Departments	Planning Departments	Departments	● Bound	
Depts Next Year	Planning Departments	Departments (1)	● Unbound	

2. Open the new viewpoint, create a new request, and make changes to the viewpoint.

   In this case, we add Germany as a child of EMA. Then add departments 330 and 340.



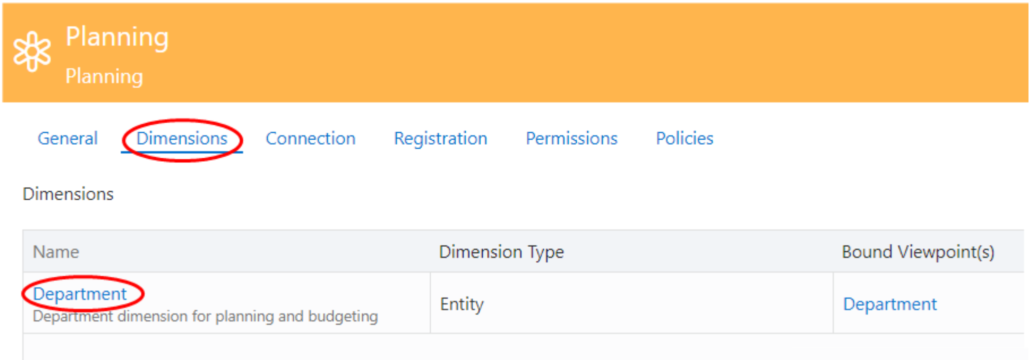
Name	Description
GEO	Total Geography
ASN	Asia
EMA	Europe
FRA	France
GER	Germany
330	Germany Organization
340	Germany Sales
NAM	North America

When we're satisfied that all changes have been made, we submit the request to commit the changes.

3. Bind the Depts Next Year viewpoint to the Departments dimension. Binding rules are enforced when you bind the new viewpoint to the dimension.

   a. From **Applications**, click **Planning**.

   b. Click **Dimensions** and then click the **Department** dimension.

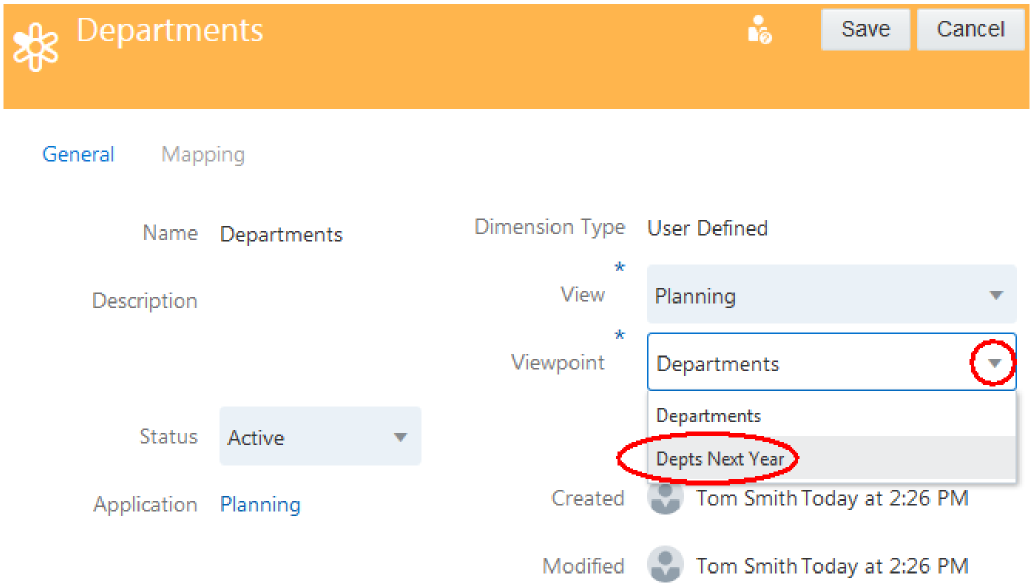


Dimensions

Name	Dimension Type	Bound Viewpoint(s)
Department Department dimension for planning and budgeting	Entity	Department

   c. Click **Edit**.

   d. From **Viewpoint**, select **Depts Next Year** and then click **Save**.



Name Departments

Description

Status Active

Application Planning

Dimension Type User Defined

* View Planning

* Viewpoint Departments

Departments

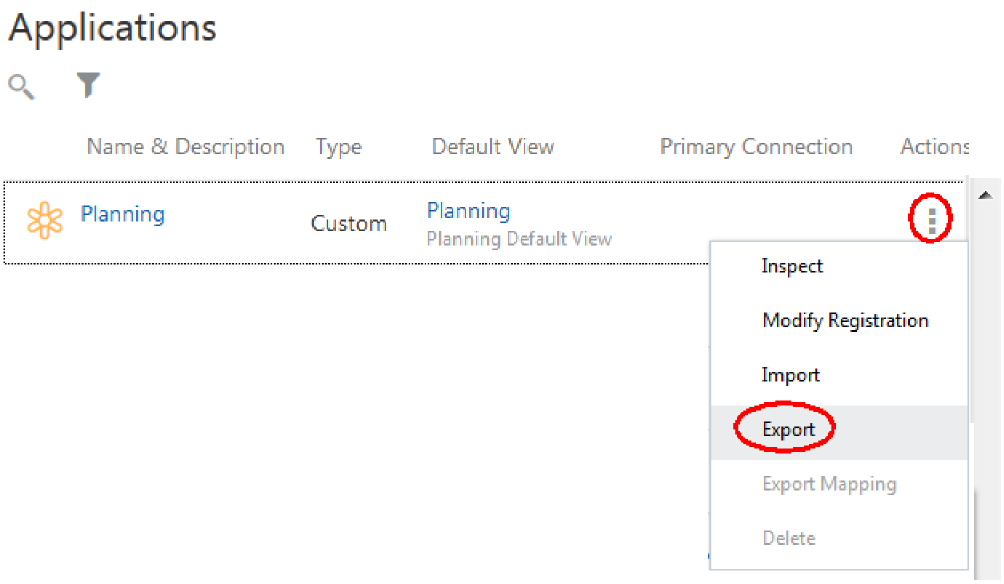
Depts Next Year

Created Tom Smith Today at 2:26 PM

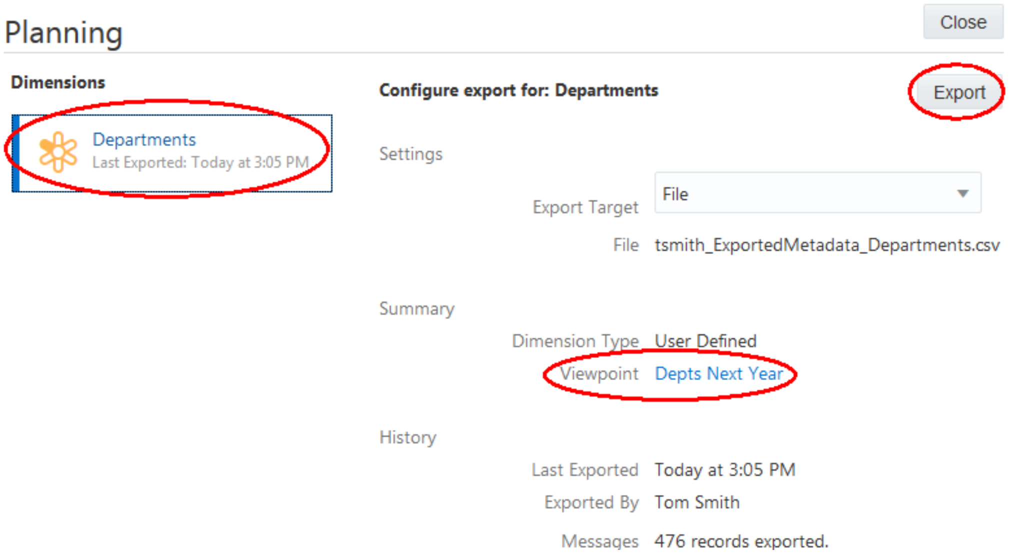
Modified Tom Smith Today at 2:26 PM

4. Export the Departments dimension to the external application.

   a. From **Applications**, select **Planning**, click ⋮ and then select **Export**.



b. Click the **Departments** dimension and then click **Export**.



Configure export for: Departments

Settings

Export Target: File

File: tsmith_ExportedMetadata_Departments.csv

Summary

Dimension Type: User Defined

Viewpoint: Depts Next Year

History

Last Exported: Today at 3:05 PM

Exported By: Tom Smith

Messages: 476 records exported.

c. Select to save or open the file.

49

Backing up and Restoring Cloud EDM

Snapshots are used to back up the Oracle Fusion Cloud Enterprise Data Management environment.

A maintenance snapshot is automatically created every day at a time that you select. This snapshot is named Artifact Snapshot and includes all artifacts in your Cloud EDM environment:

- EDMC Repository and Data Grants
- Users, Groups, and Assigned Roles

Each day, the new maintenance snapshot replaces the previous snapshot. Oracle recommends that you download the snapshot to a local computer to back up the environment.

You can also export artifacts to create snapshots that can be used to rollback the service to a previous state or restore the service in the event of a disaster. You select the artifacts to export. You can export all artifacts in the service or a subset of artifacts. Snapshots that you create by exporting artifacts are kept on the server for 60 days and can also be downloaded to a local computer for archiving purposes.

This scenario provides detailed steps for backing up your Cloud EDM environment by creating your own snapshot of exported artifacts.

For information on backing up with EPM Automate, see Back up Application Snapshot to a Computer.

Administrator Steps for Backing up Cloud EDM

Administrator Prerequisites

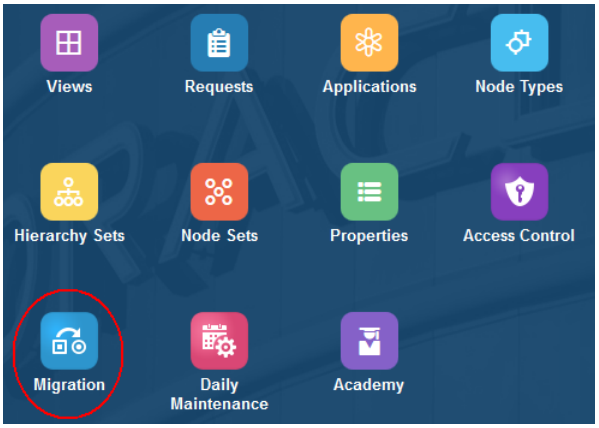
You must be a Service Administrator to access the Migration functionality.

Administrator Process

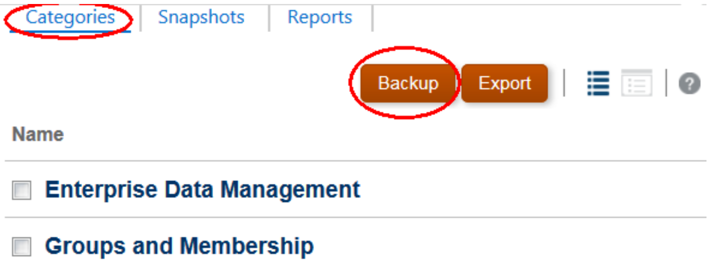
1. Export artifacts to a snapshot.
2. Download the snapshot to a local computer.

Administrator Detailed Steps

1. Click **Migration**.



2. From **Categories**, click **Backup** to create a snapshot that includes all artifacts of the Oracle Fusion Cloud Enterprise Data Management environment.

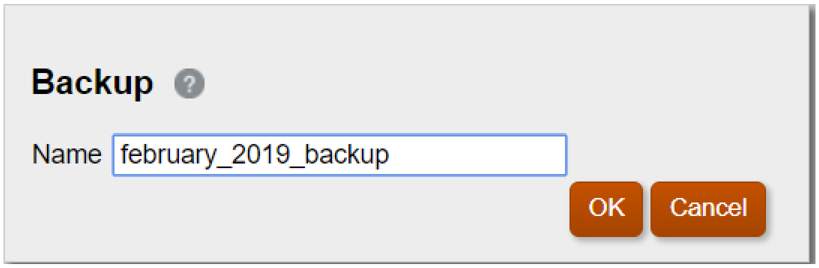


Name

□ Enterprise Data Management

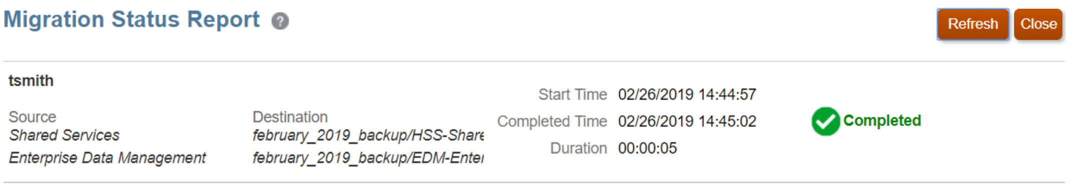
□ Groups and Membership

3. Enter a name for the snapshot and then click **OK**.



Name february_2019_backup

4. Click **Refresh** to update the Migration Status Report until the snapshot is completed.



tsmith

Source
Shared Services
Enterprise Data Management

Destination
february_2019_backup/HSS-Share
february_2019_backup/EDM-Ente

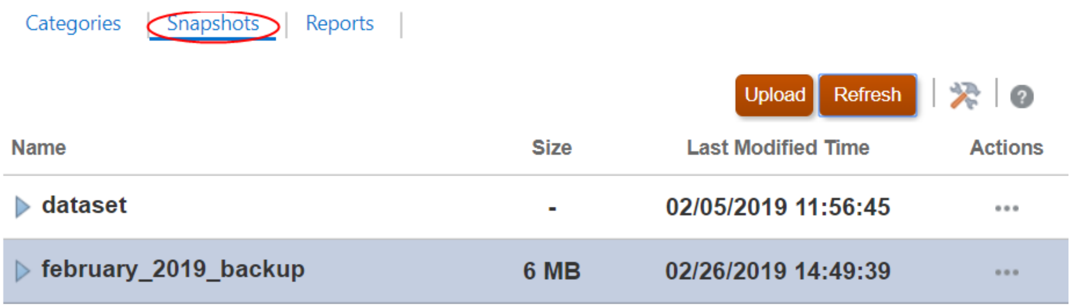
Start Time 02/26/2019 14:44:57
Completed Time 02/26/2019 14:45:02
Duration 00:00:05

✓ Completed

> **Note:**
>
> If the export fails for any reason, the report displays **Failed** as the status. Click **Failed** to open the Migration Details screen, which indicates why the export failed and the corrective action. You can attempt the export operation again after correcting the error that caused the export to fail.

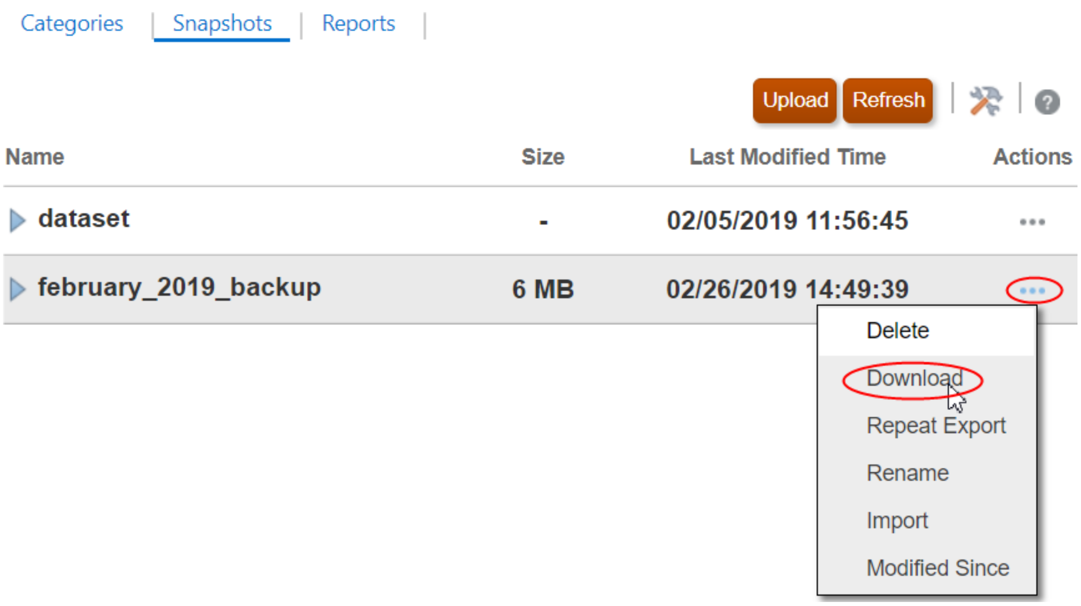
5. Click **Close** to close the Migration Status Report.

   The folder containing the exported artifacts is listed on Snapshots.



Name	Size	Last Modified Time	Actions
dataset	-	02/05/2019 11:56:45	•••
february_2019_backup	6 MB	02/26/2019 14:49:39	•••

6. To download the snapshot files to a local computer, from **Snapshots**, click ••• next to the snapshot and then select **Download**.



Name	Size	Last Modified Time	Actions
dataset	-	02/05/2019 11:56:45	•••
february_2019_backup	6 MB	02/26/2019 14:49:39	•••

7. Save the compressed file to the local computer.

About Restoring the Cloud EDM Environment

You can restore artifacts from previous snapshots. For example, you can restore the artifacts and data to the state they were in a few days ago by restoring them from a snapshot that you backed up to a local machine. In such scenarios, you should upload the snapshot to Oracle Enterprise Data Management Cloud before restoring artifacts.

Snapshots that you create or upload to the service are stored on the server for 60 days, after which they are automatically deleted.

Administrator Steps for Restoring Cloud EDM

Administrator Prerequisites

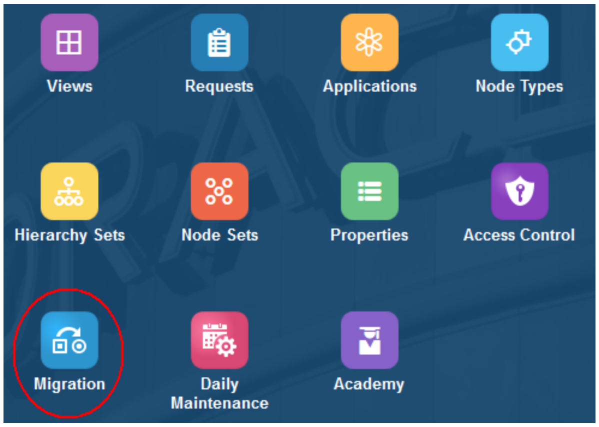
You must be a Service Administrator to access the Migration functionality.

Administrator Process

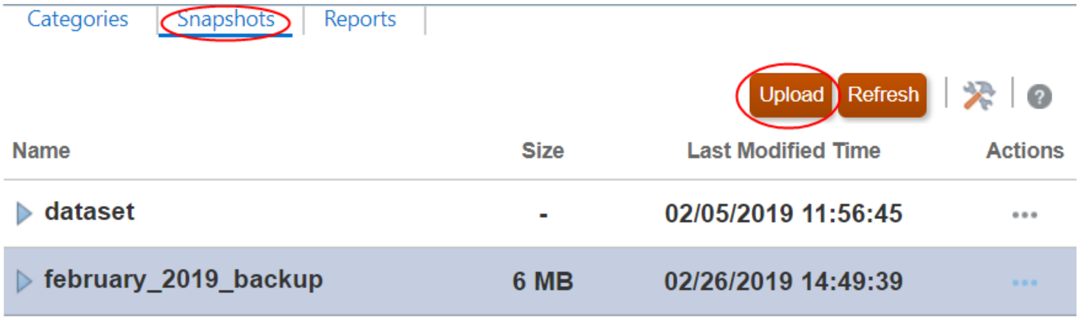
1. Upload the snapshot to restore.
2. Import the snapshot.

Administrator Detailed Steps

1. Click **Migration**.

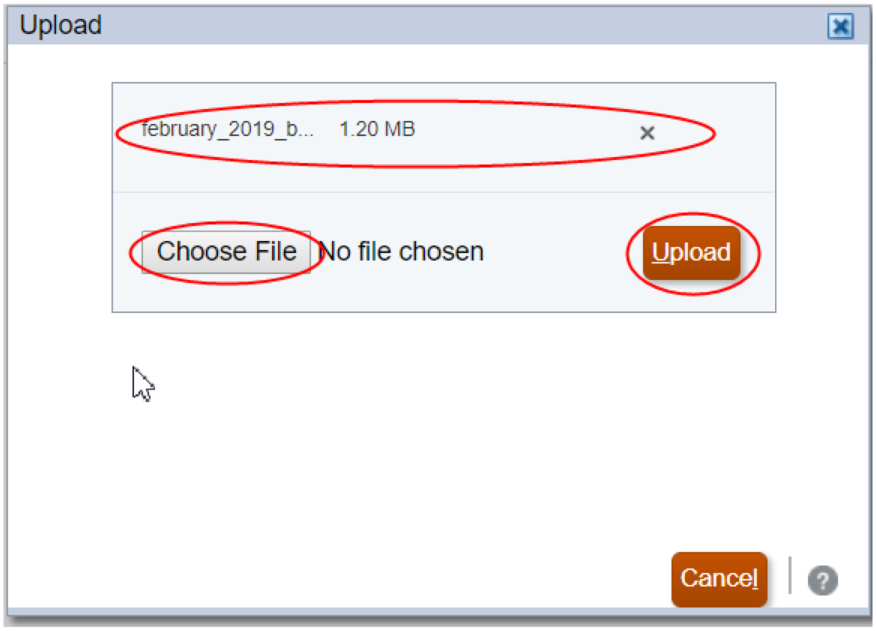


2. From **Snapshots**, click **Upload**.

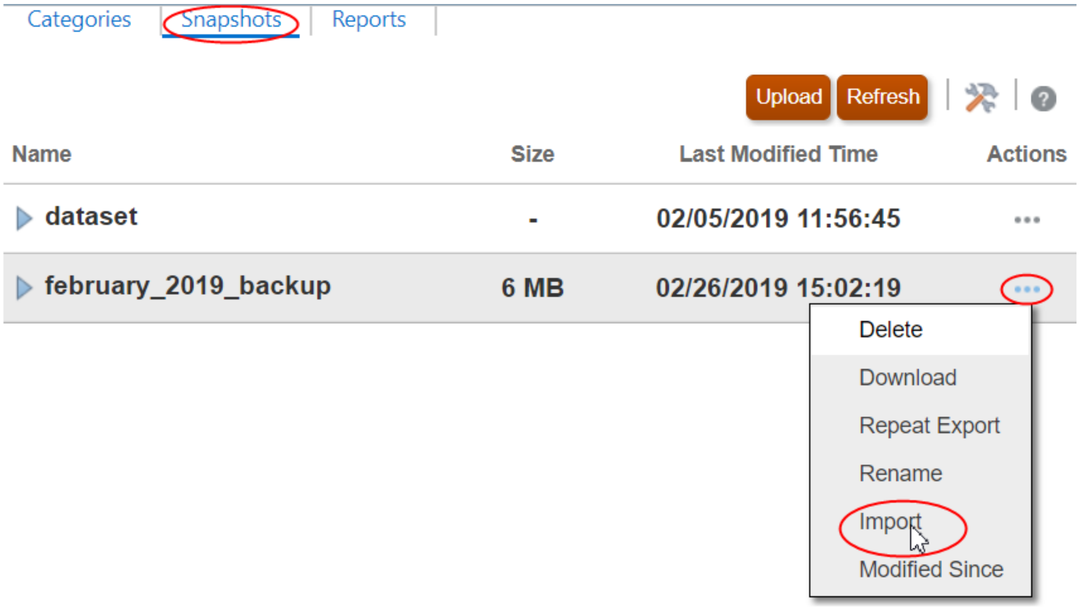


Name	Size	Last Modified Time	Actions
dataset	-	02/05/2019 11:56:45	•••
february_2019_backup	6 MB	02/26/2019 14:49:39	•••

3. Click **Choose File**, locate the snapshot files that you want to upload, and then click **Upload**.

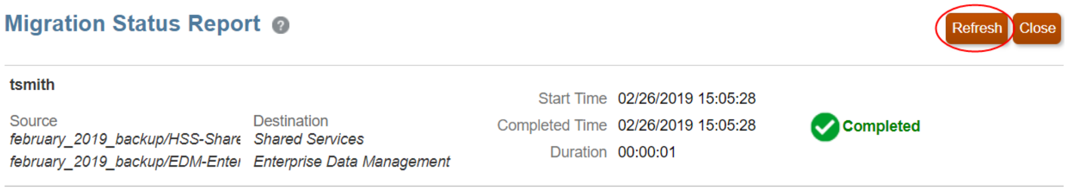


4. From Snapshots, select the snapshot that you uploaded, click ••• , and then select **Import**.



Name	Size	Last Modified Time	Actions
dataset	-	02/05/2019 11:56:45	•••
february_2019_backup	6 MB	02/26/2019 15:02:19	•••

5. Click **OK** to confirm the import.
6. Click **Refresh** to update the Migration Status Report until the import is completed.



Migration Status Report

tsmith

Source	Destination
february_2019_backup/HSS-Share	Shared Services
february_2019_backup/EDM-Ente	Enterprise Data Management

Start Time 02/26/2019 15:05:28
Completed Time 02/26/2019 15:05:28
Duration 00:00:01

✓ Completed

7. Click **Close** to close the Migration Status Report.

50

Managing Governance Workflows and Approvals

Request workflows move requests through Submit and Approve stages. Submitters create requests of proposed changes and submit them for approval. Approval policies determine which users are invited to approve a request as well as the manner in which the request is approved.

This scenario demonstrates how to:

- Assign permissions to users so they can submit requests
- Enable and configure approval policies
- Submit requests
- Approve requests

Administrator Steps

The application owner assigns a user the *Participant(Write)* permission allowing them to submit changes via requests. After assigning permissions, the application owner configures one or more approval policies to enable other users to review and approve the requests being submitted.

Administrator Prerequisites

You must have *Owner* or *Metadata Manager* permission on the application to be able to grant users the *Participant(Write)* permission and to configure approval policies.

Administrator Process

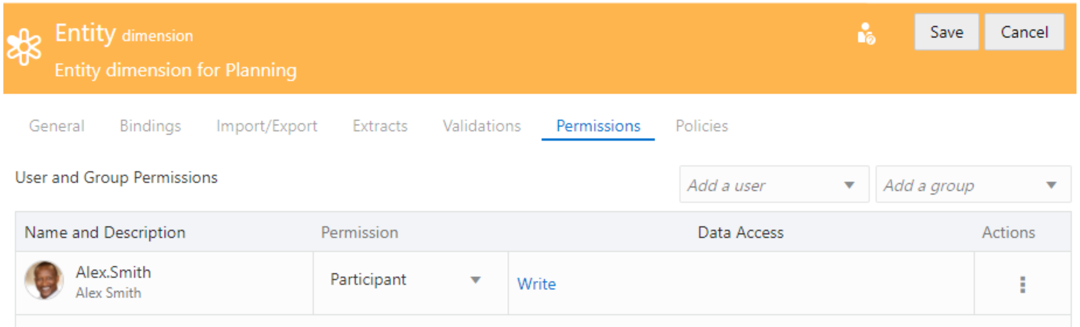
1. Assign user *Participant(Write)* permission for a dimension. See Working with Permissions
2. Enable and configure approval policy for the dimension. See Configuring Policies.

Administrator Detailed Steps

Step 1: Assign User *Participant(Write)* Permission

Assigning a user *Participant(Write)* permission to a specific dimension gives the user permission to data in that dimension. To assign a user *Participant(Write)* permission to a dimension:

1. Open the dimension inspector.
2. On the **Permissions** tab, click **Edit**.
3. In the **Add a user** or **Add a group** drop down list, select the user or group that you want to grant permission to.
   By default, the user is granted *Participant* permission with Read data access.
4. To grant *Participant(Write)*, in the Data Access column, click **Read**, and then select **All** for Allowed Actions and **Edit All** for Properties.



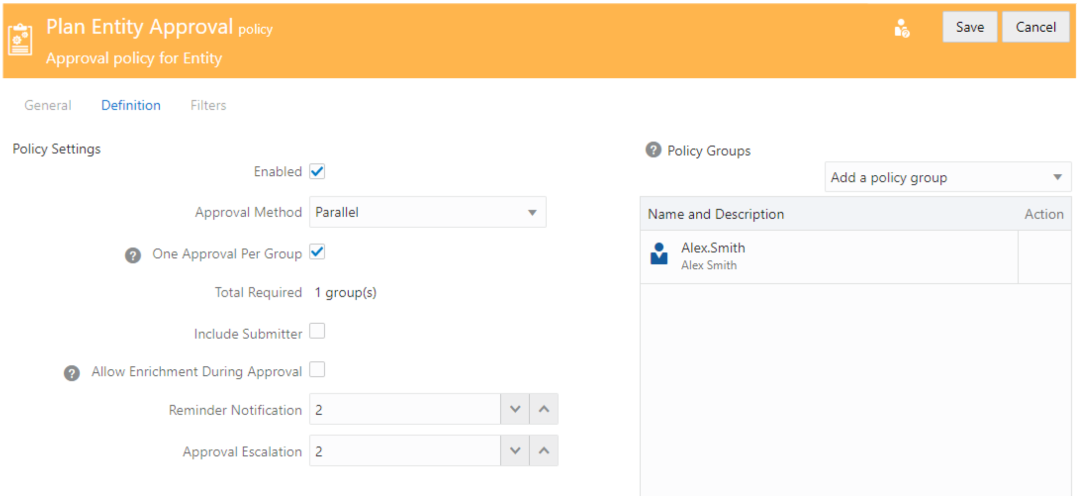
Name and Description	Permission	Data Access	Actions
Alex.Smith Alex Smith	Participant	Write	⋮

5. Click **Save**.

Step 2: Enable and configure approval policy for the dimension

To manage changes to a particular dimension, set up an approval policy for the dimension.

1. Open the dimension inspector.
2. On the **Policies** tab, click the name of the approval policy.
3. **Optional**: On the **General** tab, click **Edit** and change the policy name or description.
4. On the **Definition** tab, click **Edit**, and define the policy settings. For more information, see Creating and Enabling Approval Policies.



Enabled ☑
Approval Method Parallel
One Approval Per Group ☑
Total Required 1 group(s)
Include Submitter □
Allow Enrichment During Approval □
Reminder Notification 2
Approval Escalation 2

Name and Description	Action
Alex.Smith Alex Smith	

5. Click **Save**.

User Steps

The submitter creates a draft request, adds request items to it, and submits the request for approval.

The approver reviews the proposed changes in the request and approves (or pushes back or rejects) the request.

User Prerequisites

Users who draft and work on requests must be set up with the *Participant(Write)* permission for the dimension.

User Process

Submitter:

1. Create a request and add request items.
2. **Optional**: Add comments or attachments to the request.
3. Submit request for approval.

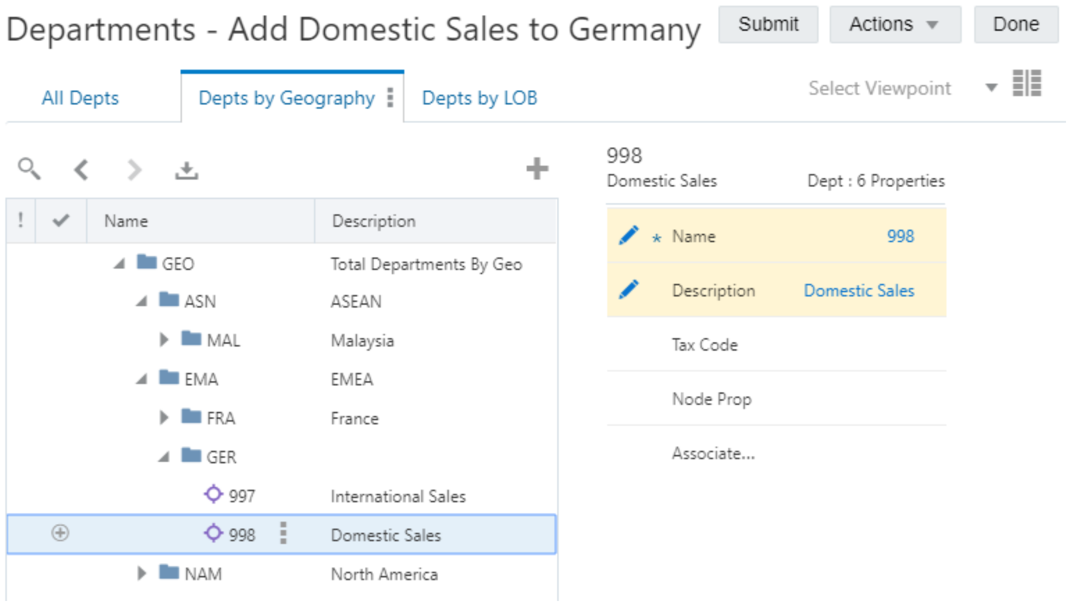
Approver:

1. Open the request from the invitation email or from the request To Do list.
2. Review the changes in the request.
3. **Optional**: Add comments or attachments to the request.
4. Approve, push back, or reject the request.

User Detailed Steps

Submitter:

1. Open a view and create a draft request.
2. Add request items.



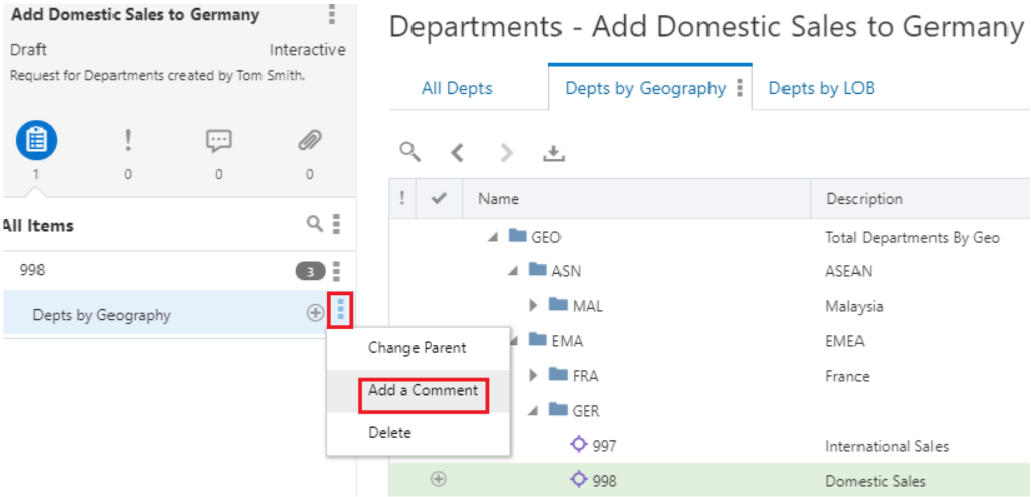
Departments - Add Domestic Sales to Germany

Name	Description
GEO	Total Departments By Geo
ASN	ASEAN
MAL	Malaysia
EMA	EMEA
FRA	France
GER	
997	International Sales
998	Domestic Sales
NAM	North America

998
Domestic Sales
Dept : 6 Properties

* Name	998
Description	Domestic Sales
Tax Code	
Node Prop	
Associate...	

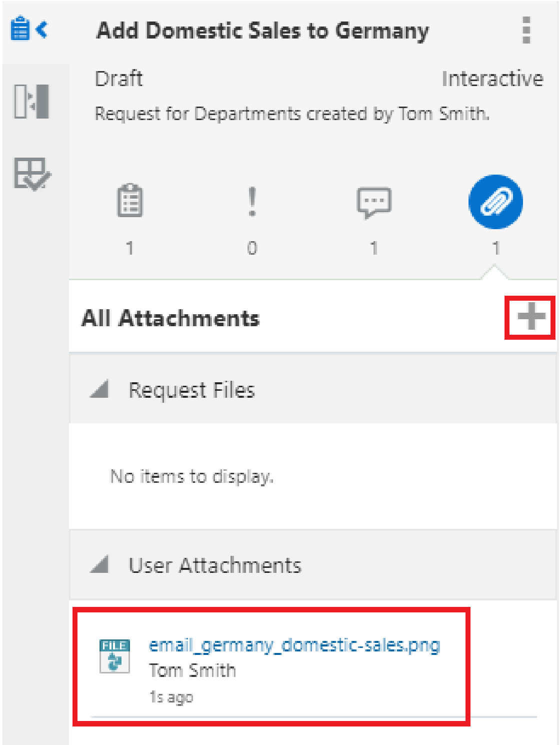
3. Optionally:
   - Add request comments.
   - Add item level comments



Request for Departments created by Tom Smith.

Name	Description
GEO	Total Departments By Geo
ASN	ASEAN
MAL	Malaysia
EMA	EMEA
FRA	France
GER	
997	International Sales
998	Domestic Sales

- Add an attachment.



Request for Departments created by Tom Smith.

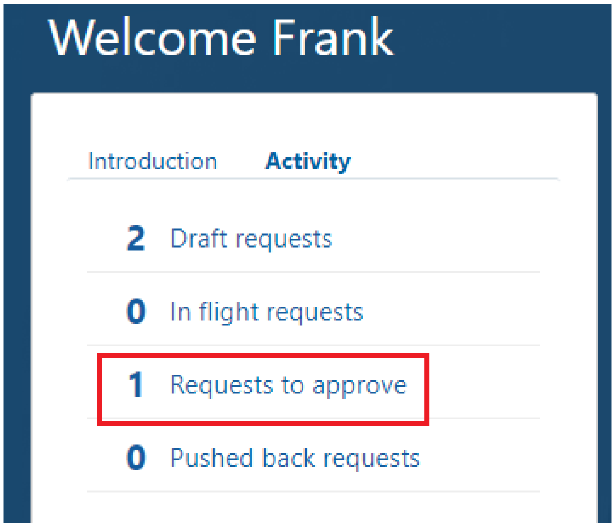
No items to display.

email_germany_domestic-sales.png
Tom Smith
1s ago

4. Click **Submit**. The request status changes from Draft to In Flight. The request stage moves from Submit to Approve.

Approver:

1. A user who is invited to approve a request can click on the link in the email or log in to Oracle Fusion Cloud Enterprise Data Management and, from the To Do list, click on the **Requests to approve** link.



Welcome Frank

Introduction Activity

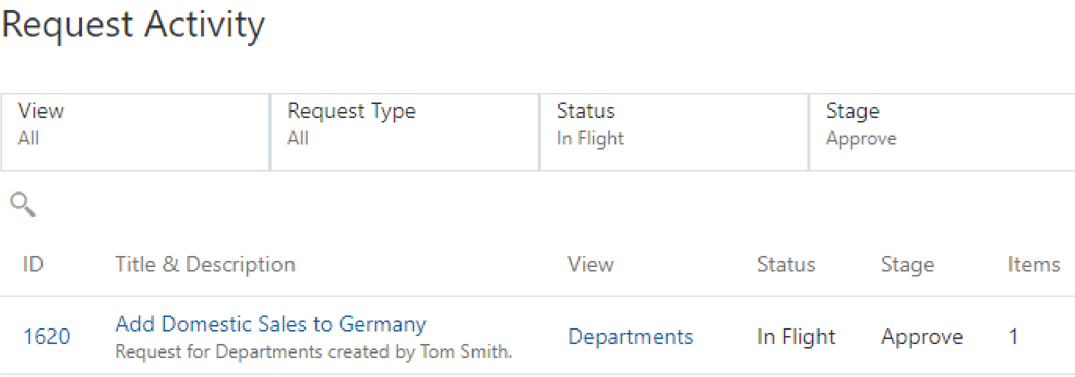
2 Draft requests

0 In flight requests

1 Requests to approve

0 Pushed back requests

2. From the Request Activity list, click the request to approve.

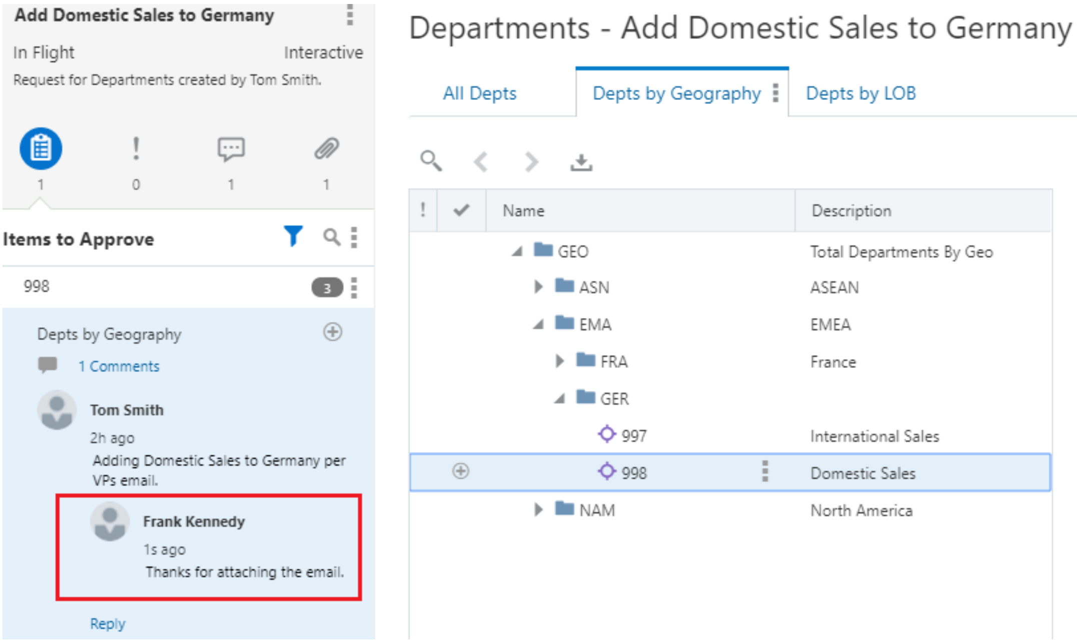


Request Activity

View	Request Type	Status	Stage
All	All	In Flight	Approve

ID	Title & Description	View	Status	Stage	Items
1620	Add Domestic Sales to Germany Request for Departments created by Tom Smith.	Departments	In Flight	Approve	1

3. View or add comments; download any attachments and review or add an attachment.



Add Domestic Sales to Germany

In Flight Interactive

Request for Departments created by Tom Smith.

1 0 1 1

Items to Approve

998 3

Depts by Geography

1 Comments

Tom Smith

2h ago

Adding Domestic Sales to Germany per VPs email.

Frank Kennedy

1s ago

Thanks for attaching the email.

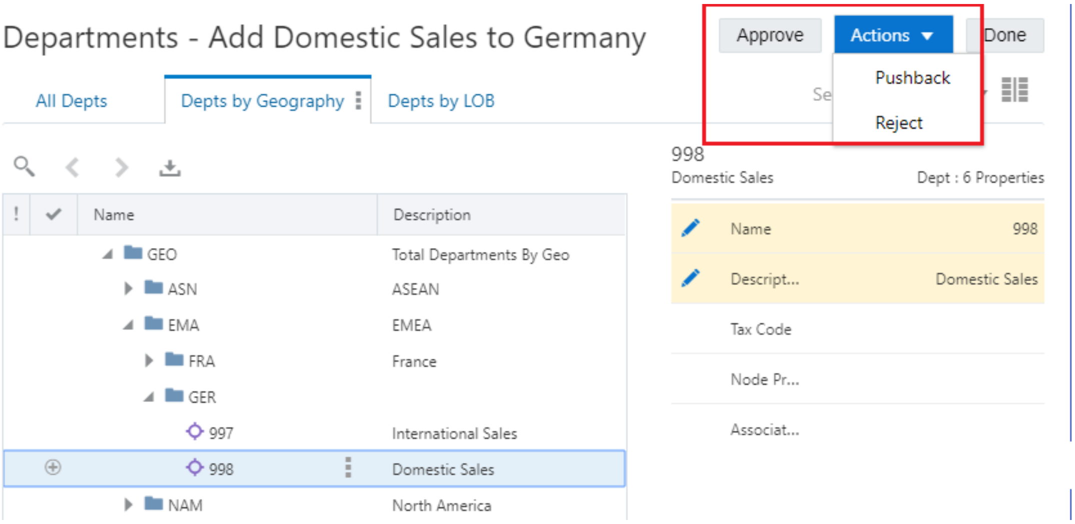
Reply

Departments - Add Domestic Sales to Germany

All Depts | Depts by Geography | Depts by LOB

Name	Description
GEO	Total Departments By Geo
ASN	ASEAN
EMA	EMEA
FRA	France
GER	
997	International Sales
998	Domestic Sales
NAM	North America

4. Click **Approve**. Or, if the request should not be approved, select to Pushback or Reject the request. Pushing back a request returns it to the submitter who can make changes and resubmit it. Reject moves the request to Closed.



Outcome

An administrator can assign permissions to users so they can submit requests and can enable and configure approval policies. Workflow approvals are initiated when a user submits a request for data that is managed by an approval policy. Requests are reviewed and approved by the users specified in the policy.

51

Aligning Applications with Subscriptions and Governance Workflow

This scenario demonstrates how to align across your corporate applications by using subscriptions to synchronize dimension changes and governance workflow to ensure data quality. The steps walk you through creating a maintenance view and setting up subscriptions, and then they follow a typical user workflow where a user submits a request in a source viewpoint, subscription requests are generated for target viewpoints, and those target requests go through a data governance workflow for approval.

Suppose you have applications for Corporate Planning, Corporate GL, and Financial Close, and you want to synchronize changes made to the product dimension in the GL application with the product dimensions in the other two applications. You can create a maintenance view that contains viewpoints for the product dimensions in each of the three applications, and then create subscriptions so that changes that are made to the product dimension in the Corporate GL viewpoint will generate subscription requests to make those same changes to the product dimensions in the Corporate Planning and Financial Close viewpoints.

The source request and the generated subscription requests take part in a data governance workflow that requires review and approval to ensure data quality.

The scenario follows these users:

Table 51-1 Business Scenario Users and Tasks

Name	Role	Tasks
Administrator	Administrator	1. Create a maintenance view that contains viewpoints for the product dimensions for Corporate GL, Corporate Planning, and Financial Close applications 2. Create subscriptions from the Corporate GL Cost Center viewpoint to the Corporate Planning and Financial Close viewpoints.
Betty Anderson	Corporate GL data associate	Submit request to add a new product to the corporate GL.

Table 51-1 (Cont.) Business Scenario Users and Tasks

Name	Role	Tasks
Alex Smith	Corporate GL and Financial Close data manager. Member of the Corp Accounting group	Review and approve the request to add a new product to corporate GL. **Note:** When Alex approves the request, the service generates two subscription requests for the Corporate Planning and Financial Close applications. The subscriptions are set to submit automatically, but the subscription request for Corporate Planning will fail to submit because of a validation error.
Martin Conway	Corporate Planning data manager Member of the Corp Planning group	Review and correct the subscription auto-submit failure and submit for approval.
Kerry Lane	Enterprise data manager	Approve the subscription requests for the Corporate Planning and Financial Close applications.

Administrator Steps

The administrator creates a maintenance view to synchronize changes between the Corporate GL product dimension and the Corporate Planning and Financial Close product dimensions and then sets up subscriptions to enact those changes.

Administrator Prerequisites

You must have the *Views - Create* role to create views. You must also have *Owner* or *Metadata Manager* permission on an application to assign permissions for all data objects in that application to other users and groups.

Administrator Steps

1. Create a maintenance view that contains viewpoints for the product dimensions for Corporate GL, Corporate Planning, and Financial Close applications. See Creating a View.
2. Create subscriptions from the Corporate GL viewpoint to the Corporate Planning and Financial Close viewpoints. See Subscribing to Viewpoints.

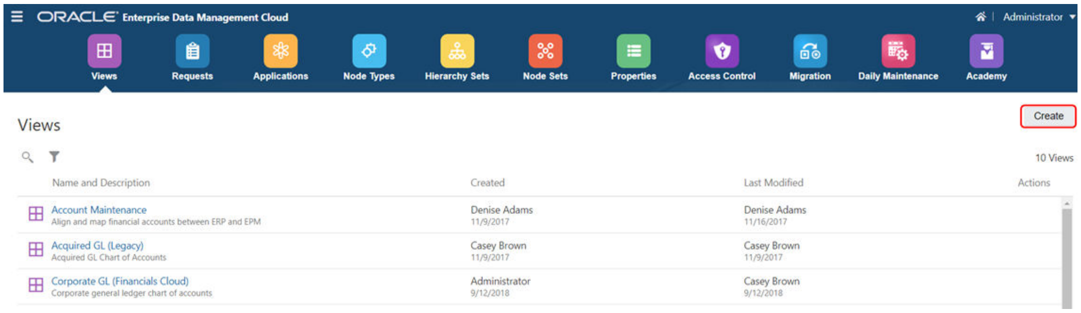
Detailed Steps

Related Topics

- Step 1: Create a Maintenance View
- Step 2: Create Subscriptions

Step 1: Create a Maintenance View

1. From Views, click **Create**.

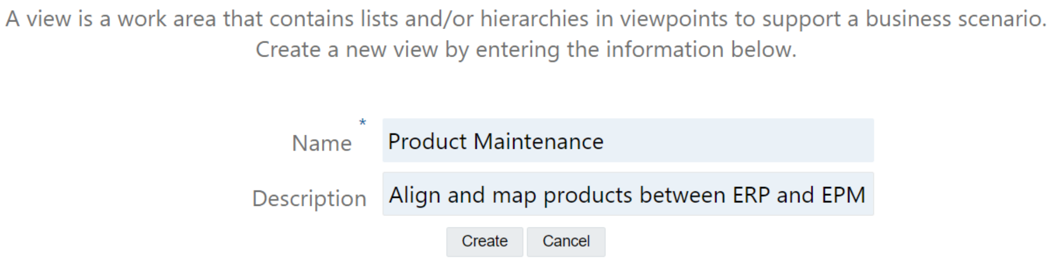


Views

10 Views

Name and Description	Created	Last Modified	Actions
Account Maintenance Align and map financial accounts between ERP and EPM	Denise Adams 11/9/2017	Denise Adams 11/16/2017	
Acquired GL (Legacy) Acquired GL Chart of Accounts	Casey Brown 11/9/2017	Casey Brown 11/9/2017	
Corporate GL (Financials Cloud) Corporate general ledger chart of accounts	Administrator 9/12/2018	Casey Brown 9/12/2018	

2. Enter a name and description for the maintenance view, and then click **Create**.



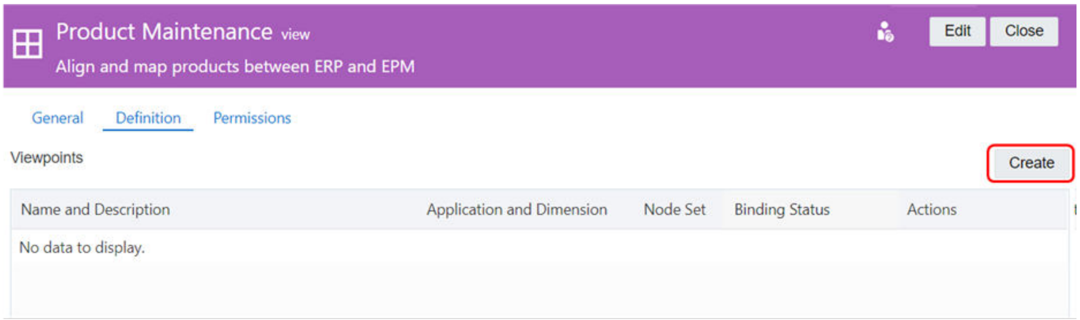
A view is a work area that contains lists and/or hierarchies in viewpoints to support a business scenario. Create a new view by entering the information below.

Name * Product Maintenance

Description Align and map products between ERP and EPM

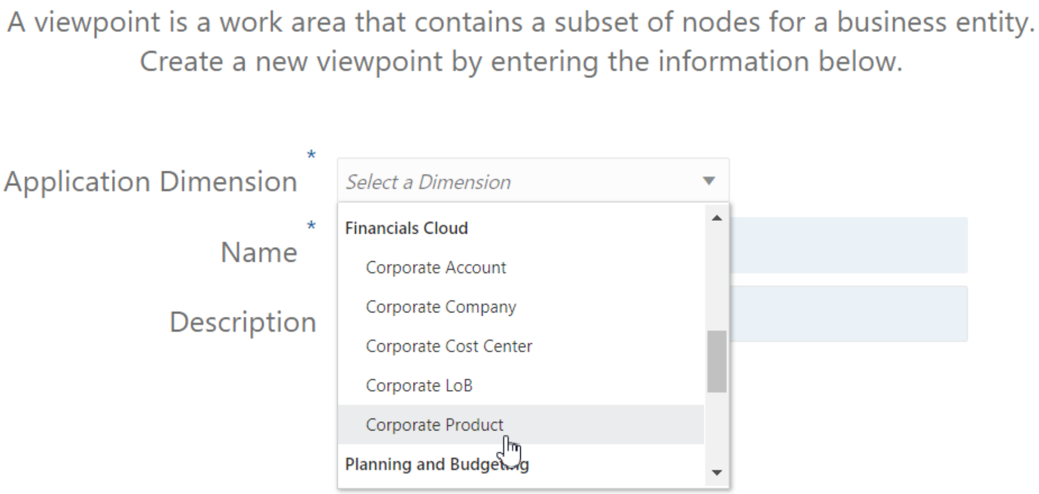
The maintenance view is displayed in the view inspector.

3. On the Definition tab of the view inspector, click **Create** to define the viewpoints in the maintenance view.



Name and Description	Application and Dimension	Node Set	Binding Status	Actions
No data to display.				

4. In **Application Dimension**, use the drop down list to select the dimension for which you want to create the viewpoint. In this scenario, we will choose the **Corporate Product** dimension in the Financials Cloud application.



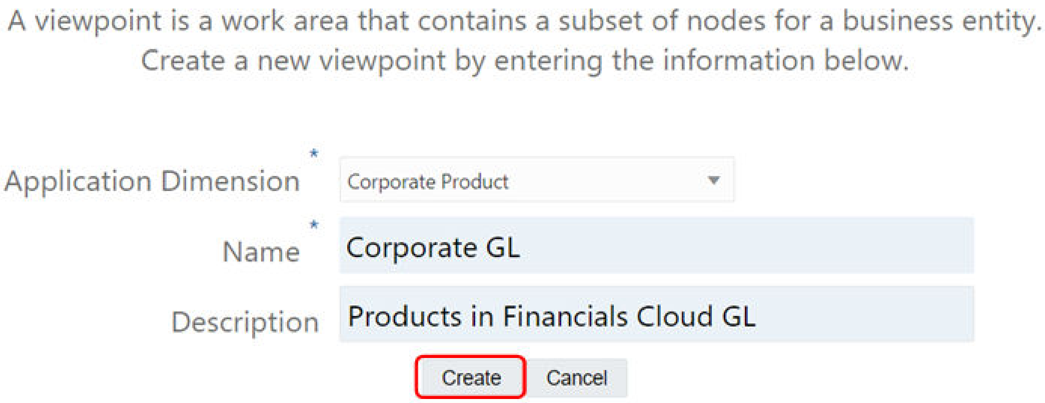
A viewpoint is a work area that contains a subset of nodes for a business entity. Create a new viewpoint by entering the information below.

Application Dimension * Select a Dimension

Name *

Description

5. Enter a name and, optionally, a description for the viewpoint, and then click **Create**.



A viewpoint is a work area that contains a subset of nodes for a business entity. Create a new viewpoint by entering the information below.

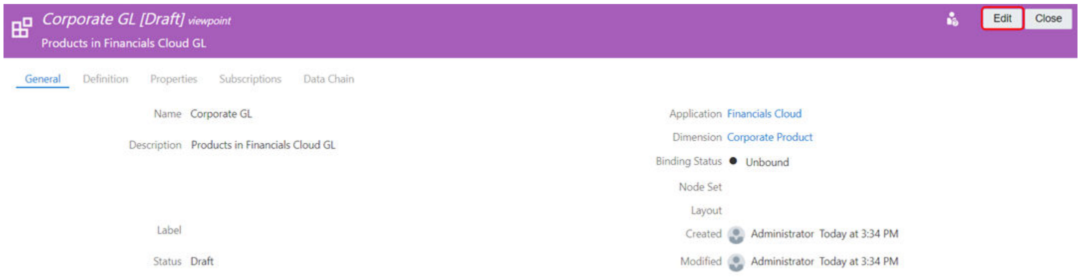
Application Dimension * Corporate Product

Name * Corporate GL

Description Products in Financials Cloud GL

The viewpoint is displayed in draft mode in the viewpoint inspector.

6. From the viewpoint inspector, click **Edit**.



Corporate GL [Draft] viewpoint
Products in Financials Cloud GL

Name Corporate GL

Description Products in Financials Cloud GL

Label

Status Draft

Application Financials Cloud

Dimension Corporate Product

Binding Status ● Unbound

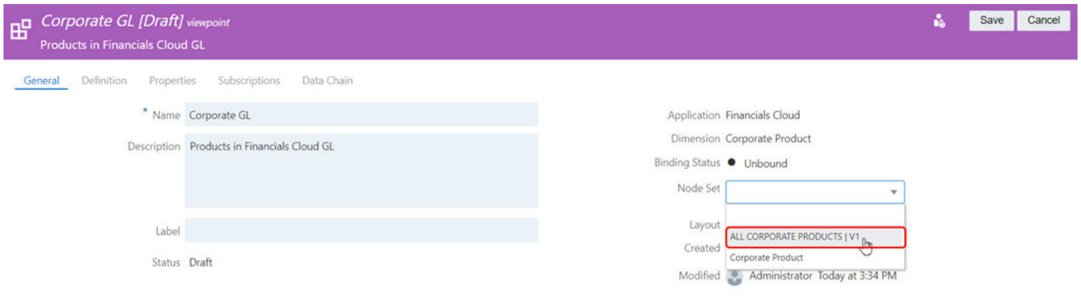
Node Set

Layout

Created Administrator Today at 3:34 PM

Modified Administrator Today at 3:34 PM

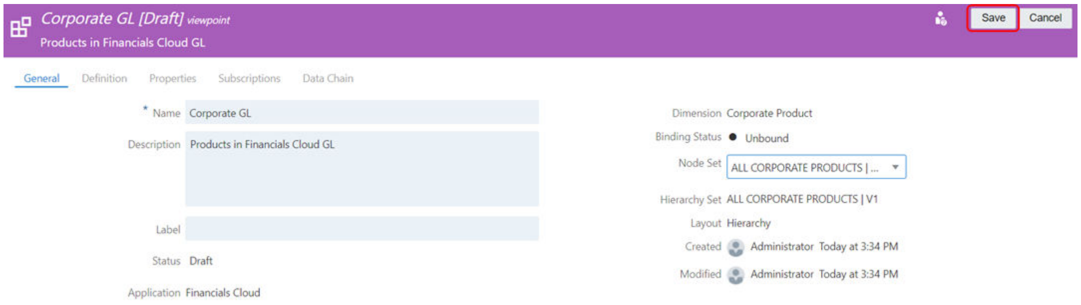
7. In **Node Set**, select the drop down menu and then select the node set for the viewpoint. In this scenario, we will choose **ALL CORPORATE PRODUCTS | V1**.


Corporate GL [Draft] viewpoint
Products in Financials Cloud GL
Save
Cancel
General
Definition
Properties
Subscriptions
Data Chain

* Name: Corporate GL
Description: Products in Financials Cloud GL
Label:
Status: Draft
Application: Financials Cloud
Dimension: Corporate Product
Binding Status: ● Unbound
Node Set:
ALL CORPORATE PRODUCTS | V1
Corporate Product
Layout
Created
Modified: Administrator Today at 3:34 PM

When you select the node set, the hierarchy set is automatically populated.

8. Click **Save**.

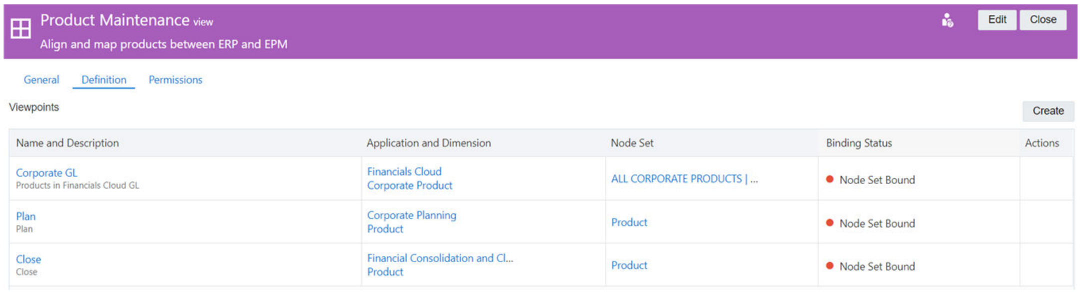

Corporate GL [Draft] viewpoint
Products in Financials Cloud GL
Save
Cancel
General
Definition
Properties
Subscriptions
Data Chain

* Name: Corporate GL
Description: Products in Financials Cloud GL
Label:
Status: Draft
Application: Financials Cloud
Dimension: Corporate Product
Binding Status: ● Unbound
Node Set: ALL CORPORATE PRODUCTS | ...
Hierarchy Set: ALL CORPORATE PRODUCTS | V1
Layout: Hierarchy
Created: Administrator Today at 3:34 PM
Modified: Administrator Today at 3:34 PM

The Binding Status field is updated to display **Node Set Bound**, which indicates that the viewpoint contains a node set that is bound to the Products dimension. See Understanding Binding Rules.

9. From the view inspector, repeat steps **3-8** to create viewpoints for these dimensions:

Viewpoint Name	Viewpoint Description	Application	Dimension	Node Set
Plan	Plan	Corporate Planning	Product	Product
Close	Close	Financial Consolidation and Close	Product	Product



Viewpoints

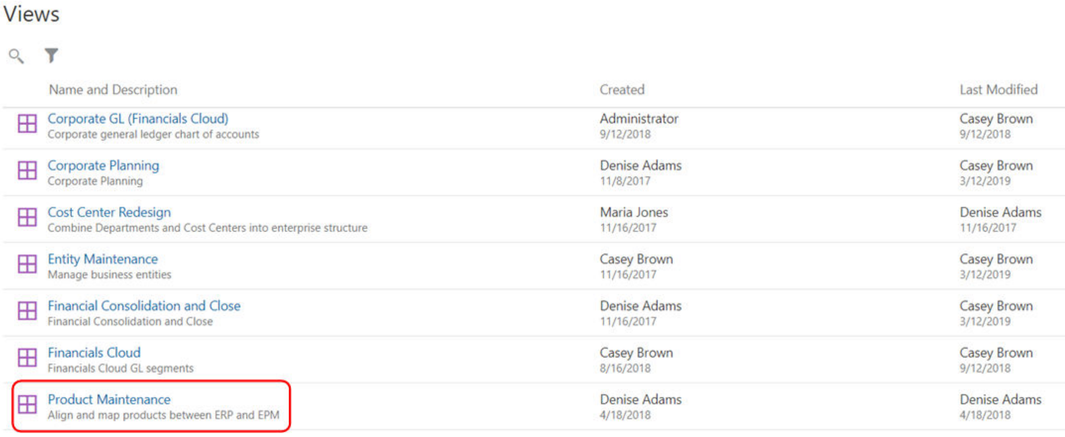
Name and Description	Application and Dimension	Node Set	Binding Status	Actions
Corporate GL Products in Financials Cloud GL	Financials Cloud Corporate Product	ALL CORPORATE PRODUCTS | ...	● Node Set Bound	
Plan Plan	Corporate Planning Product	Product	● Node Set Bound	
Close Close	Financial Consolidation and Cl... Product	Product	● Node Set Bound	

Step 2: Create Subscriptions

After you create the maintenance view with the viewpoints for Corporate GL, Corporate Planning and Financial Close, you create subscriptions for the Corporate Planning and

Financial Close viewpoints so that when changes are made in the Corporate GL viewpoint, subscription requests are automatically generated to make those same changes in the other two viewpoints.

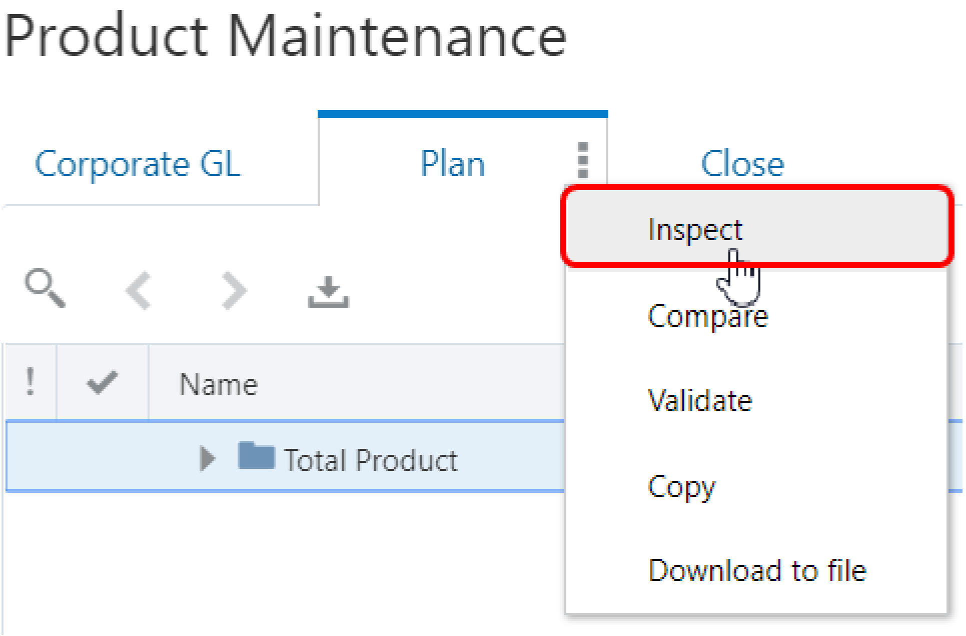
1. From Views, click **Product Maintenance**.



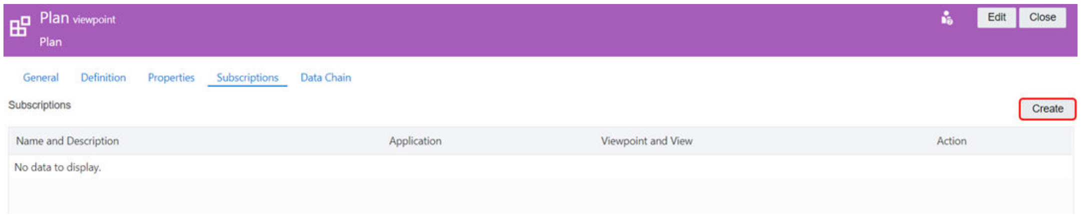
Views

Name and Description	Created	Last Modified
Corporate GL (Financials Cloud) Corporate general ledger chart of accounts	Administrator 9/12/2018	Casey Brown 9/12/2018
Corporate Planning Corporate Planning	Denise Adams 11/8/2017	Casey Brown 3/12/2019
Cost Center Redesign Combine Departments and Cost Centers into enterprise structure	Maria Jones 11/16/2017	Denise Adams 11/16/2017
Entity Maintenance Manage business entities	Casey Brown 11/16/2017	Casey Brown 3/12/2019
Financial Consolidation and Close Financial Consolidation and Close	Denise Adams 11/16/2017	Casey Brown 3/12/2019
Financials Cloud Financials Cloud GL segments	Casey Brown 8/16/2018	Casey Brown 9/12/2018
Product Maintenance Align and map products between ERP and EPM	Denise Adams 4/18/2018	Denise Adams 4/18/2018

2. On the Plan tab, click ⋮ , and then select **Inspect**.



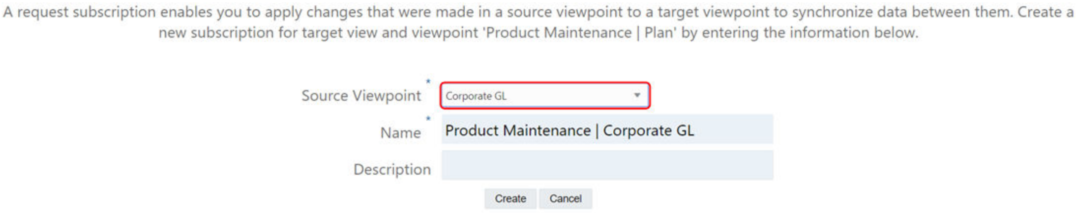
3. On the Subscriptions tab, click **Create**.



Subscriptions

Name and Description	Application	Viewpoint and View	Action
No data to display.			

4. In the Subscription wizard, click the **Source Viewpoint** drop down menu and select the **Corporate GL** viewpoint in the Product Maintenance view. After you select the source viewpoint, the subscription name is automatically populated with the source view and viewpoint. Optionally, enter a description and then click **Create**.



A request subscription enables you to apply changes that were made in a source viewpoint to a target viewpoint to synchronize data between them. Create a new subscription for target view and viewpoint 'Product Maintenance | Plan' by entering the information below.

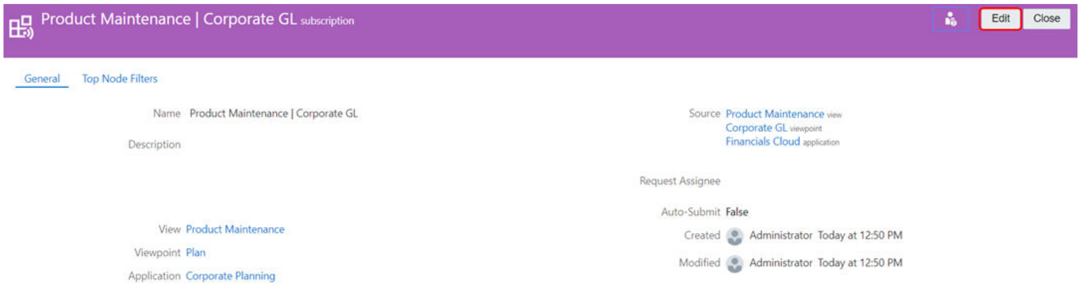
Source Viewpoint * Corporate GL

Name * Product Maintenance | Corporate GL

Description

The subscription is displayed in the subscription inspector.

5. Next, we specify an assignee for the request. Click **Edit**.



Product Maintenance | Corporate GL subscription

Name Product Maintenance | Corporate GL

Description

View Product Maintenance

Viewpoint Plan

Application Corporate Planning

Source Product Maintenance view
Corporate GL viewpoint
Financials Cloud application

Request Assignee

Auto-Submit False

Created Administrator Today at 12:50 PM

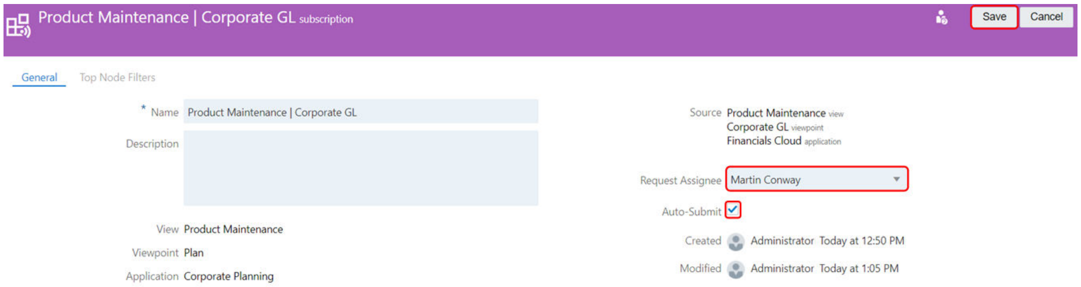
Modified Administrator Today at 12:50 PM

6. In the subscription inspector, perform these actions:

   a. Click the **Request Assignee** drop down menu and select the person to whom the subscription requests should be assigned. In this scenario, we selected Martin Conway, the Corporate Planning data manager.

   b. Select the **Auto-Submit** check box.

   c. Click **Save**, and then click **Close**.



Product Maintenance | Corporate GL subscription

* Name Product Maintenance | Corporate GL

Description

View Product Maintenance

Viewpoint Plan

Application Corporate Planning

Source Product Maintenance view
Corporate GL viewpoint
Financials Cloud application

Request Assignee Martin Conway

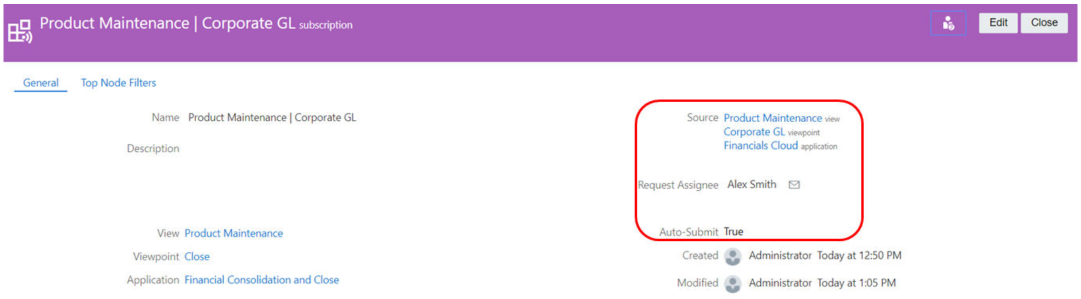
Auto-Submit ☑

Created Administrator Today at 12:50 PM

Modified Administrator Today at 1:05 PM

7. On the Close tab, repeat steps **2-6** to create another subscription with these settings:

   - **Source view**: Product Maintenance
   - **Source viewpoint**: Corporate GL
   - **Request Assignee**: Alex Smith (Financial Close data manager)
   - **Auto-Submit**: True



Product Maintenance | Corporate GL subscription

Name Product Maintenance | Corporate GL

Description

View Product Maintenance

Viewpoint Close

Application Financial Consolidation and Close

Source Product Maintenance view
Corporate GL viewpoint
Financials Cloud application

Request Assignee Alex Smith

Auto-Submit True

Created Administrator Today at 12:50 PM

Modified Administrator Today at 1:05 PM

User Steps

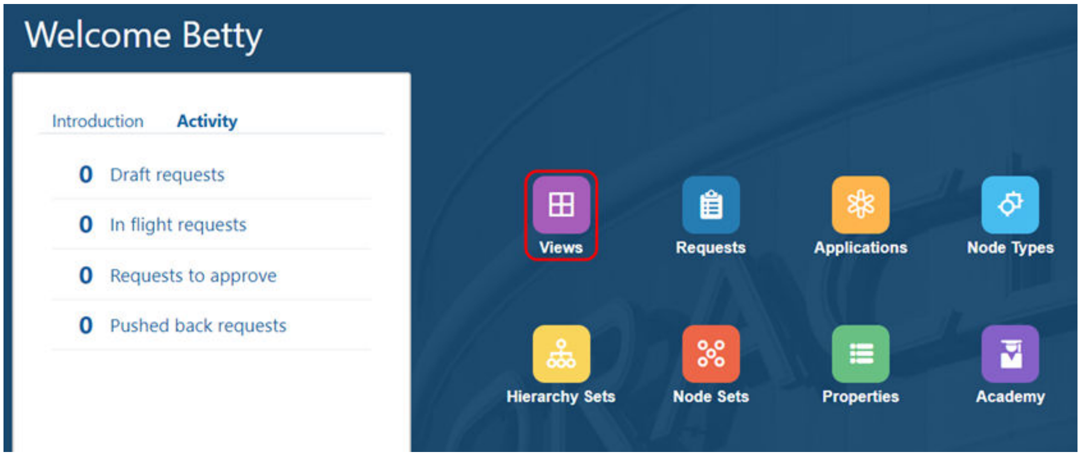
After the administrator sets up the maintenance view and subscriptions to align the product dimensions of the Corporate GL, Corporate Planning, and Financial Close applications, the scenario walks through a user flow to demonstrate the synchronization of enterprise data across applications.

The user flow follows this sequence:

1. Betty Anderson (Corporate GL data associate) enters a request to add a product to the corporate GL.
2. Alex Smith (data manager for both Corporate GL and Financial Close) approves the request.
   The system generates subscription requests to add the same product to the Corporate Planning and Financial Close applications. Because the subscription requests are set to auto-submit, the request for Financial Close is submitted. However, the request for Corporate Planning fails validation and must be reviewed before it can be submitted.
3. Martin Anderson (data manager for Corporate Planning) reviews the subscription request for Planning, fixes the validation error, and submits it.
4. Kerry Lane (Enterprise data manager) approves both the Corporate Planning and Financial Close requests.

Corporate GL Data Associate Steps

1. Log into the service as Betty Anderson, and click **Views**.



Welcome Betty

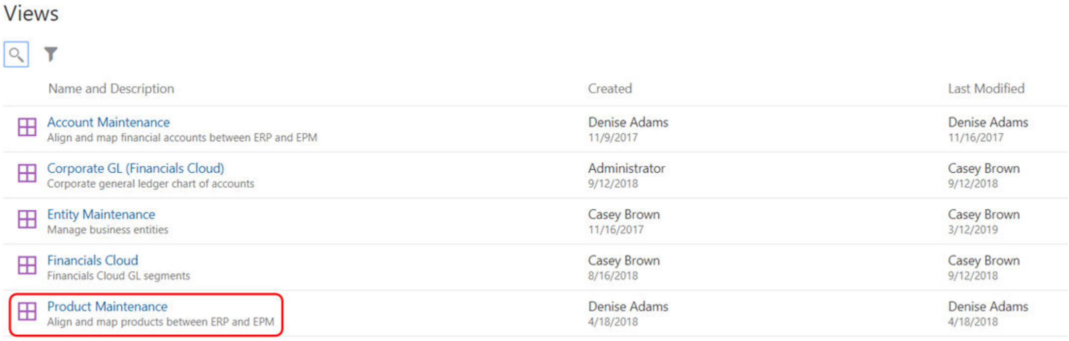
0 Draft requests

0 In flight requests

0 Requests to approve

0 Pushed back requests

2. Click the **Product Maintenance** view.

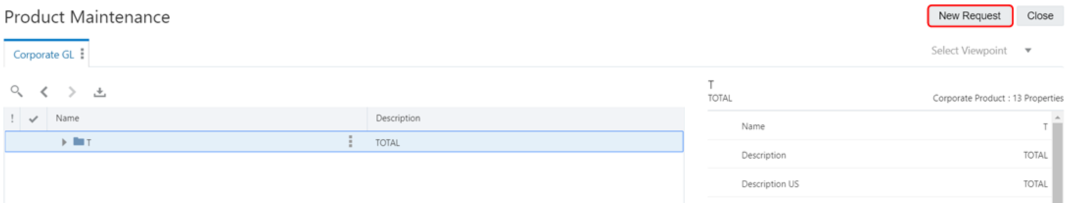


Views

Name and Description	Created	Last Modified
Account Maintenance Align and map financial accounts between ERP and EPM	Denise Adams 11/9/2017	Denise Adams 11/16/2017
Corporate GL (Financials Cloud) Corporate general ledger chart of accounts	Administrator 9/12/2018	Casey Brown 9/12/2018
Entity Maintenance Manage business entities	Casey Brown 11/16/2017	Casey Brown 3/12/2019
Financials Cloud Financials Cloud GL segments	Casey Brown 8/16/2018	Casey Brown 9/12/2018
Product Maintenance Align and map products between ERP and EPM	Denise Adams 4/18/2018	Denise Adams 4/18/2018

The Product Maintenance view is displayed. Because Betty does not have permissions to the Corporate Planning or Financial Close applications, she is only able to see the Corporate GL viewpoint.

3. Click **New Request**.



Product Maintenance

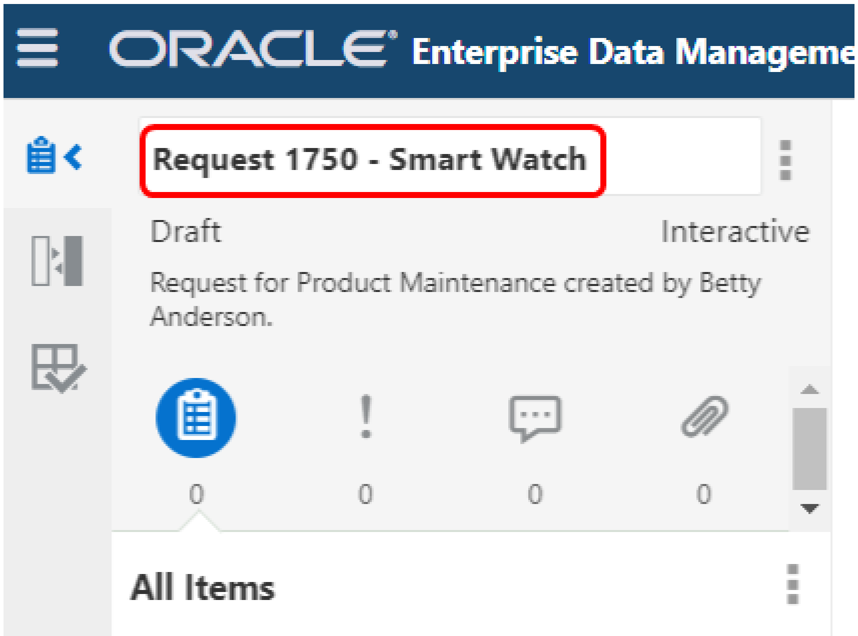
!	✓	Name	Description
		T	TOTAL

T
TOTAL

Corporate Product : 13 Properties

Name	T
Description	TOTAL
Description US	TOTAL

4. In the left panel, click the request name and append – `Smart Watch`.



Request 1750 - Smart Watch

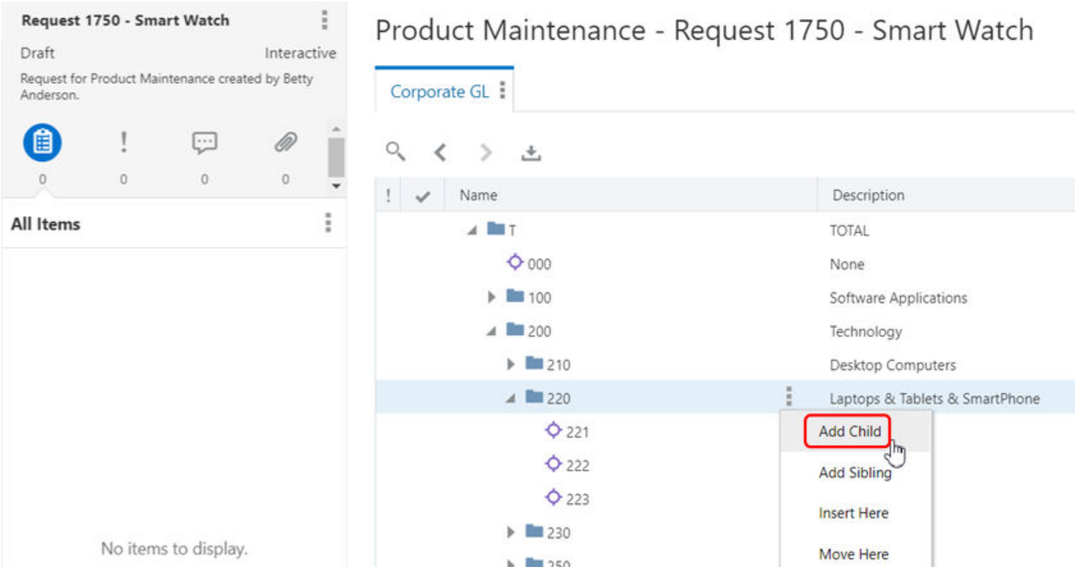
Draft

Interactive

Request for Product Maintenance created by Betty Anderson.

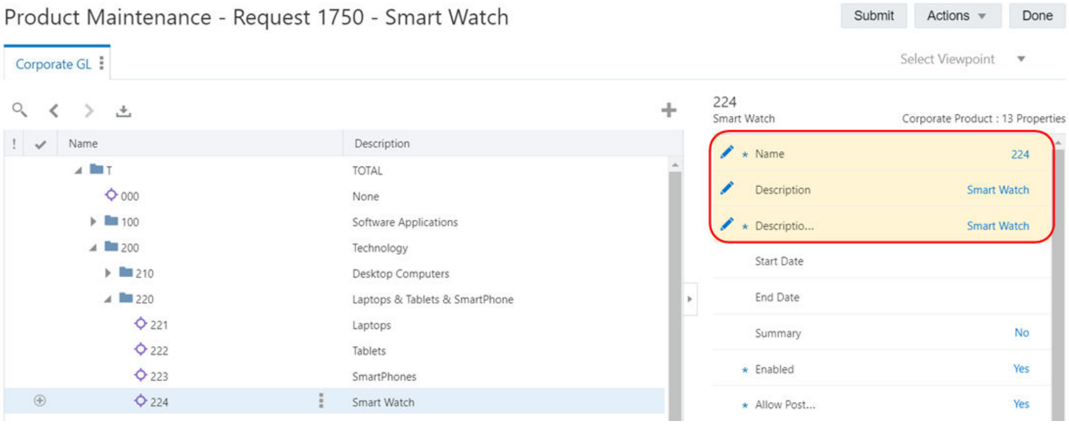
All Items

5. Expand the product hierarchy and select node 220, then click ⋮ and select **Add Child**.


Request 1750 - Smart Watch
Draft
Interactive
Request for Product Maintenance created by Betty Anderson.
0
0
0
0
All Items
No items to display.
Product Maintenance - Request 1750 - Smart Watch
Corporate GL
Add Child
Add Sibling
Insert Here
Move Here

Name	Description
T	TOTAL
000	None
100	Software Applications
200	Technology
210	Desktop Computers
220	Laptops & Tablets & SmartPhone
221	
222	
223	
230	
250	

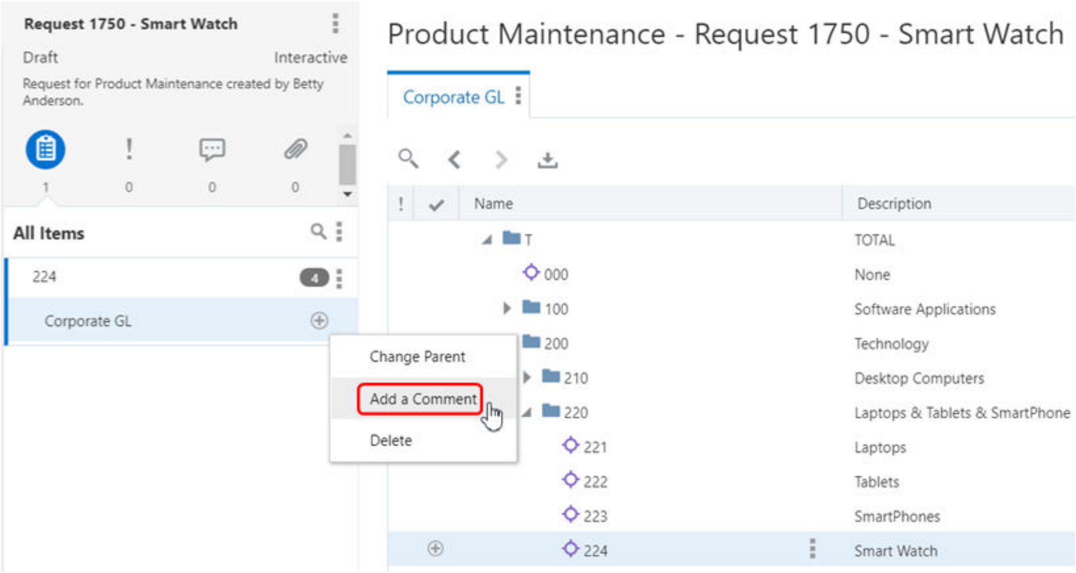
6. In the properties panel, set these properties:
   a. **Name**: 224
   b. **Description**: Smart Watch
   c. **Description US**: Smart Watch


Product Maintenance - Request 1750 - Smart Watch
Submit
Actions
Done
Corporate GL
Select Viewpoint
224
Smart Watch
Corporate Product : 13 Properties

Name	Description
T	TOTAL
000	None
100	Software Applications
200	Technology
210	Desktop Computers
220	Laptops & Tablets & SmartPhone
221	Laptops
222	Tablets
223	SmartPhones
224	Smart Watch

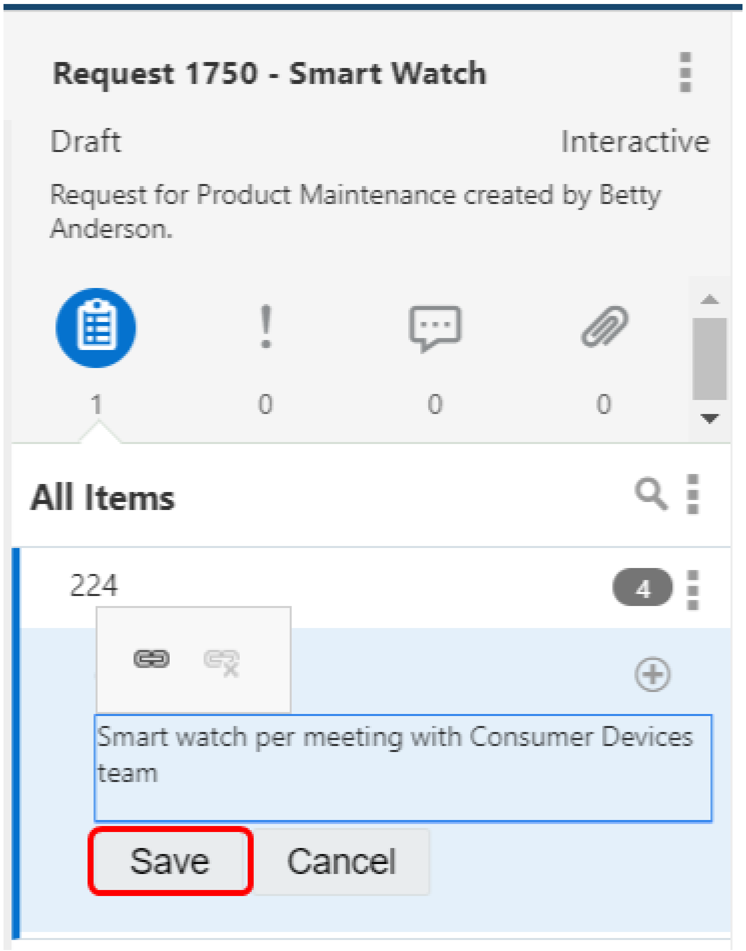
* Name	224
Description	Smart Watch
* Descriptio...	Smart Watch
Start Date	
End Date	
Summary	No
* Enabled	Yes
* Allow Post...	Yes

7. In the request panel, click ⋮ next to Corporate GL, and select **Add a Comment**.


Request 1750 - Smart Watch
Draft
Interactive
Request for Product Maintenance created by Betty Anderson.
1
0
0
0
All Items
224
4
Corporate GL
Change Parent
Add a Comment
Delete
Product Maintenance - Request 1750 - Smart Watch
Corporate GL

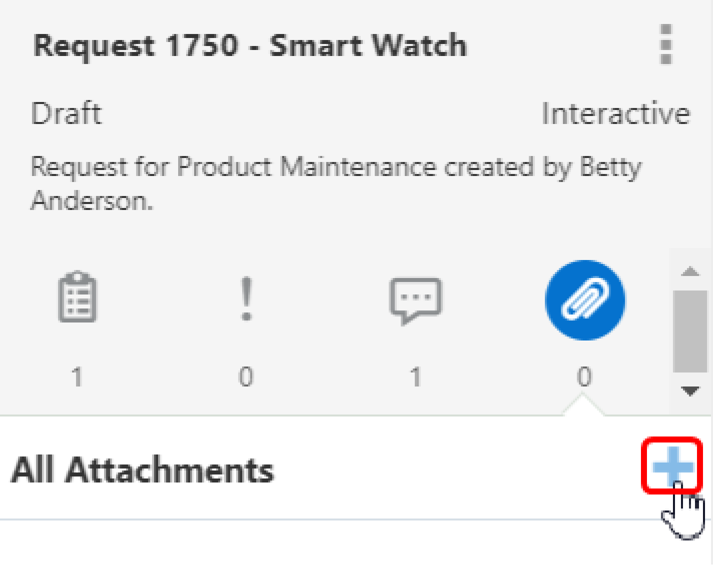
Name	Description
T	TOTAL
000	None
100	Software Applications
200	Technology
210	Desktop Computers
220	Laptops & Tablets & SmartPhone
221	Laptops
222	Tablets
223	SmartPhones
224	Smart Watch

8. Enter a comment, and click **Save**. In this scenario, we entered `Smart watch per meeting with Consumer Devices team`.


Request 1750 - Smart Watch
Draft
Interactive
Request for Product Maintenance created by Betty Anderson.
1
0
0
0
All Items
224
4
Save
Cancel

Smart watch per meeting with Consumer Devices team

9. Click the attachments icon, and then click +.



Request 1750 - Smart Watch

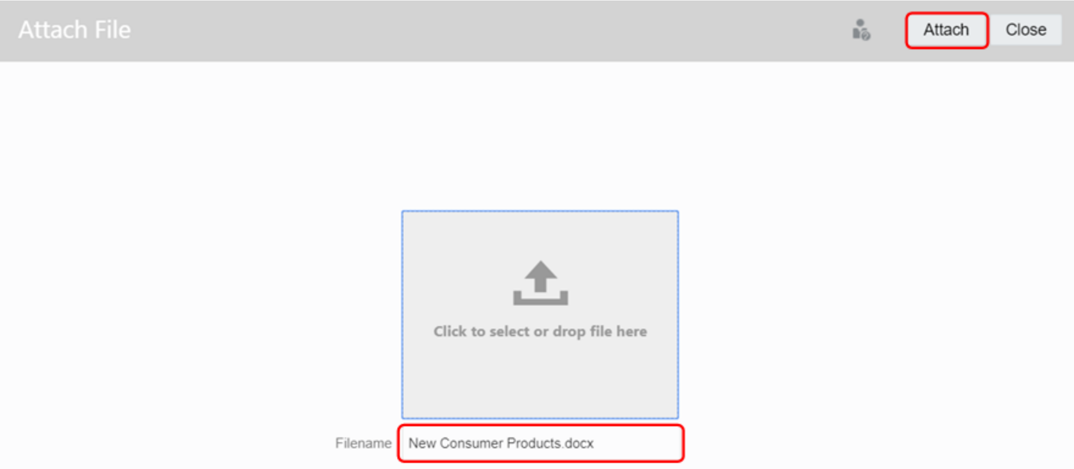
Draft

Interactive

Request for Product Maintenance created by Betty Anderson.

All Attachments

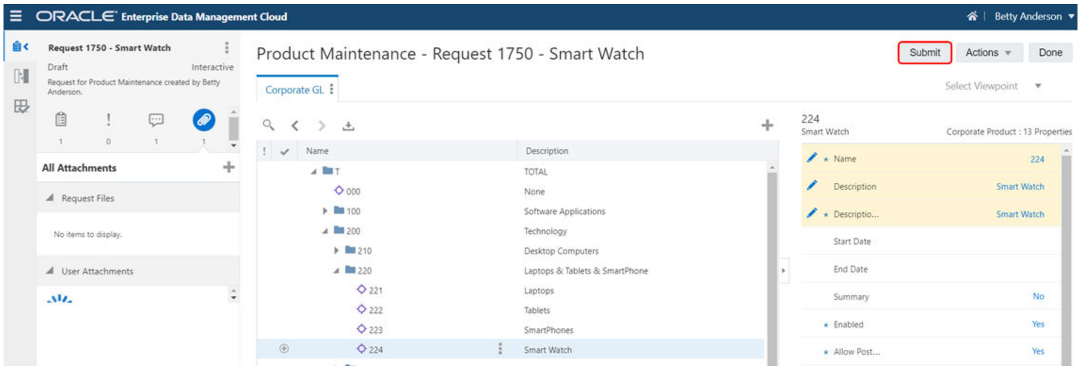
10. Select a file to attach, and then click **Attach**. In this scenario, we attached `New Consumer Products.docx`.



Attach File

Filename: New Consumer Products.docx

11. Click **Submit**.



Product Maintenance - Request 1750 - Smart Watch

Request 1750 - Smart Watch

Draft

Interactive

Request for Product Maintenance created by Betty Anderson.

All Attachments

Request Files

No items to display.

User Attachments

Name	Description
T	TOTAL
000	None
100	Software Applications
200	Technology
210	Desktop Computers
220	Laptops & Tablets & SmartPhone
221	Laptops
222	Tablets
223	SmartPhones
224	Smart Watch

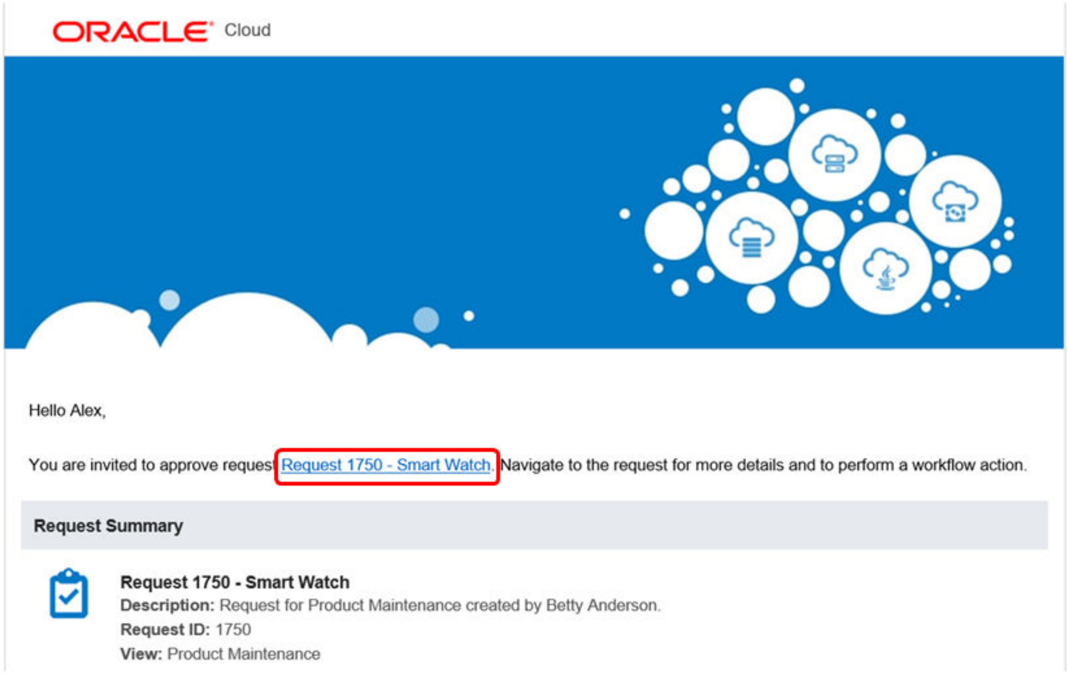
224
Smart Watch
Corporate Product : 13 Properties

* Name	224
Description	Smart Watch
* Descriptio...	Smart Watch
Start Date	
End Date	
Summary	No
* Enabled	Yes
* Allow Post...	Yes

When the request is submitted, a workflow policy alerts Alex, the Corporate GL data manager, of the change request waiting for approval.

Corporate GL Data Manager Steps

1. In the email notification of the request approval, click the link to navigate to the request.



ORACLE Cloud

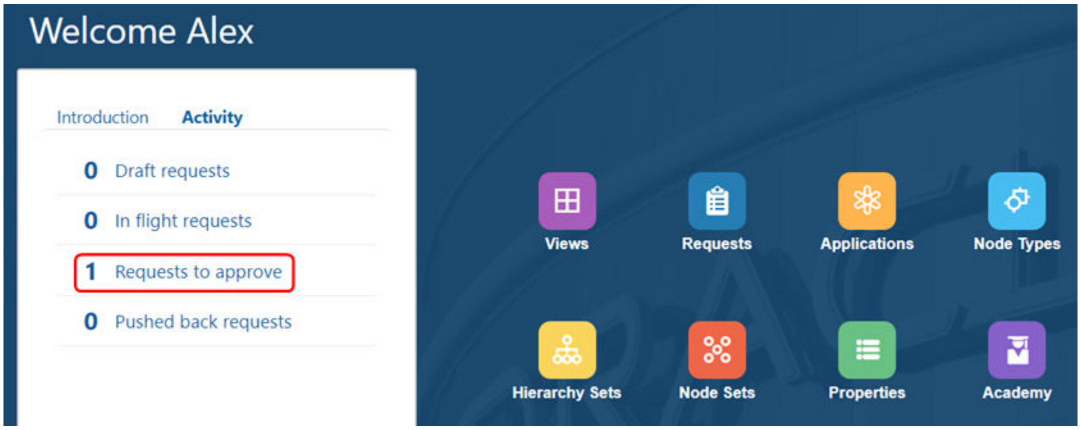
Hello Alex,

You are invited to approve request Request 1750 - Smart Watch. Navigate to the request for more details and to perform a workflow action.

Request Summary

Request 1750 - Smart Watch
Description: Request for Product Maintenance created by Betty Anderson.
Request ID: 1750
View: Product Maintenance

2. In the Activity List, click **Requests to approve**.

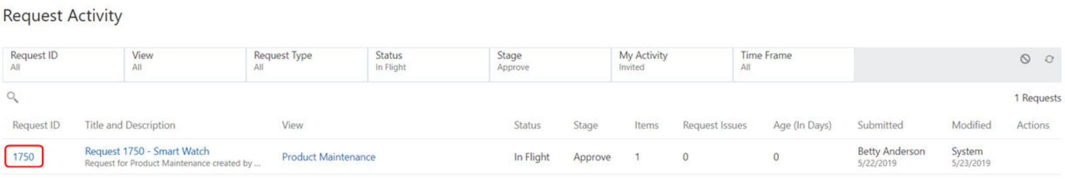


Welcome Alex

Introduction Activity

0 Draft requests
0 In flight requests
1 Requests to approve
0 Pushed back requests

3. In Request Activity, click the **Request ID** for the Smart Watch request.



Request Activity

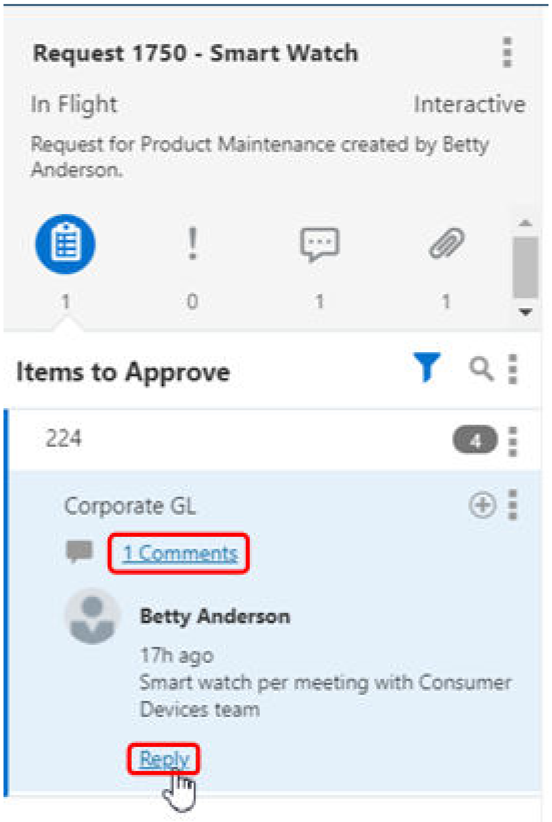
Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	In Flight	Approve	Invited	All

1 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1750	Request 1750 - Smart Watch Request for Product Maintenance created by ...	Product Maintenance	In Flight	Approve	1	0	0	Betty Anderson 5/22/2019	System 5/23/2019	

The request is displayed in the maintenance view. Because Alex is also the Financial Close data manager, his data permissions enable him to see the Close viewpoint as well as the GL viewpoint.

4. In the request panel, click **Comments** to view Betty's comment, and then click **Reply**.



Request 1750 - Smart Watch

In Flight	Interactive

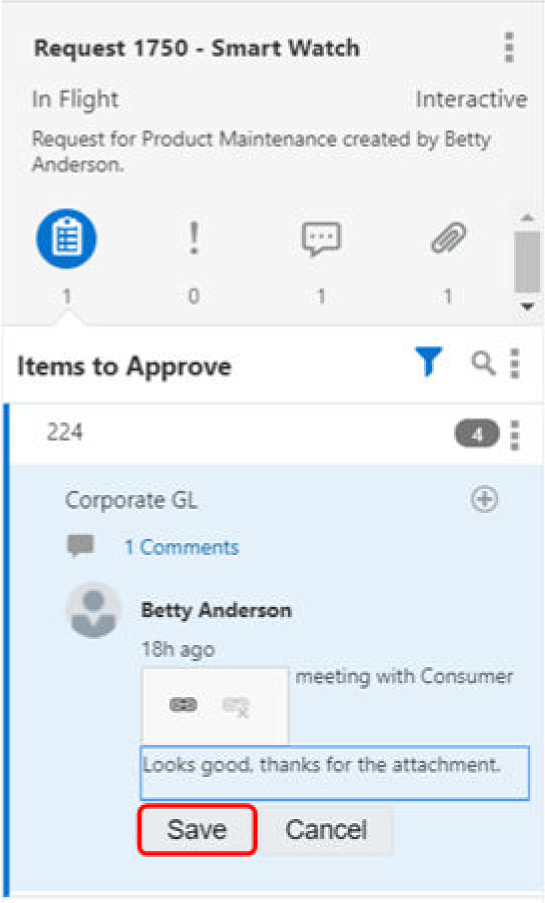
Request for Product Maintenance created by Betty Anderson.

Betty Anderson

17h ago

Smart watch per meeting with Consumer Devices team

5. Enter a reply and then click **Save**.



Request 1750 - Smart Watch

In Flight Interactive

Request for Product Maintenance created by Betty Anderson.

1 0 1 1

Items to Approve

224 4

Corporate GL

1 Comments

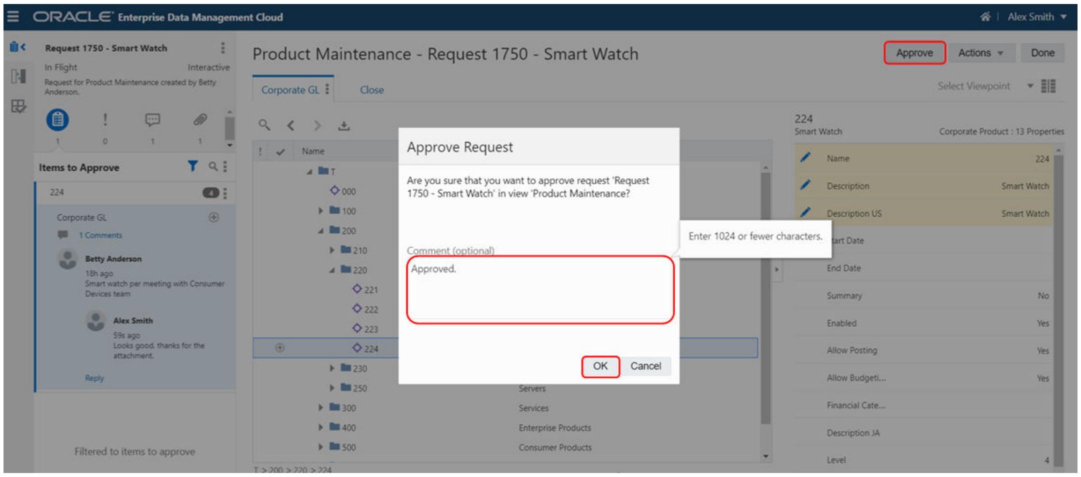
Betty Anderson

18h ago

meeting with Consumer

Looks good, thanks for the attachment.

6. Perform these actions:
    a. Click **Approve**.
    b. (Optional): Enter a comment.
    c. Click **OK**.



Request 1750 - Smart Watch

In Flight Interactive

Request for Product Maintenance created by Betty Anderson.

Items to Approve

224

Corporate GL

1 Comments

Betty Anderson

18h ago

Smart watch per meeting with Consumer Devices team

Alex Smith

59s ago

Looks good, thanks for the attachment.

Reply

Filtered to items to approve

Product Maintenance - Request 1750 - Smart Watch

Approve Request

Are you sure that you want to approve request 'Request 1750 - Smart Watch' in view 'Product Maintenance?

Enter 1024 or fewer characters.

Comment (optional)

Approved.

T
000
100
200
210
220
221
222
223
224
230
250 Servers
300 Services
400 Enterprise Products
500 Consumer Products

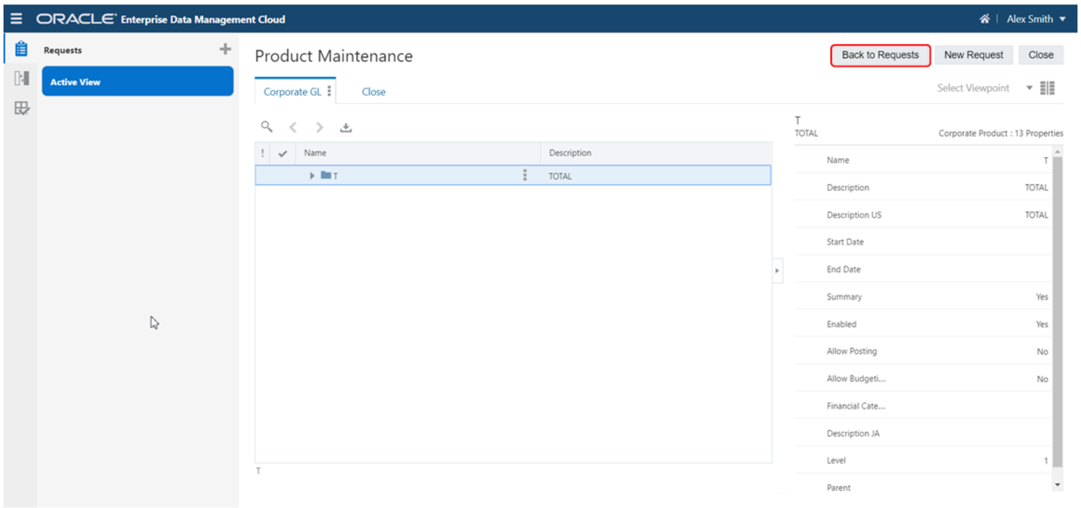
T > 200 > 220 > 224

224
Smart Watch
Corporate Product : 13 Properties

Name	224
Description	Smart Watch
Description US	Smart Watch
tart Date	
End Date	
Summary	No
Enabled	Yes
Allow Posting	Yes
Allow Budgeti...	Yes
Financial Cate...	
Description JA	
Level	4

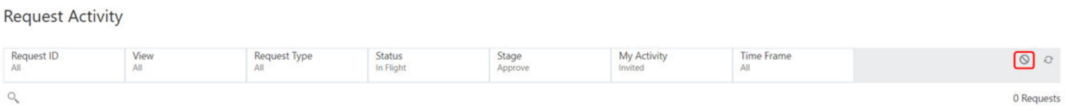
After you approve the request, subscription requests are generated for the Financial Close and Corporate Planning applications.

7. Let's review the subscription requests that were generated. Click **Back to Requests**.



Name	T
Description	TOTAL
Description US	TOTAL
Start Date	
End Date	
Summary	Yes
Enabled	Yes
Allow Posting	No
Allow Budgeti...	No
Financial Cate...	
Description JA	
Level	1
Parent	

8. Click the **Reset All Filters** button.

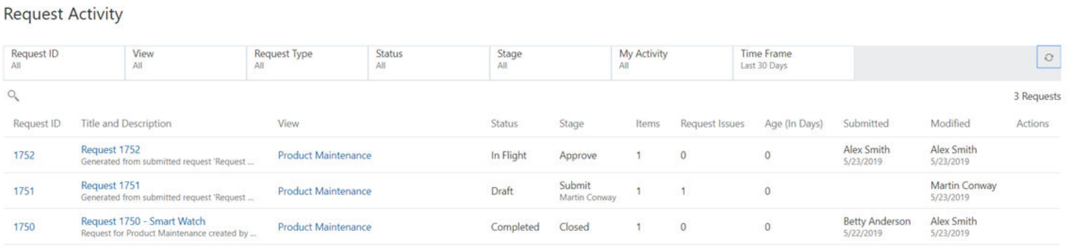

Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	In Flight	Approve	Invited	All

0 Requests

9. Review the statuses of the following requests:

   - Request 1750 is **Completed**. This is the request that Betty entered for the Corporate GL application.
   - Request 1751 is in **Draft**. This is the subscription request that was generated for the Corporate Planning application; however, a validation error prevented it from being submitted. Martin, the Corporate Planning data manager, will need to take action to correct the validation error.
   - Request 1752 is **In Flight**. This is the subscription request that was generated for Financial Close. It is waiting to be approved by Kerry, the data manager.


Request Activity

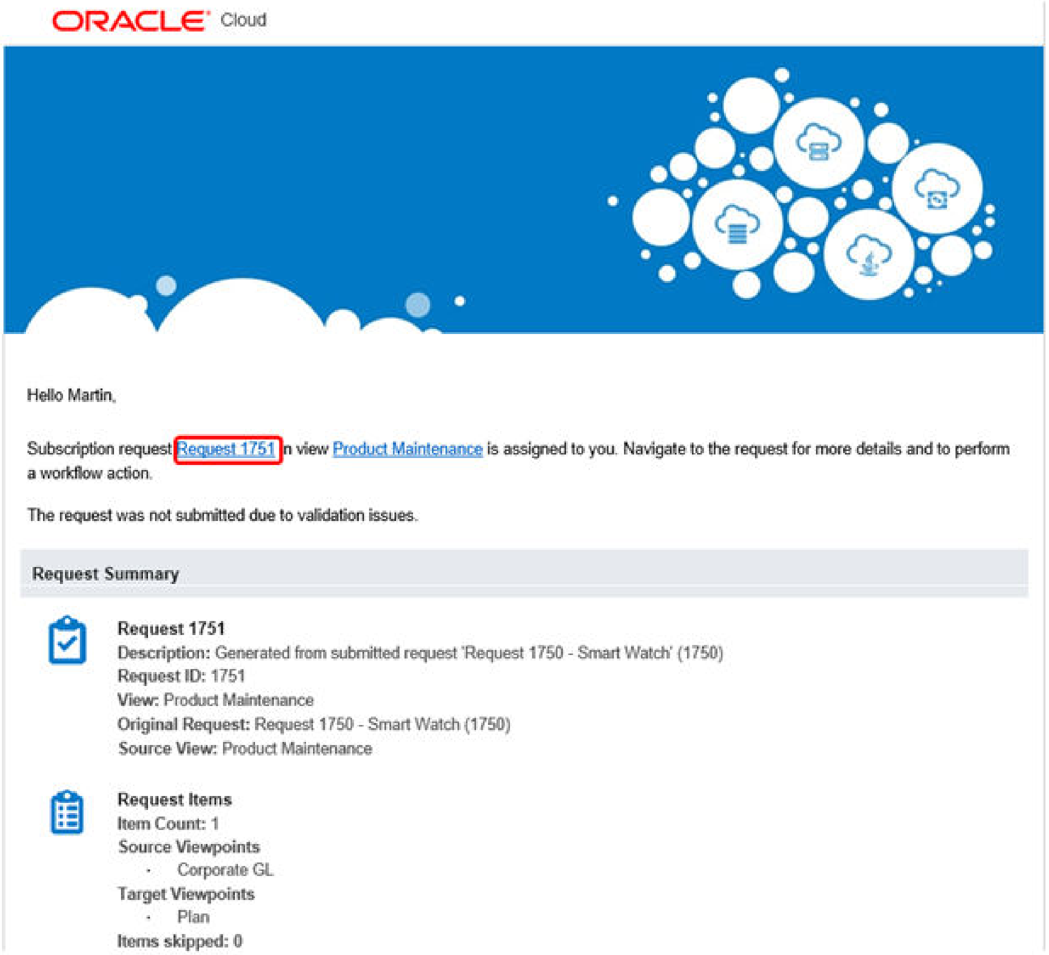
Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	All	All	All	Last 30 Days

3 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1752	Request 1752 Generated from submitted request 'Request ...	Product Maintenance	In Flight	Approve	1	0	0	Alex Smith 5/23/2019	Alex Smith 5/23/2019	
1751	Request 1751 Generated from submitted request 'Request ...	Product Maintenance	Draft	Submit Martin Conway	1	1	0		Martin Conway 5/23/2019	
1750	Request 1750 - Smart Watch Request for Product Maintenance created by ...	Product Maintenance	Completed	Closed	1	0	0	Betty Anderson 5/22/2019	Alex Smith 5/23/2019	

Corporate Planning Data Manager Steps

1. In the email notification of the request validation issue, a statement is displayed that the request was not submitted due to validation issues. Click the link to navigate to the request.



Hello Martin,

Subscription request Request 1751 in view Product Maintenance is assigned to you. Navigate to the request for more details and to perform a workflow action.

The request was not submitted due to validation issues.

Request Summary

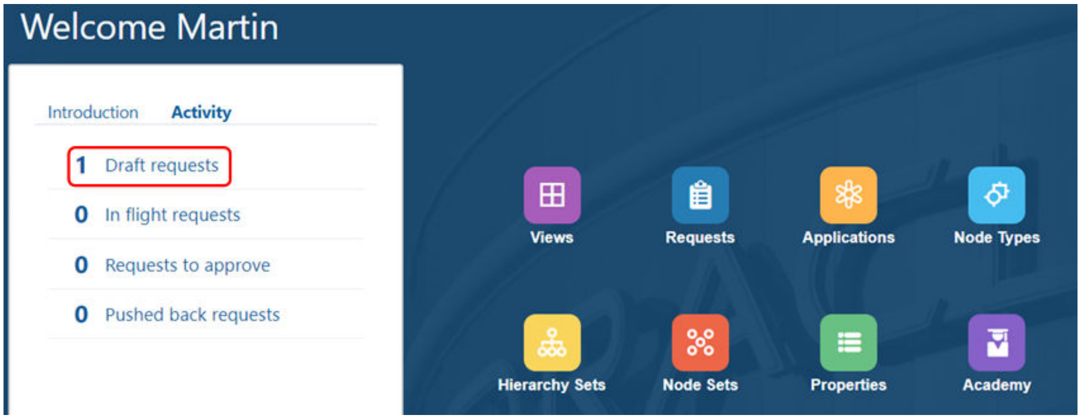
Request 1751
Description: Generated from submitted request 'Request 1750 - Smart Watch' (1750)
Request ID: 1751
View: Product Maintenance
Original Request: Request 1750 - Smart Watch (1750)
Source View: Product Maintenance

Request Items
Item Count: 1
Source Viewpoints
- Corporate GL

Target Viewpoints
- Plan

Items skipped: 0

2. In the Activity List, click **Draft Requests**.

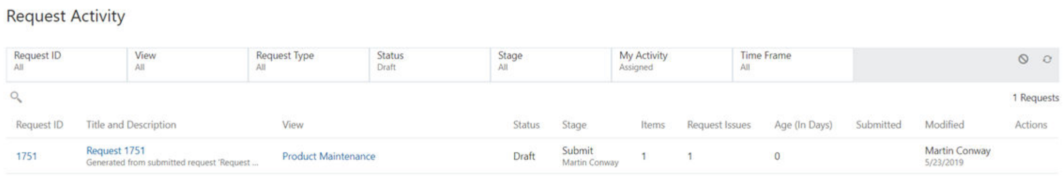


Welcome Martin

Introduction Activity

1 Draft requests
0 In flight requests
0 Requests to approve
0 Pushed back requests

3. In Request Activity, click request 1751 to open it in the maintenance view.



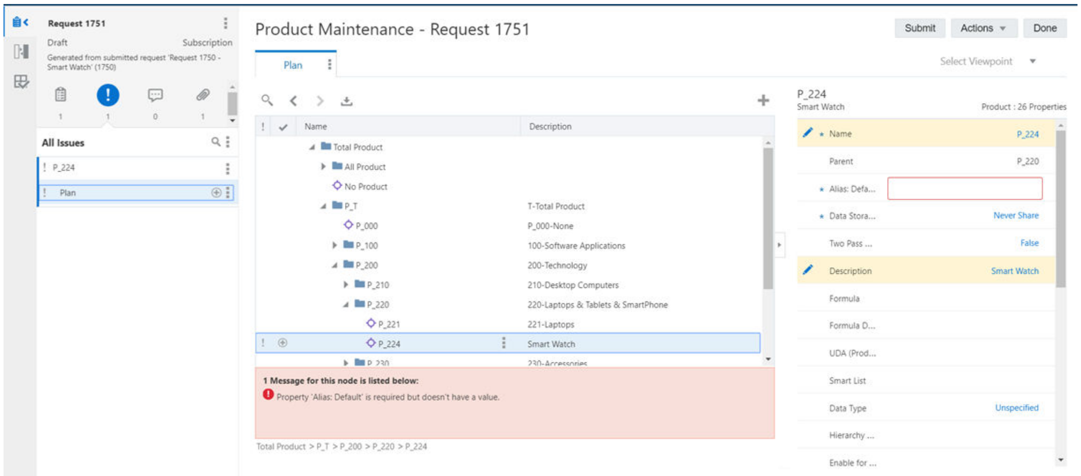
Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	Draft	All	Assigned	All

1 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1751	Request 1751 Generated from submitted request 'Request ...	Product Maintenance	Draft	Submit Martin Conway	1	1	0		Martin Conway 5/23/2019	

4. Click **Request Issues**, then expand the request issue to display the validation error. In this scenario, the request is missing a required property, Alias: Default.



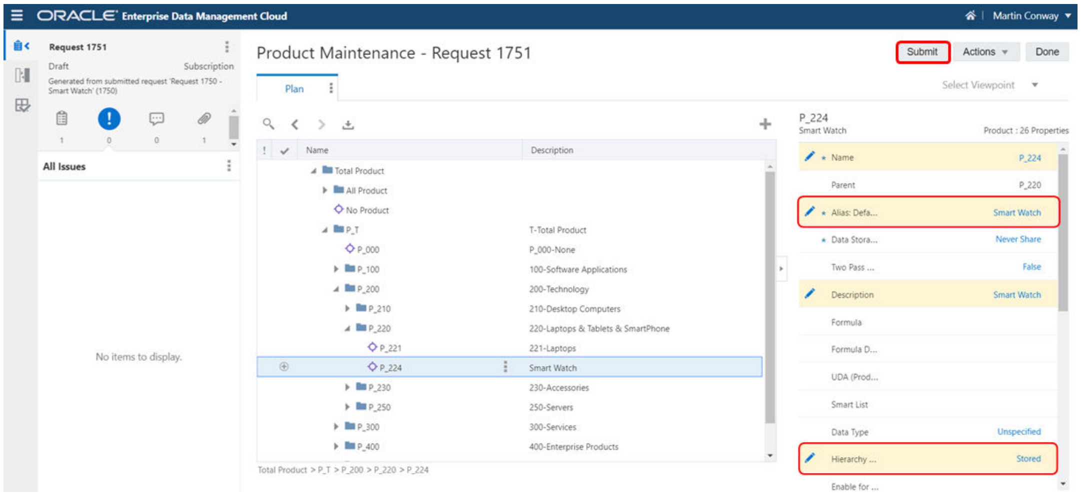
1 Message for this node is listed below:

Property 'Alias: Default' is required but doesn't have a value.

P_224
Smart Watch
Product : 26 Properties

Name	P_224
Parent	P_220
Alias: Defa...	
Data Stora...	Never Share
Two Pass ...	False
Description	Smart Watch
Formula	
Formula D...	
UDA (Prod...	
Smart List	
Data Type	Unspecified
Hierarchy ...	
Enable for ...	

5. In the property panel, for **Alias: Default** enter `Smart Watch`. Optionally, you can further enrich the request. In this scenario, we selected `Stored` for the **Hierarchy Type**. When you are finished, click **Submit**.



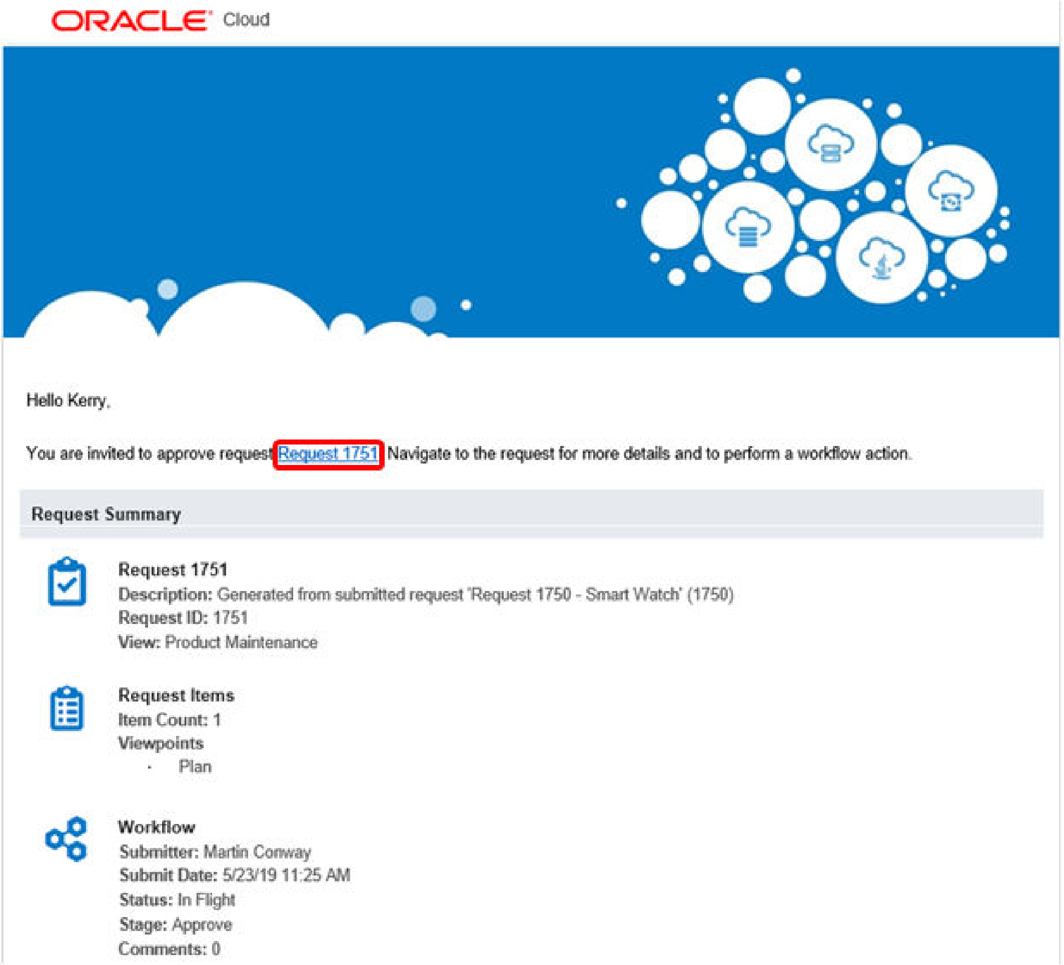
P_224
Smart Watch
Product : 26 Properties

Name	P_224
Parent	P_220
Alias: Defa...	Smart Watch
Data Stora...	Never Share
Two Pass ...	False
Description	Smart Watch
Formula	
Formula D...	
UDA (Prod...	
Smart List	
Data Type	Unspecified
Hierarchy ...	Stored
Enable for ...	

With the validation issue fixed, the request is submitted to Kerry, the enterprise data manager, for approval.

Enterprise Data Manager Steps

1. As the enterprise data manager, you will have received email requests for approval for both of the subscription requests. In either email notification, click the link to navigate to the request.



Hello Kerry,

You are invited to approve request Request 1751 Navigate to the request for more details and to perform a workflow action.

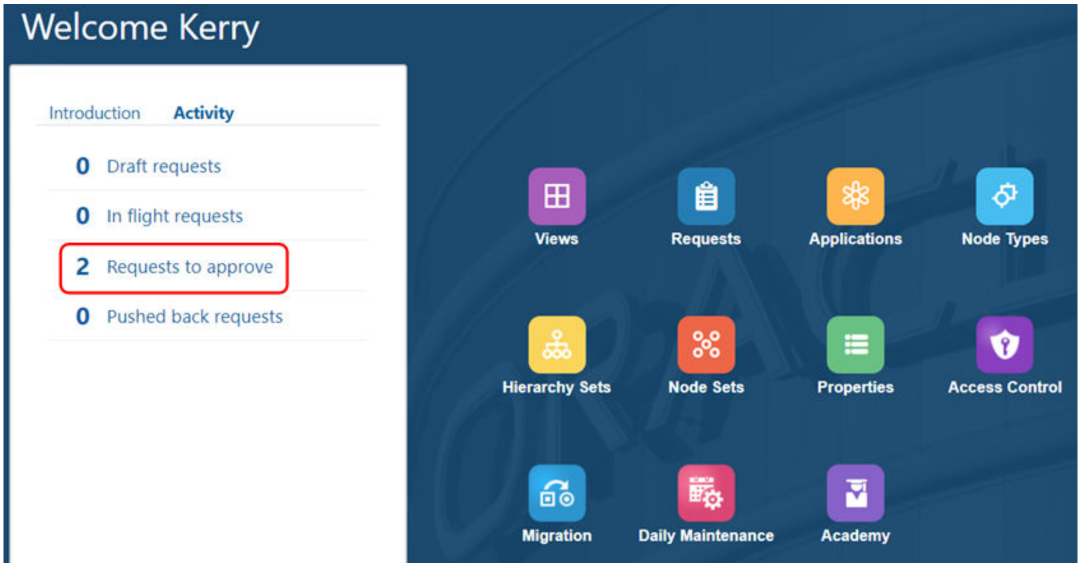
Request Summary

Request 1751
Description: Generated from submitted request 'Request 1750 - Smart Watch' (1750)
Request ID: 1751
View: Product Maintenance

Request Items
Item Count: 1
Viewpoints
- Plan

Workflow
Submitter: Martin Conway
Submit Date: 5/23/19 11:25 AM
Status: In Flight
Stage: Approve
Comments: 0

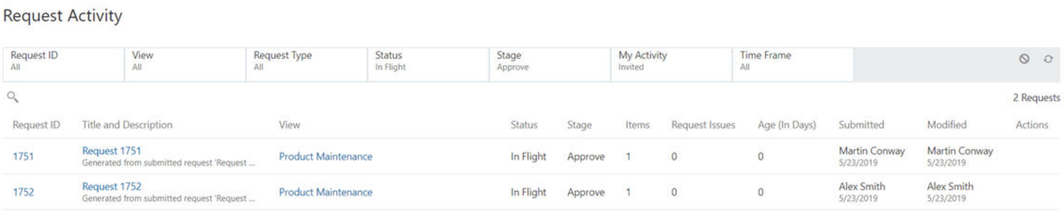
2. Click **Requests to Approve**.


Welcome Kerry

Introduction Activity

0 Draft requests
0 In flight requests
2 Requests to approve
0 Pushed back requests

3. In Request Activity, click **Request 1751** to approve it.



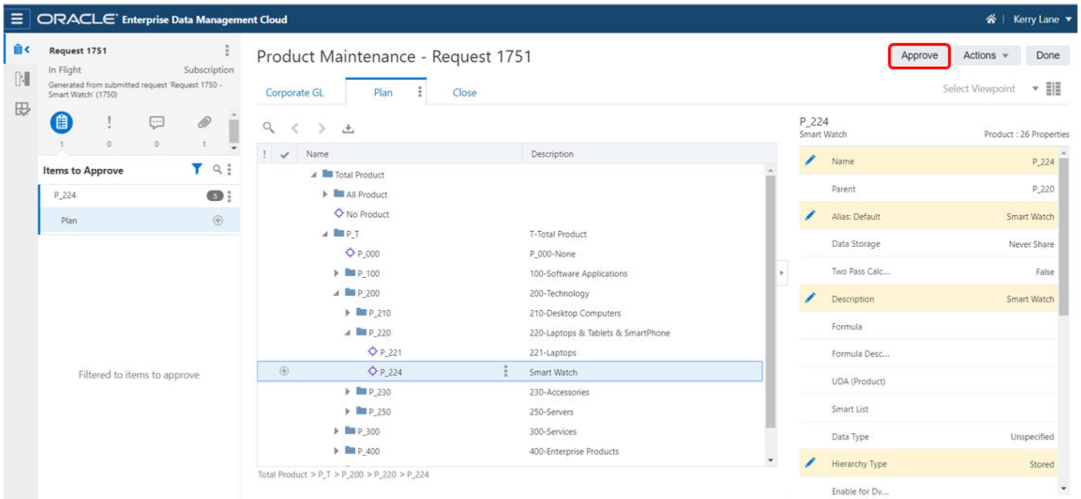
Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	In Flight	Approve	Invited	All

2 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1751	Request 1751 Generated from submitted request 'Request ...	Product Maintenance	In Flight	Approve	1	0	0	Martin Conway 5/23/2019	Martin Conway 5/23/2019	
1752	Request 1752 Generated from submitted request 'Request ...	Product Maintenance	In Flight	Approve	1	0	0	Alex Smith 5/23/2019	Alex Smith 5/23/2019	

4. Review the request details, and then click **Approve**.


ORACLE Enterprise Data Management Cloud
Kerry Lane
Request 1751
In Flight
Subscription
Generated from submitted request 'Request 1750 - Smart Watch' (1750)
Items to Approve
P_224
Plan
Filtered to items to approve
Approve
Actions
Done
Corporate GL
Plan
Close
Select Viewpoint

Product Maintenance - Request 1751

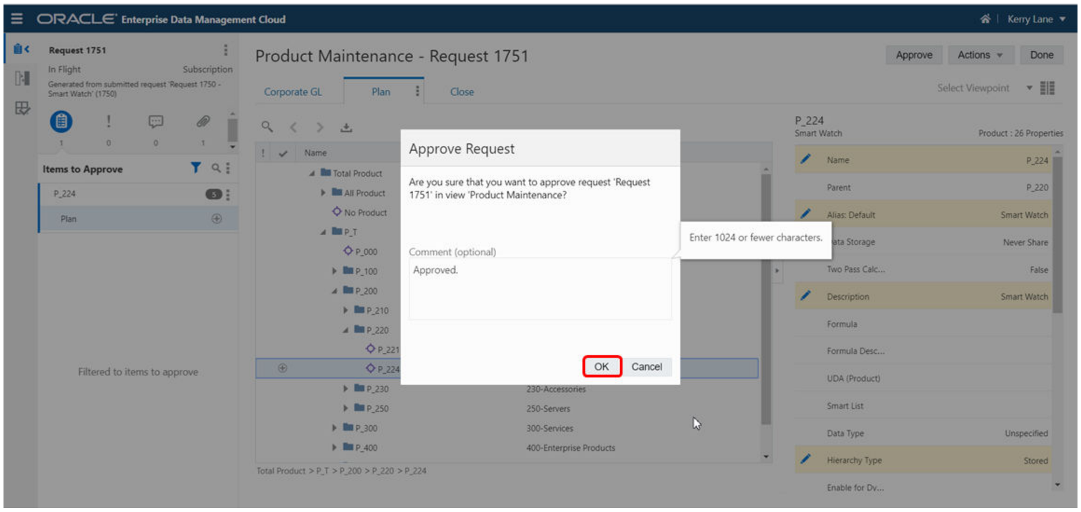
Name	Description
Total Product	
All Product	
No Product	
P_T	T-Total Product
P_000	P_000-None
P_100	100-Software Applications
P_200	200-Technology
P_210	210-Desktop Computers
P_220	220-Laptops & Tablets & SmartPhone
P_221	221-Laptops
P_224	Smart Watch
P_230	230-Accessories
P_250	250-Servers
P_300	300-Services
P_400	400-Enterprise Products

Total Product > P_T > P_200 > P_220 > P_224

P_224
Smart Watch
Product : 26 Properties

Name	P_224
Parent	P_220
Alias: Default	Smart Watch
Data Storage	Never Share
Two Pass Calc...	False
Description	Smart Watch
Formula	
Formula Desc...	
UDA (Product)	
Smart List	
Data Type	Unspecified
Hierarchy Type	Stored
Enable for Dv...	

5. Optionally enter a comment, and then click **OK**.



Approve Request

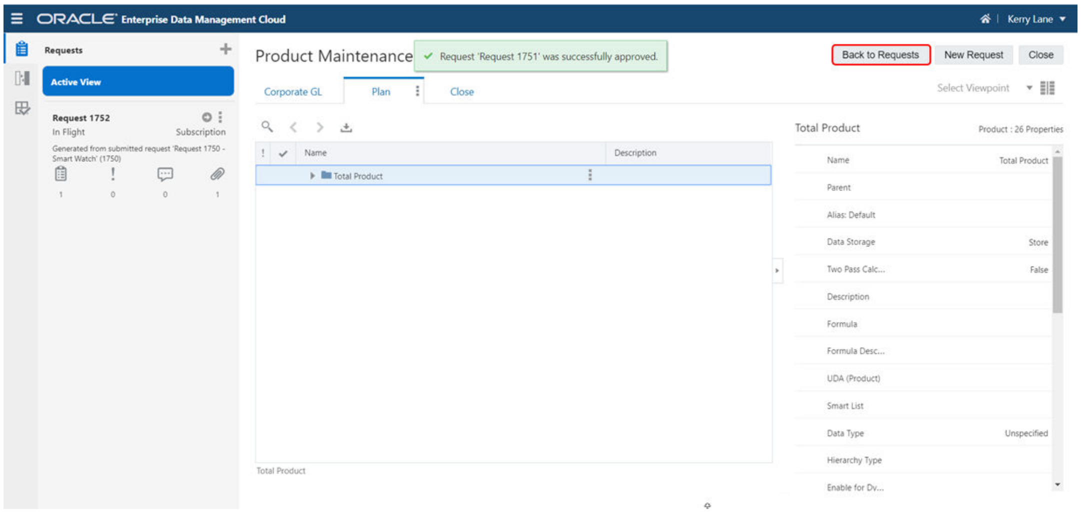
Are you sure that you want to approve request 'Request 1751' in view 'Product Maintenance?

Comment (optional)

Approved.

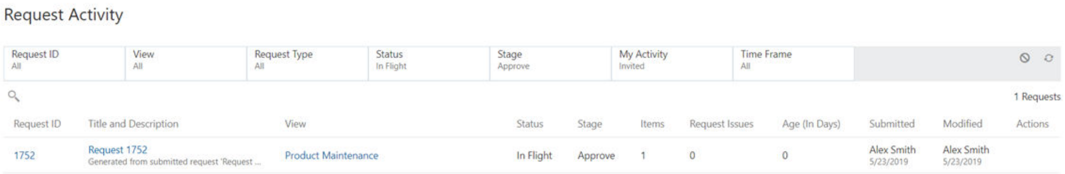
Enter 1024 or fewer characters.

6. Click **Back to Requests**.



Request 'Request 1751' was successfully approved.

7. In Request Activity, select **Request 1752** to approve it.



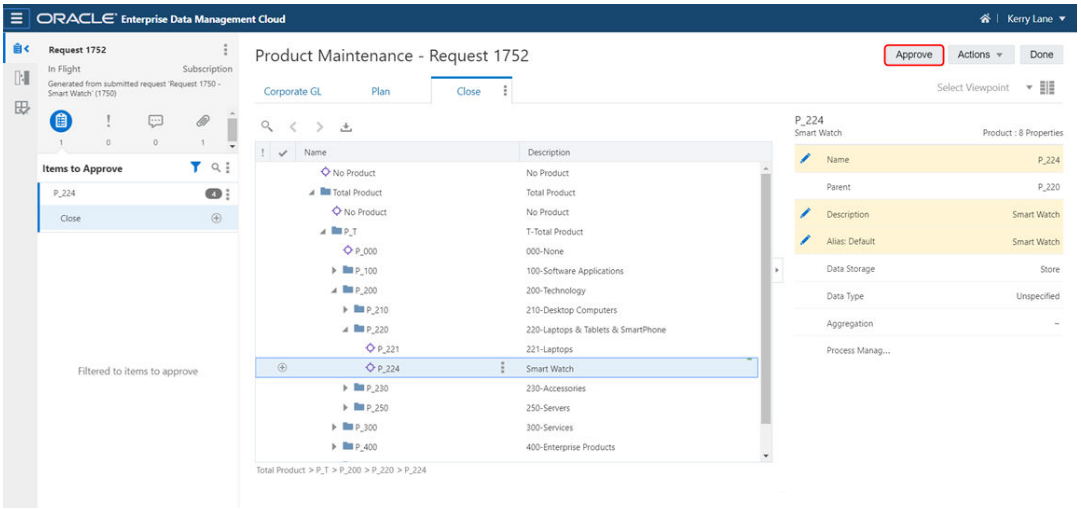
Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	In Flight	Approve	Invited	All

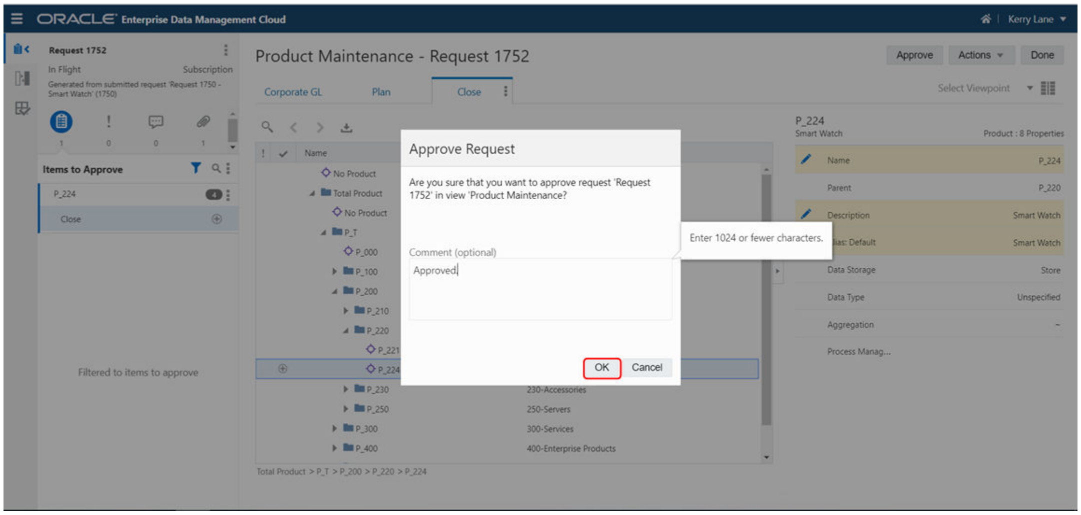
1 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1752	Request 1752 Generated from submitted request 'Request ...	Product Maintenance	In Flight	Approve	1	0	0	Alex Smith 5/23/2019	Alex Smith 5/23/2019	

8. Review the request details, and then click **Approve**.



9. Optionally enter a comment, and then click **OK**.



Approve Request

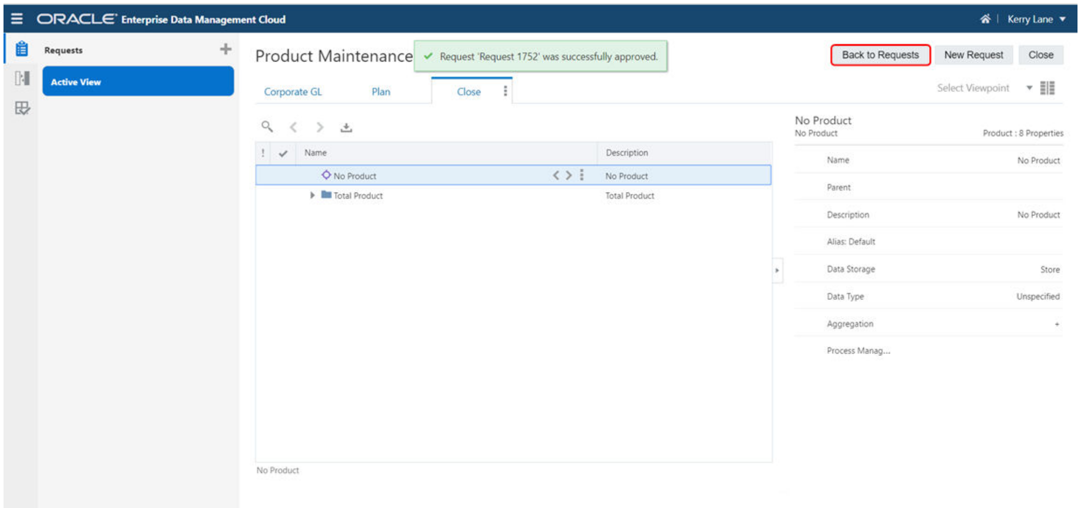
Are you sure that you want to approve request 'Request 1752' in view 'Product Maintenance?

Comment (optional)

Approved

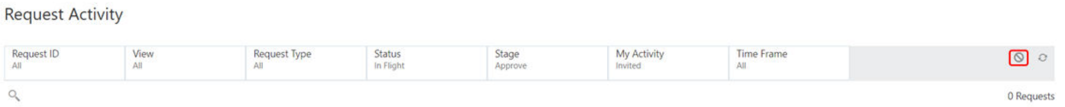
Enter 1024 or fewer characters.

10. Let's review the request statuses. Click **Back to Requests**.



Request 'Request 1752' was successfully approved.

11. In Request Activity, click the **Clear All Filters** button.

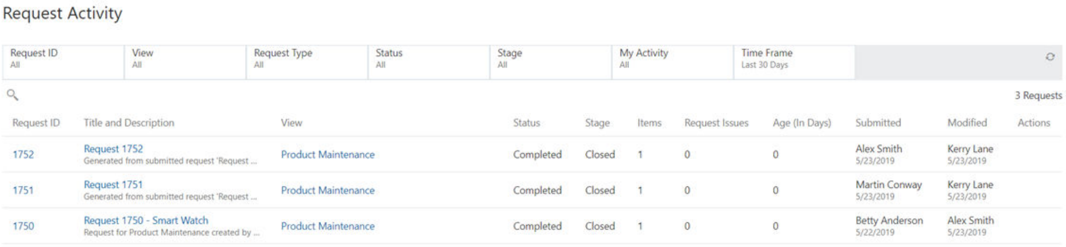


Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	In Flight	Approve	Invited	All

0 Requests

12. Verify that all three requests have a status of Closed.



Request Activity

Request ID	View	Request Type	Status	Stage	My Activity	Time Frame
All	All	All	All	All	All	Last 30 Days

3 Requests

Request ID	Title and Description	View	Status	Stage	Items	Request Issues	Age (In Days)	Submitted	Modified	Actions
1752	Request 1752 Generated from submitted request 'Request ...	Product Maintenance	Completed	Closed	1	0	0	Alex Smith 5/23/2019	Kerry Lane 5/23/2019	
1751	Request 1751 Generated from submitted request 'Request ...	Product Maintenance	Completed	Closed	1	0	0	Martin Conway 5/23/2019	Kerry Lane 5/23/2019	
1750	Request 1750 - Smart Watch Request for Product Maintenance created by ...	Product Maintenance	Completed	Closed	1	0	0	Betty Anderson 5/22/2019	Alex Smith 5/23/2019	

52 Transferring Applications Between Environments

In this scenario, we want to add a new Planning application called **Corporate Planning - UK** to our production environment. We have an existing test environment that we can use to build the new application, but before we build it there we want to refresh that test environment with a snapshot of our production environment. This ensures that the test environment is up to date with production.

The general steps for this scenario are as follows:

1. Use *migration* to clone the production environment to the test environment to prepare for new application development.
2. Build the new application in our test environment.
3. Use a *template* to transfer the new application from the test environment to the production environment.

Administrator Steps

Administrator Prerequisites

You must be a Service Administrator to access migration. The Service Administrator role will enable you to perform all of the steps in this scenario.

Administrator Process

1. Take a snapshot of your production environment and download it to your local computer. See Using Migration.
2. In your test environment, recreate the service to start with a clean environment. See Recreating the Service.
3. Refresh the test environment with the snapshot of your production environment. See Using Migration.
4. Build the new application in your test environment. See Understanding Registering Applications.
5. In your test environment, export the new application to a template. See Working with Templates
6. In your production environment, upload and preview the template import and correct any errors. See Previewing Template Imports and Reviewing Import Results.
7. Complete the transfer by importing the application from the template. See Working with Templates.

Step 1: Refreshing and Preparing the Test Environment for Application Development

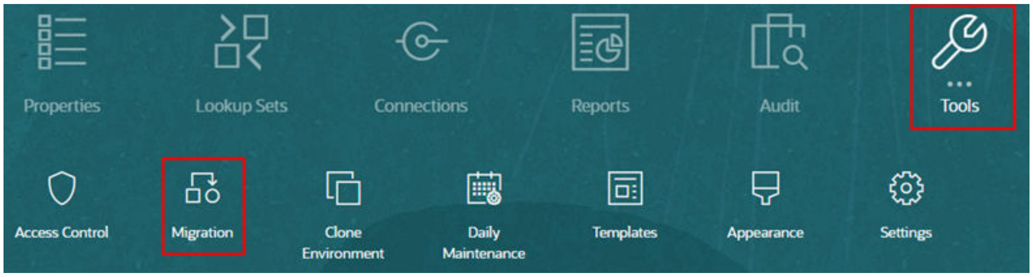
Caution:

This scenario includes steps to be performed in both the production and test environments. Make sure that you are performing the steps in the correct environment before proceeding. The names of the environments are capitalized in each step to help you determine the correct environment for that step.

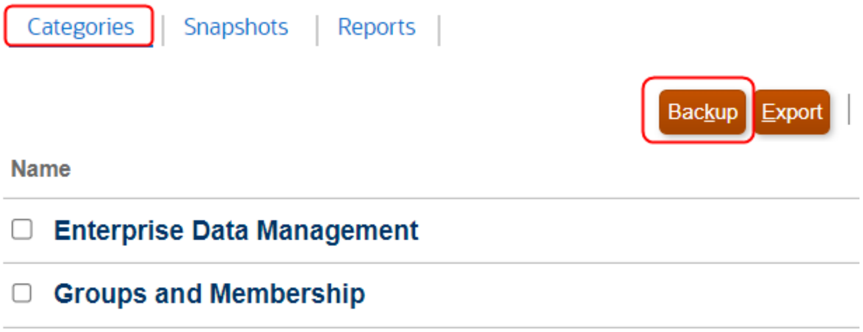
1. Before you refresh your test environment with a snapshot of your production environment, it is a best practice to take a backup snapshot of the test environment to preserve existing test data.

   To back up your TEST environment:

   a. Click **Tools**, and then **Migration**.

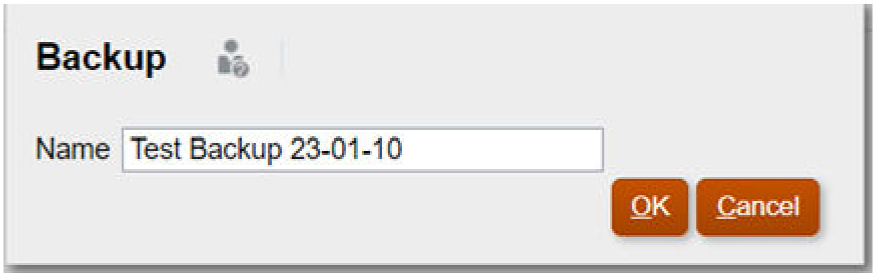


   b. From **Categories**, click **Backup** to create a backup snapshot of your current test environment.



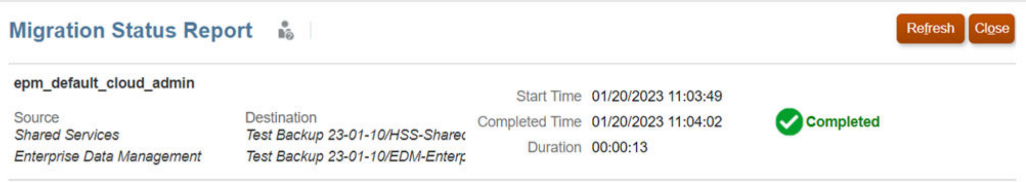
□ Enterprise Data Management
□ Groups and Membership

   c. Enter a name for your backup and click OK.



Name: Test Backup 23-01-10

d. Click **Refresh** to update the status of the backup, and then click **Close** to close the window when the backup is completed.

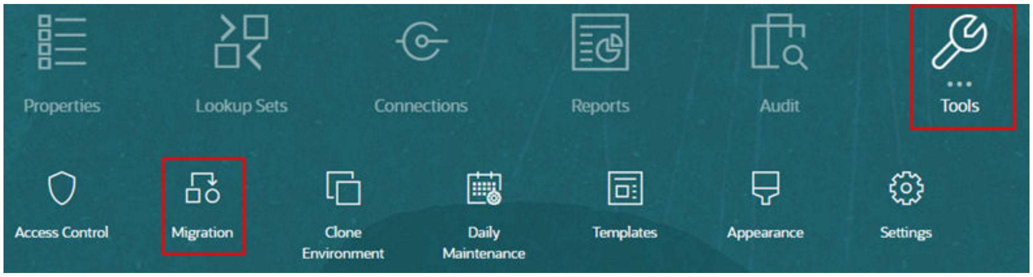


epm_default_cloud_admin

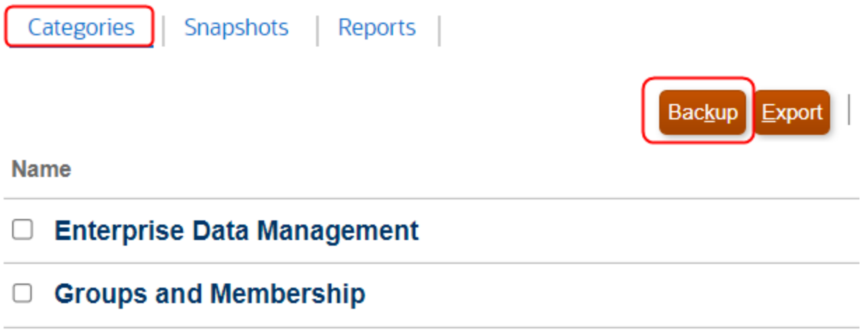
Source	Destination			
Shared Services	*Test Backup 23-01-10/HSS-Sharec*	Start Time	01/20/2023 11:03:49	
Enterprise Data Management	*Test Backup 23-01-10/EDM-Enterp*	Completed Time	01/20/2023 11:04:02	✓ Completed
		Duration	00:00:13	

2. Take a migration snapshot of your PRODUCTION environment.

a. Click **Tools**, and then **Migration**.

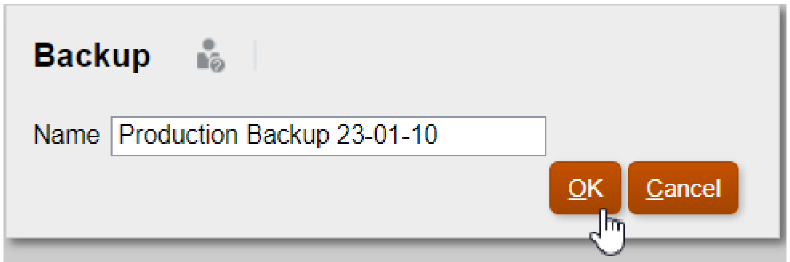


b. From **Categories**, click **Backup** to create a migration snapshot of your current production environment.



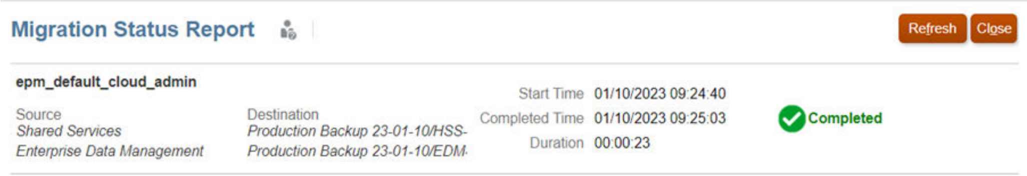
	Name
□	Enterprise Data Management
□	Groups and Membership

c. Enter a name for the snapshot, and then click **OK**.



Name: Production Backup 23-01-10

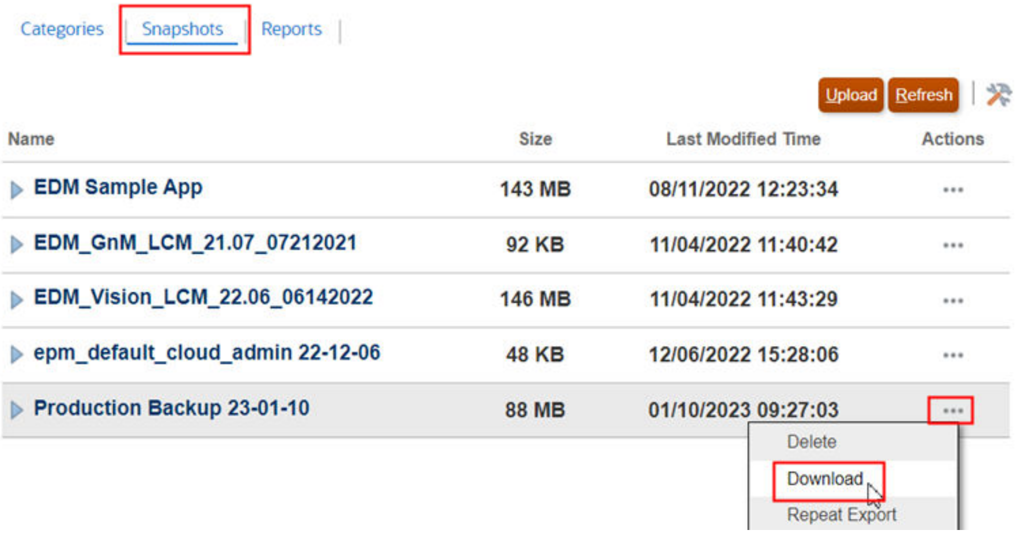
d. Click **Refresh** to update the status of the backup, and then click **Close** to close the window when the backup is complete.



epm_default_cloud_admin

Source	Destination			
Shared Services	*Production Backup 23-01-10/HSS-*	Start Time	01/10/2023 09:24:40	✓ Completed
Enterprise Data Management	*Production Backup 23-01-10/EDM-*	Completed Time	01/10/2023 09:25:03	
		Duration	00:00:23	

e. From **Snapshots**, locate the backup, and then in the **Actions** column click ••• and then click **Download**.



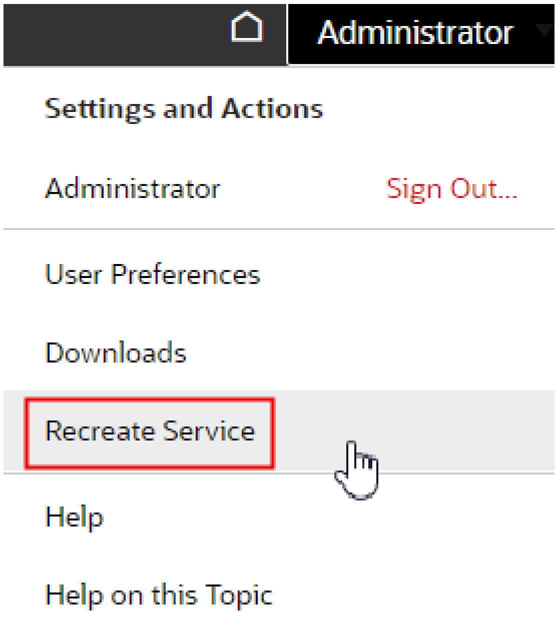
Name	Size	Last Modified Time	Actions
▷ EDM Sample App	143 MB	08/11/2022 12:23:34	•••
▷ EDM_GnM_LCM_21.07_07212021	92 KB	11/04/2022 11:40:42	•••
▷ EDM_Vision_LCM_22.06_06142022	146 MB	11/04/2022 11:43:29	•••
▷ epm_default_cloud_admin 22-12-06	48 KB	12/06/2022 15:28:06	•••
▷ Production Backup 23-01-10	88 MB	01/10/2023 09:27:03	•••

The migration snapshot is downloaded to your local machine.

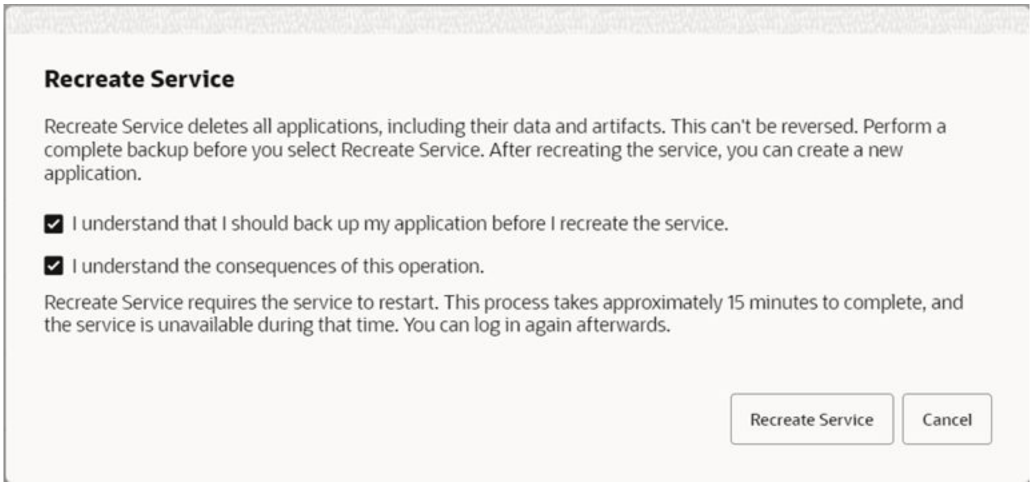
3. In your TEST environment, recreate the service. This is an optional step, but recreating the environment will establish a baseline starting point for the environment and will resynchronize the latest users and group assignments from the Identity Cloud Service (IDCS) for your test environment.

> **Caution:**
>
> Recreating the service will remove all existing applications and artifacts, including test users and groups, in your test environment. If you want to preserve your test data, ensure that you follow the procedure in Step 1 to back up your existing test environment first.

a. From the TEST environment, click the drop down menu next to your user name, and then select **Recreate Service**.



b. On the confirmation screen, select the check boxes indicating that you understand the consequences of recreating your service, and then click **Recreate Service**.



Recreate Service

Recreate Service deletes all applications, including their data and artifacts. This can't be reversed. Perform a complete backup before you select Recreate Service. After recreating the service, you can create a new application.

☑ I understand that I should back up my application before I recreate the service.

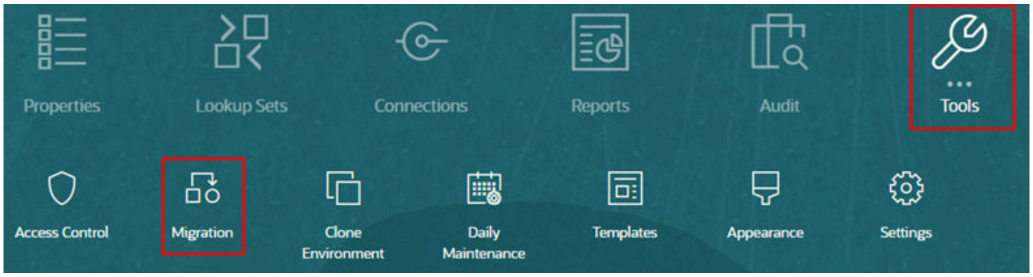
☑ I understand the consequences of this operation.

Recreate Service requires the service to restart. This process takes approximately 15 minutes to complete, and the service is unavailable during that time. You can log in again afterwards.

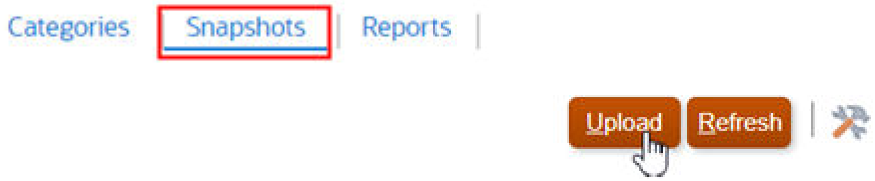
The recreate process takes approximately 15 minutes to complete.

4. When the recreate process has completed, upload your migration snapshot from your local file system.

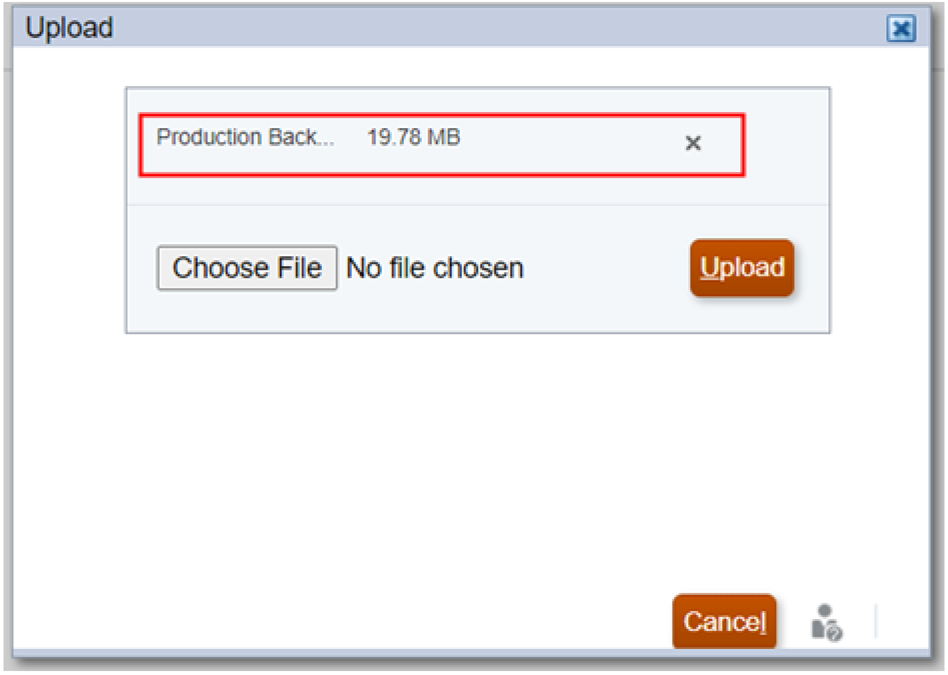
   a. Choose the **Migration** option, or from the Home page click **Tools**, and then **Migration**.



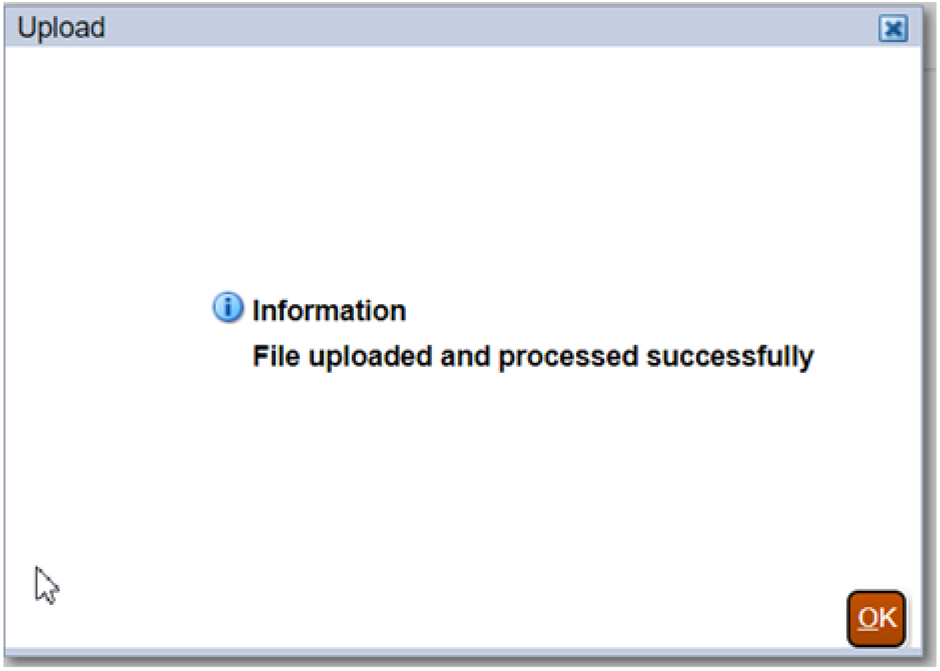
b. From Snapshots, click **Upload**.



c. Select the backup file, and click **Upload**.



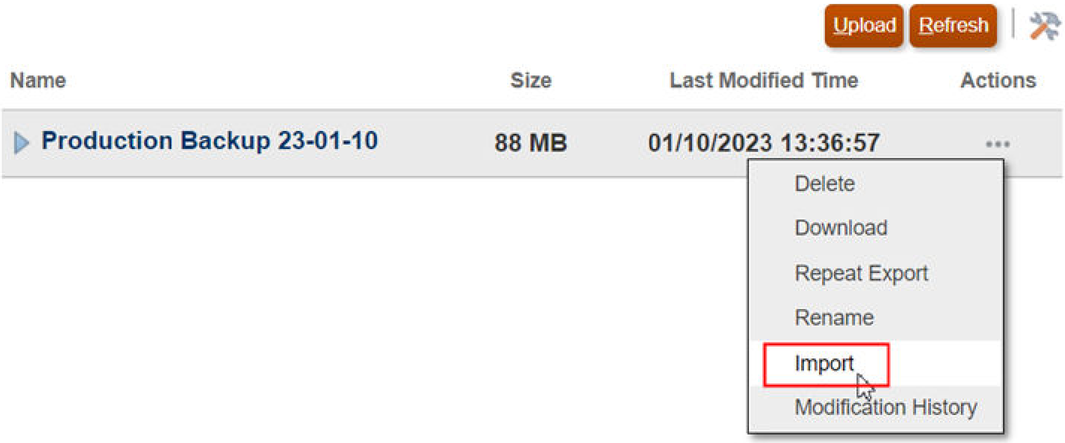
d. When the file has completed uploading, click **OK**.



Information

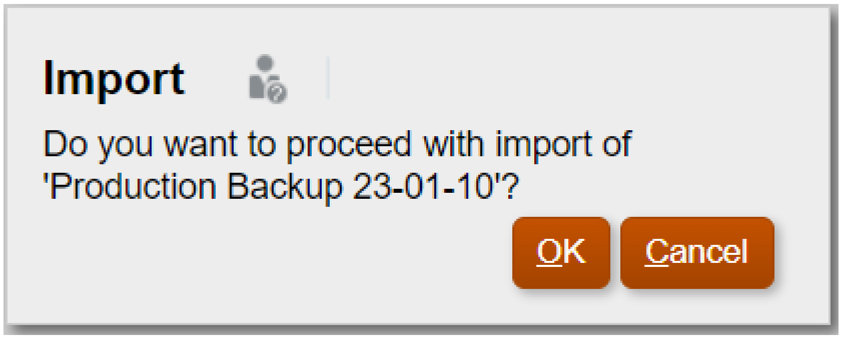
File uploaded and processed successfully

5. On the Snapshots tab, locate the uploaded snapshot and in the **Actions** column, click ••• and then click **Import**.



Name	Size	Last Modified Time	Actions
Production Backup 23-01-10	88 MB	01/10/2023 13:36:57	•••

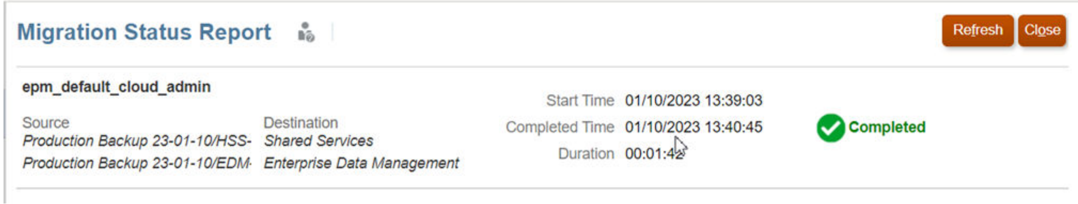
6. Click **OK** on the confirmation message.



Import

Do you want to proceed with import of 'Production Backup 23-01-10'?

7. On the Migration Status screen, click **Refresh** until the import is completed, and then click **Close**.



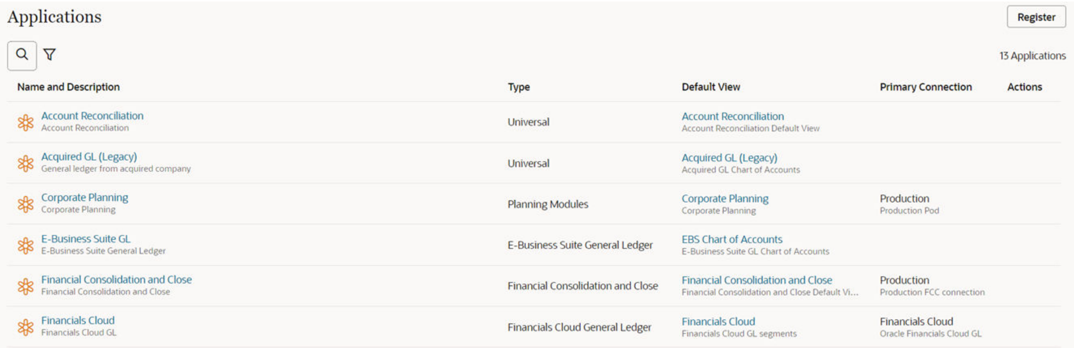
epm_default_cloud_admin

Source	Destination
Production Backup 23-01-10/HSS-	Shared Services
Production Backup 23-01-10/EDM-	Enterprise Data Management

Start Time 01/10/2023 13:39:03
Completed Time 01/10/2023 13:40:45
Duration 00:01:42

Completed

8. Verify that your TEST environment contains the applications from your PRODUCTION environment.



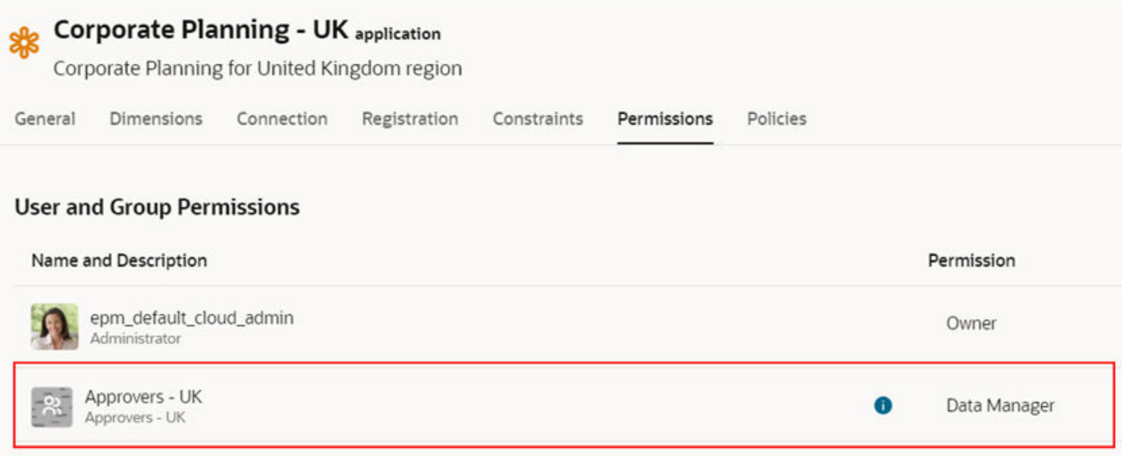
Name and Description	Type	Default View	Primary Connection	Actions
Account Reconciliation Account Reconciliation	Universal	Account Reconciliation Account Reconciliation Default View		
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts		
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning	Production Production Pod	
E-Business Suite GL E-Business Suite General Ledger	E-Business Suite General Ledger	EBS Chart of Accounts E-Business Suite GL Chart of Accounts		
Financial Consolidation and Close Financial Consolidation and Close	Financial Consolidation and Close	Financial Consolidation and Close Financial Consolidation and Close Default Vi...	Production Production FCC connection	
Financials Cloud Financials Cloud GL	Financials Cloud General Ledger	Financials Cloud Financials Cloud GL segments	Financials Cloud Oracle Financials Cloud GL	

Step 2: Building the Application

In this scenario, we will register and build a Planning application in our test environment. The individual steps for registering a Planning application are documented in the Integrating with Planning business scenario. Our Planning application has the following details:

- Name: **Corporate Planning - UK**
- Cubes: **Plan1**
- Aliases: **Default**, **EPMO**
- Dimensions:
  - **Account**
  - **Entity**
  - **Cost Center**
  - **Product**

We have also created a new group, **Approvers - UK**, and assigned that group the *Data Manager* permission on the new application.



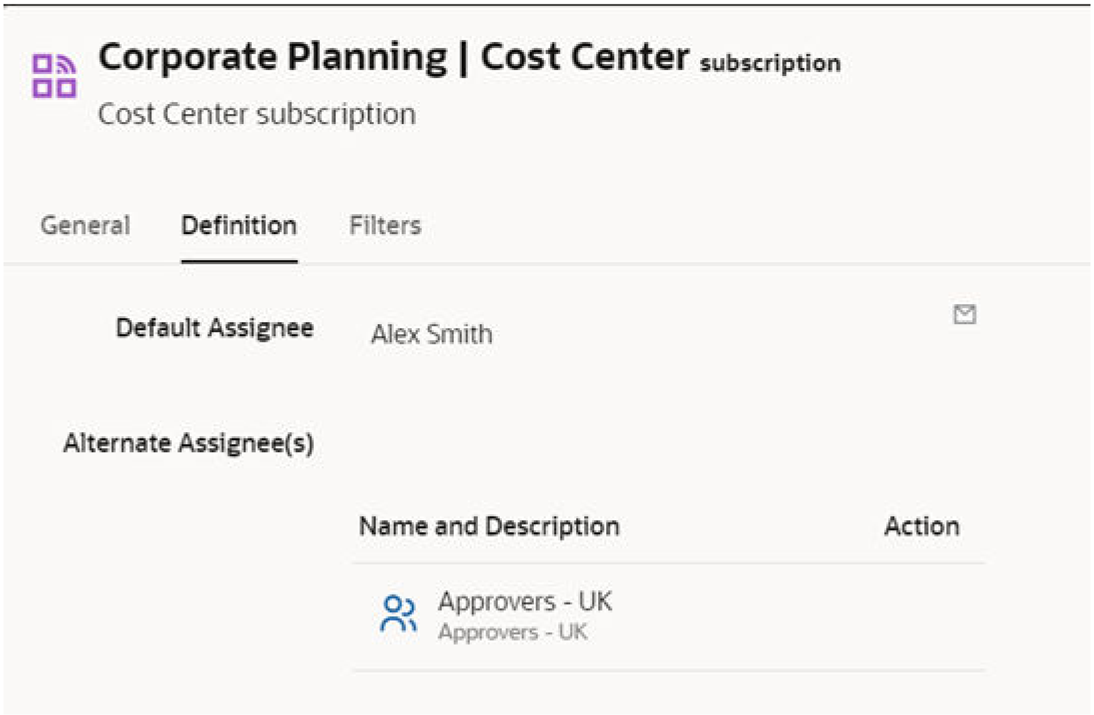
Corporate Planning - UK application

Corporate Planning for United Kingdom region

User and Group Permissions

Name and Description	Permission
epm_default_cloud_admin Administrator	Owner
Approvers - UK Approvers - UK	Data Manager

We created a subscription from the Cost Center viewpoint in our new application to the Cost Center viewpoint in the Corporate Planning application. We assigned the new group that we created above as the Alternate Assignees.



Corporate Planning | Cost Center subscription

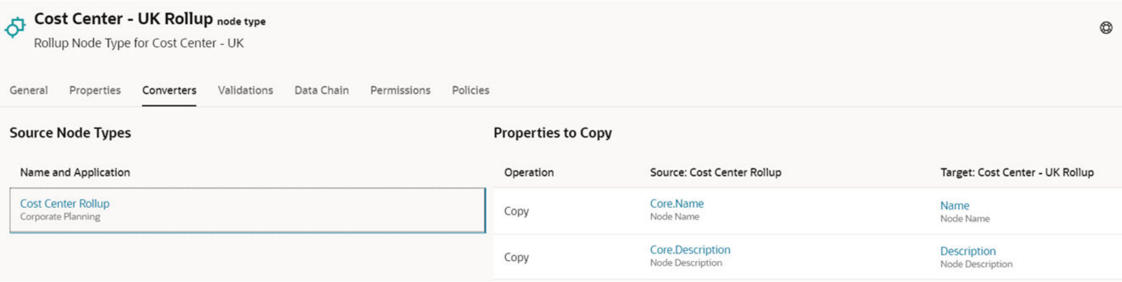
Cost Center subscription

Default Assignee: Alex Smith

Alternate Assignee(s)

Name and Description	Action
Approvers - UK Approvers - UK	

Finally, we created a Cost Center Rollup node type in the Corporate Planning application and a Cost Center - UK Rollup node type in our new Corporate Planning - UK application, and then we created a node type converter between them.



Cost Center - UK Rollup node type
Rollup Node Type for Cost Center - UK

Source Node Types

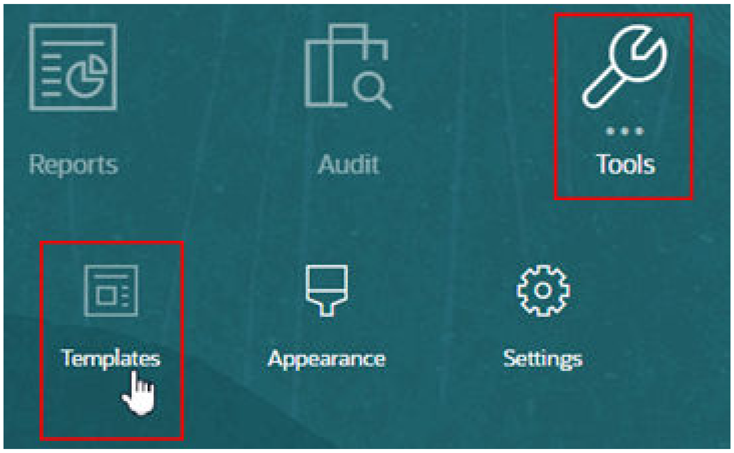
Name and Application
Cost Center Rollup Corporate Planning

Properties to Copy

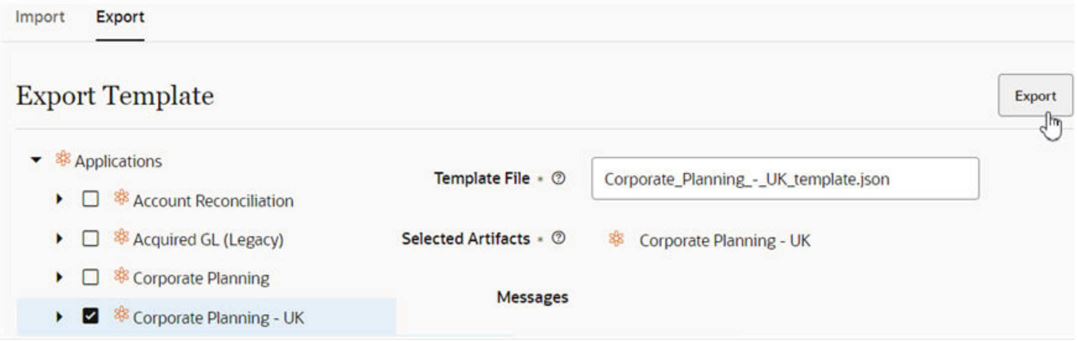
Operation	Source: Cost Center Rollup	Target: Cost Center - UK Rollup
Copy	Core.Name Node Name	Name Node Name
Copy	Core.Description Node Description	Description Node Description

Step 3: Exporting the New Application to a Template

1. In the TEST environment, click **Tools**, and then **Templates**.



2. On the **Export** tab:
   a. Expand **Applications**.
   b. Select the checkbox next to your new application.
   c. In **Template File**, enter a file name.
   d. Click **Export**.



Export Template

- Applications
  - □ Account Reconciliation
  - □ Acquired GL (Legacy)
  - □ Corporate Planning
  - ☑ Corporate Planning - UK

Template File * Corporate_Planning_-_UK_template.json

Selected Artifacts * Corporate Planning - UK

Messages

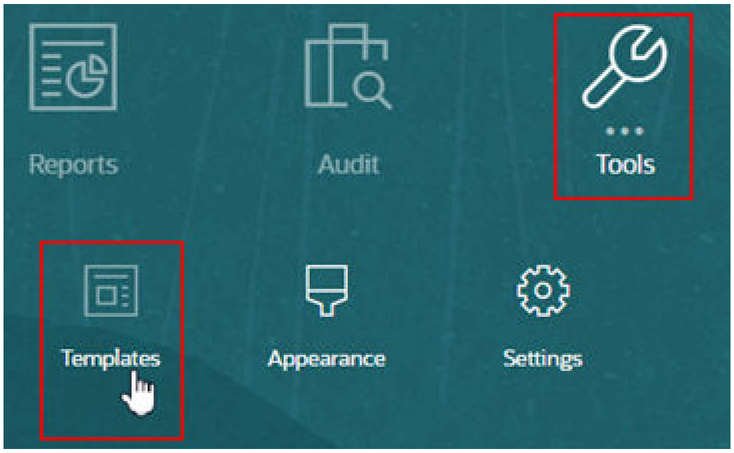
The application is exported to a `.json` file and downloaded to your local system.

Step 4: Previewing the Import of an Application from a Template

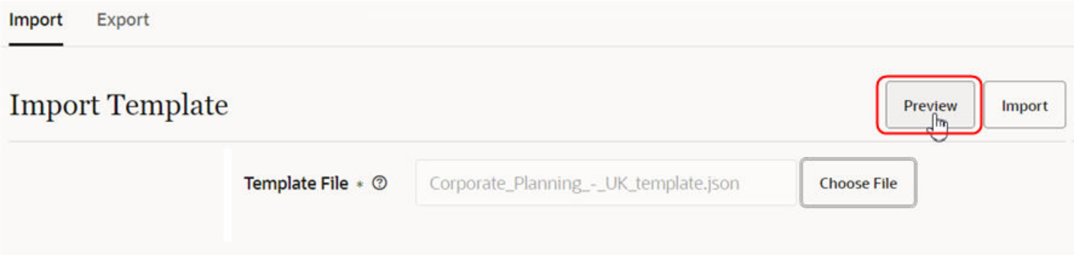
Before you import applications or dimensions from templates, it is important that you preview the import first. Previewing the import enables you to visualize the changes that the template import will create in the target environment and, more importantly, it enables you to find and correct any potential errors or warnings with the import before you run the import itself.

To preview a template import:

1. In the PRODUCTION environment, click **Tools**, and then **Templates**.

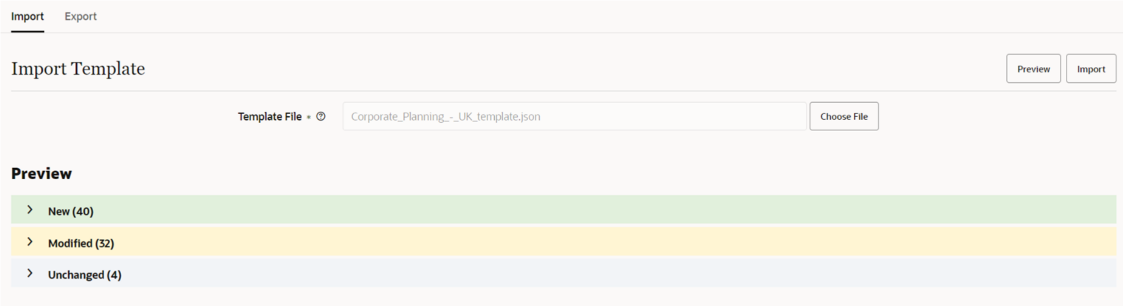


2. On the **Import** tab:
   a. Click **Choose File**.
   b. Navigate to the export file that was downloaded to your local system.
   c. Click **Preview**.



Template File * Corporate_Planning_-_UK_template.json

The preview displays number of New, Modified, and Unchanged artifacts as a result of the import.

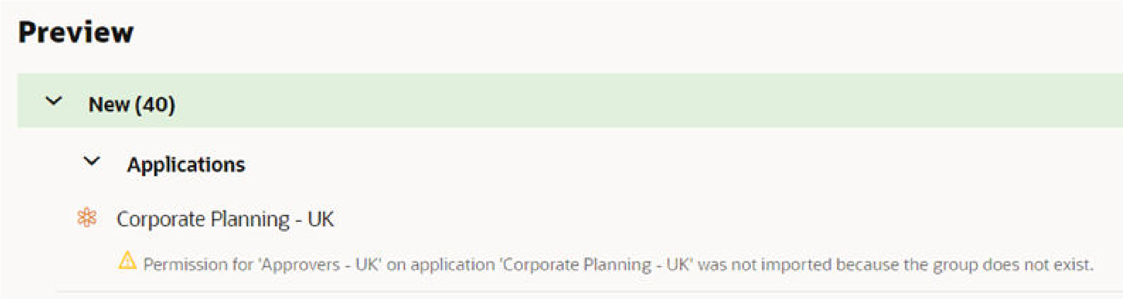


Template File • Corporate_Planning_-_UK_template.json

Reviewing Preview Warnings

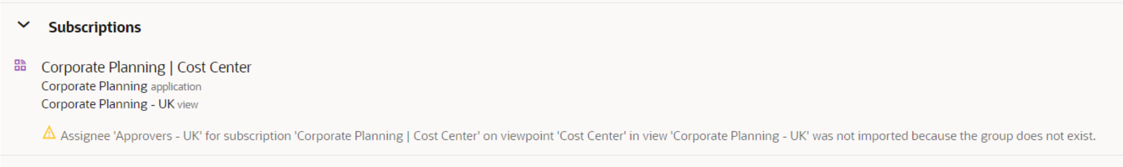
Next, expand the New, Modified, and Unchanged artifacts and review any warning messages. The following warnings were found in this scenario:

Application permission not imported:



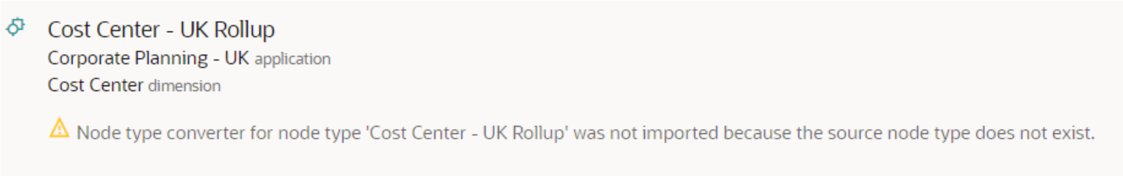
⚠ Permission for 'Approvers - UK' on application 'Corporate Planning - UK' was not imported because the group does not exist.

Viewpoint subscription not imported:


Subscriptions
Corporate Planning | Cost Center
Corporate Planning application
Corporate Planning - UK view

⚠ Assignee 'Approvers - UK' for subscription 'Corporate Planning | Cost Center' on viewpoint 'Cost Center' in view 'Corporate Planning - UK' was not imported because the group does not exist.

Node type converter not imported:



⚠ Node type converter for node type 'Cost Center - UK Rollup' was not imported because the source node type does not exist.

Next, see Step 5: Correcting Preview Import Warnings for information on how to correct these warnings.

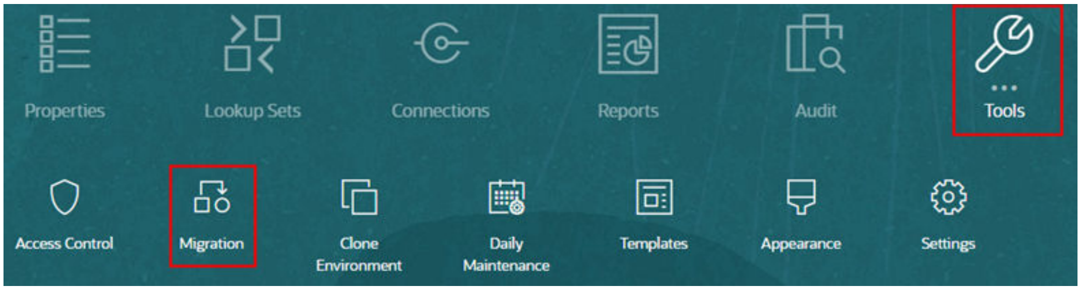
Step 5: Correcting Preview Import Warnings

Correcting Missing Group Warnings

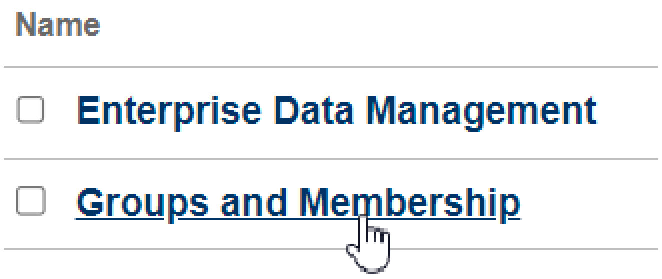
For the application permission and viewpoint subscription warnings, the messages indicate that the permission and subscription aren't able to be imported because the Approvers - UK group does not exist in the target environment. This is because we created that group in the test environment only.

To correct this, use migration to export the groups from your test environment and import them into your production environment.

1. In your TEST environment, click **Tools**, and then **Migration**.

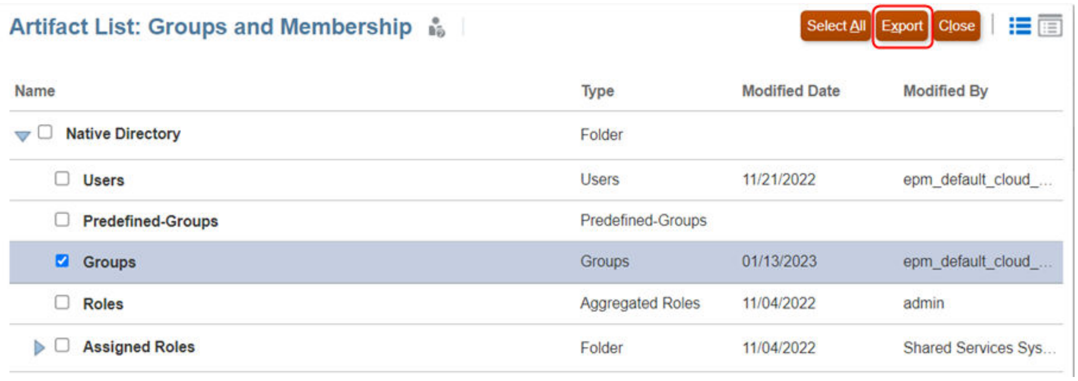


2. On the Categories tab, click **Groups and Membership**.



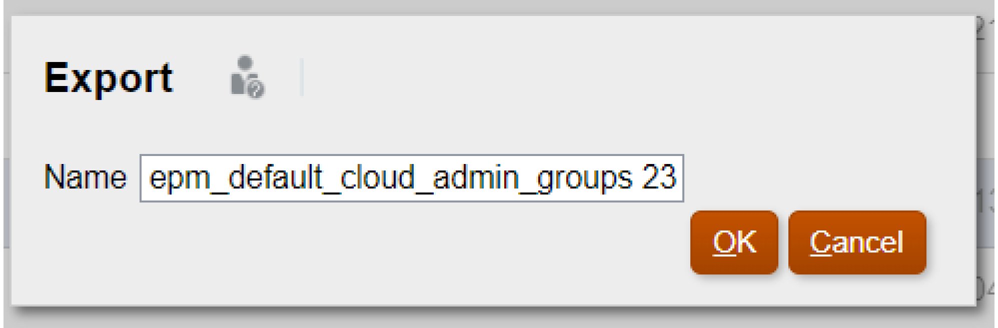
Name
□ Enterprise Data Management
□ Groups and Membership

3. Expand **Native Directory**, select **Groups**, and then click **Export**.



Name	Type	Modified Date	Modified By
□ Native Directory	Folder		
□ Users	Users	11/21/2022	epm_default_cloud_...
□ Predefined-Groups	Predefined-Groups		
☑ Groups	Groups	01/13/2023	epm_default_cloud_...
□ Roles	Aggregated Roles	11/04/2022	admin
□ Assigned Roles	Folder	11/04/2022	Shared Services Sys...

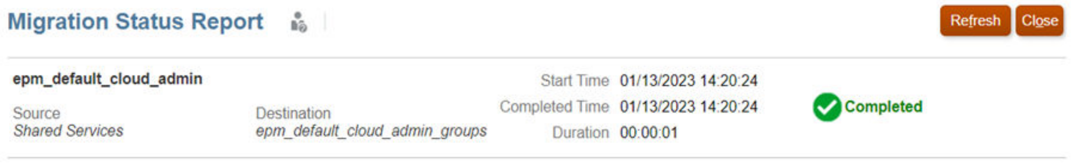
4. Enter a file name, and click **OK**.



Export

Name epm_default_cloud_admin_groups 23

5. When the export has completed, click **Close**.



Migration Status Report

epm_default_cloud_admin

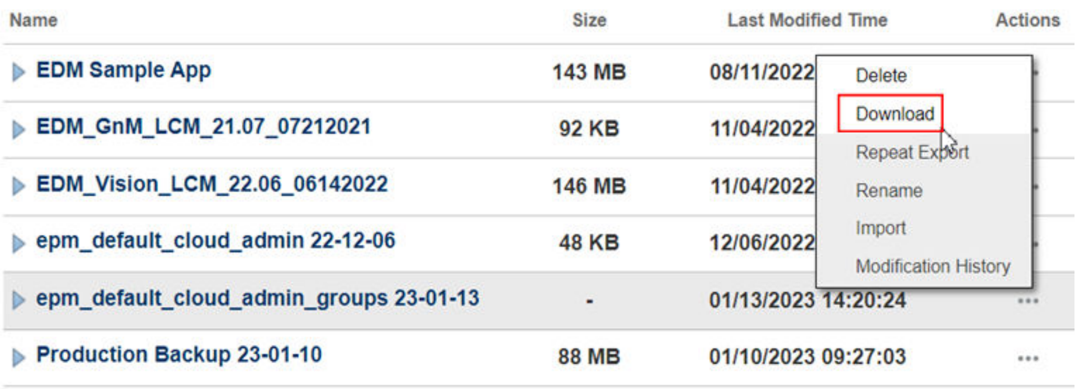
Source
Shared Services

Destination
epm_default_cloud_admin_groups

Start Time 01/13/2023 14:20:24
Completed Time 01/13/2023 14:20:24
Duration 00:00:01

✓ Completed

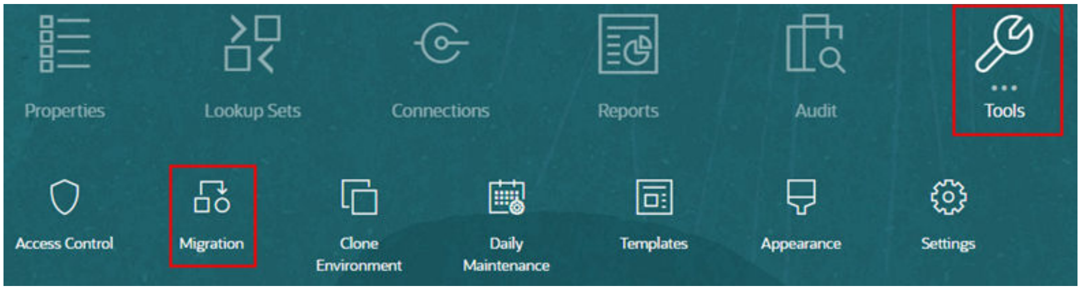
6. From **Snapshots**, locate the group export file, and then in the **Actions** column click ••• and then click **Download**.



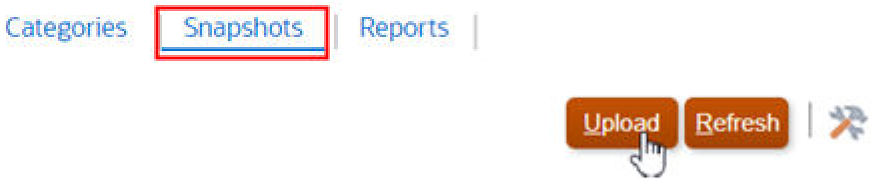
Name	Size	Last Modified Time	Actions
EDM Sample App	143 MB	08/11/2022	
EDM_GnM_LCM_21.07_07212021	92 KB	11/04/2022	
EDM_Vision_LCM_22.06_06142022	146 MB	11/04/2022	
epm_default_cloud_admin 22-12-06	48 KB	12/06/2022	
epm_default_cloud_admin_groups 23-01-13	-	01/13/2023 14:20:24	•••
Production Backup 23-01-10	88 MB	01/10/2023 09:27:03	•••

The groups snapshot is downloaded to your local machine.

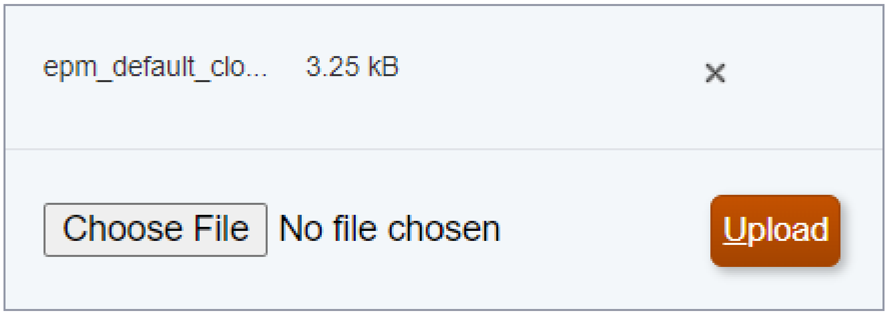
7. In your PRODUCTION environment, click **Tools**, and then **Migration**.



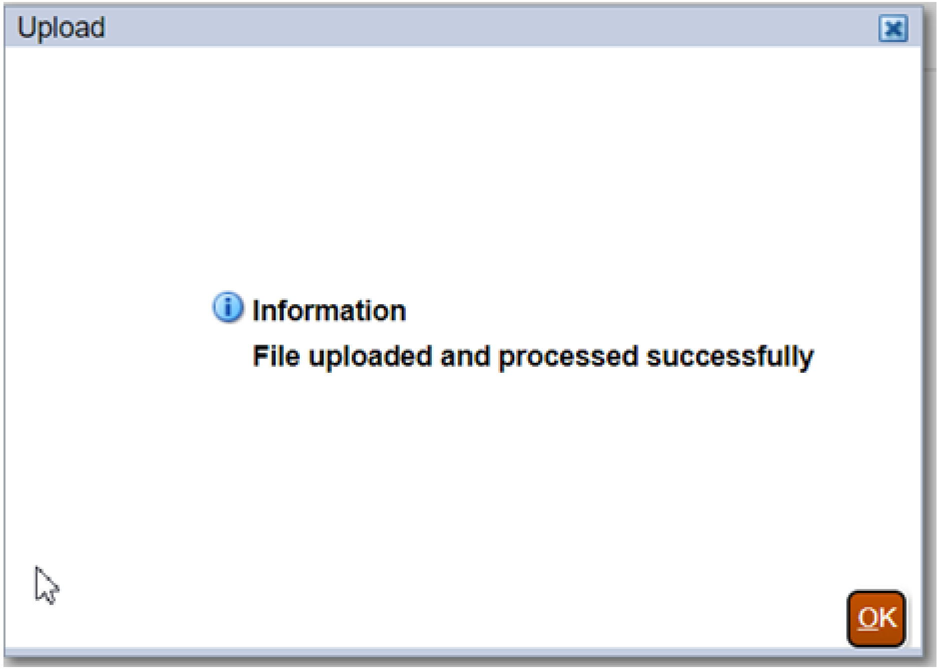
8. On the Snapshots tab, click **Upload**.



9. Click **Choose File**, select the file containing your groups, and then click **Upload**.



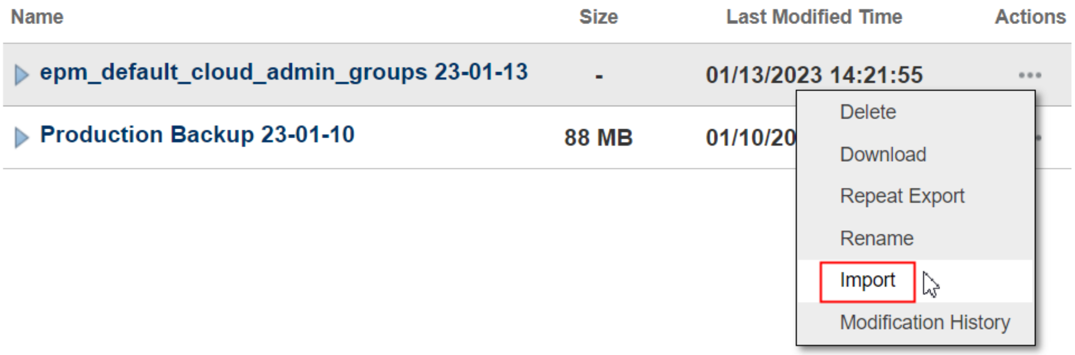
10. When the file has completed uploading, click **OK**.



Information

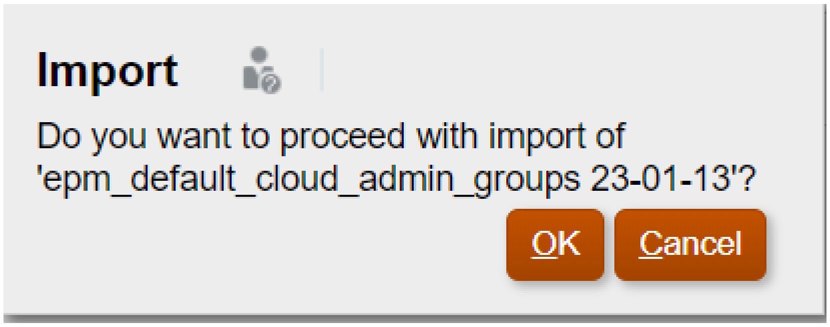
File uploaded and processed successfully

11. On the Snapshots tab, locate the file containing your groups and in the Actions column click ••• , and then select **Import**.



Name	Size	Last Modified Time	Actions
epm_default_cloud_admin_groups 23-01-13	-	01/13/2023 14:21:55	•••
Production Backup 23-01-10	88 MB	01/10/20	

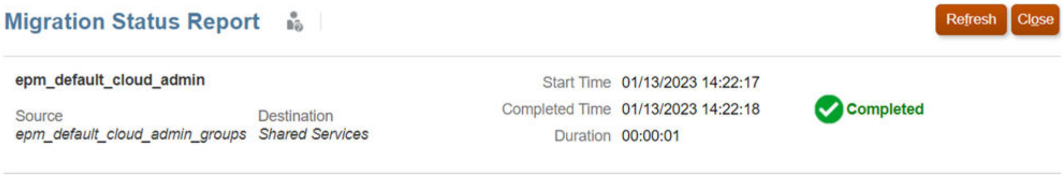
12. Click **OK** on the dialog to confirm the import.



Import

Do you want to proceed with import of 'epm_default_cloud_admin_groups 23-01-13'?

13. When the import has completed, click **Close**.



Migration Status Report

epm_default_cloud_admin

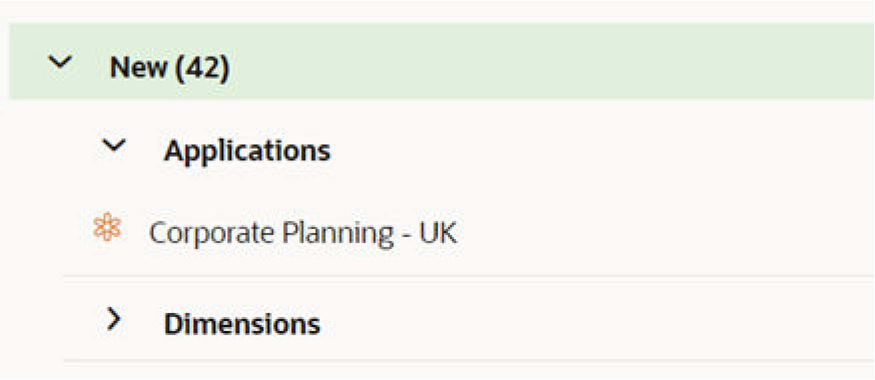
Source: *epm_default_cloud_admin_groups*
Destination: *Shared Services*

Start Time 01/13/2023 14:22:17
Completed Time 01/13/2023 14:22:18
Duration 00:00:01

Completed

Rerun the template preview and verify that the application permission and viewpoint subscription warnings are no longer displayed.

Application permission:



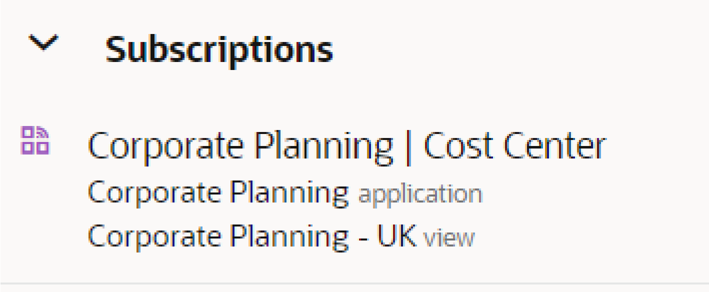
New (42)

Applications

Corporate Planning - UK

Dimensions

Viewpoint subscription:



Subscriptions

Corporate Planning | Cost Center
Corporate Planning application
Corporate Planning - UK view

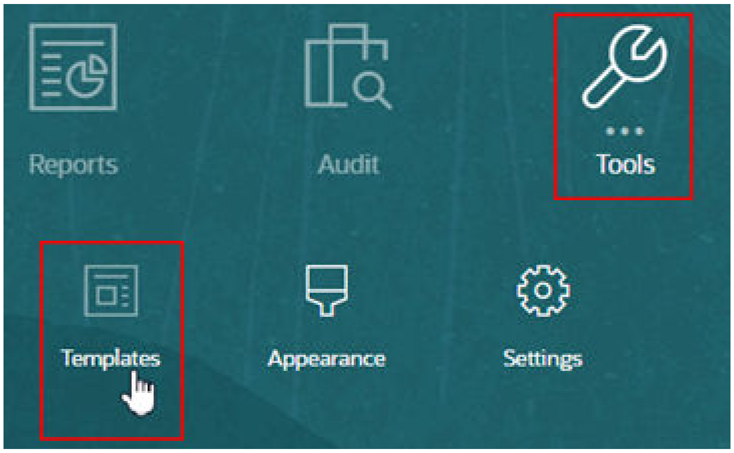
Correcting Missing Node Type Warnings

For the warning about the node type, the message indicates that the Cost Center - UK Rollup node type converter isn't able to be imported because the source node type doesn't exist in the target environment. This is because we created the Cost Center Rollup node type in the Corporate Planning application in our test environment but not in our production environment.

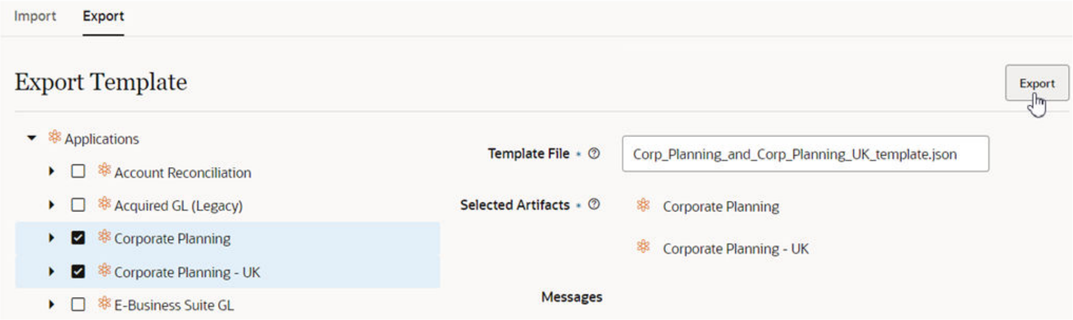
To correct this, we need to bring the Corporate Planning application from the test environment (where we added the source node type) to the production environment. We can use templates in two different ways to accomplish this:

- Export the Corporate Planning application to a separate template file in the test environment and import it first, then import the Corporate Planning - UK application.
- Export both Corporate Planning and Corporate Planning - UK to the same template file so that both are brought over together. This is the approach that we will take for this scenario.

1. In the TEST environment, click **Tools**, and then **Templates**.



2. On the **Export** tab:
   a. Expand **Applications**.
   b. Select the checkbox next to your both the **Corporate Planning** and **Corporate Planning - UK** applications.
   c. In **Template File**, enter a file name.
   d. Click **Export**.



- Applications
  - □ Account Reconciliation
  - □ Acquired GL (Legacy)
  - ☑ Corporate Planning
  - ☑ Corporate Planning - UK
  - □ E-Business Suite GL

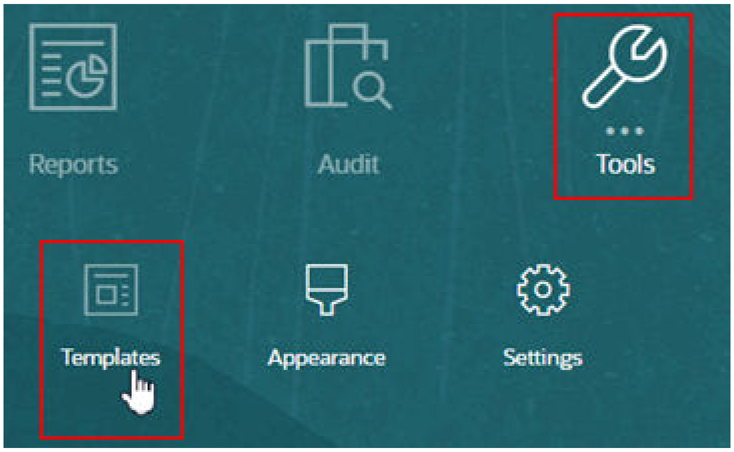
Template File • Corp_Planning_and_Corp_Planning_UK_template.json

Selected Artifacts • Corporate Planning
Corporate Planning - UK

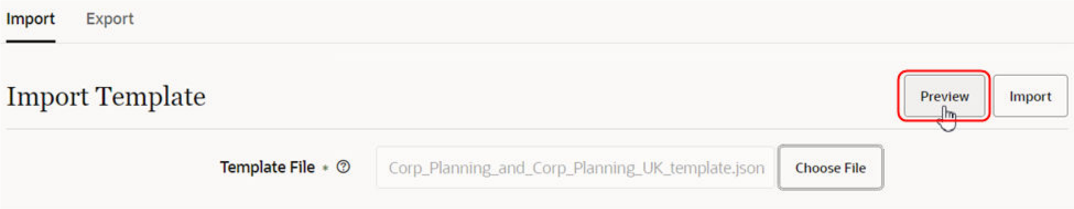
Messages

The applications are exported to a `.json` file and downloaded to your local system.

3. In the PRODUCTION environment, click **Tools**, and then **Templates**.

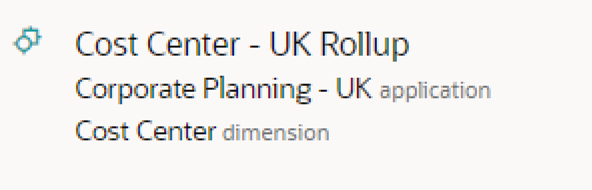


4. On the **Import** tab:
   a. Click **Choose File**.
   b. Navigate to the export file that was downloaded to your local system.
   c. Click **Preview**.



Template File • Corp_Planning_and_Corp_Planning_UK_template.json

5. Verify that the node type converter warning is no longer displayed.



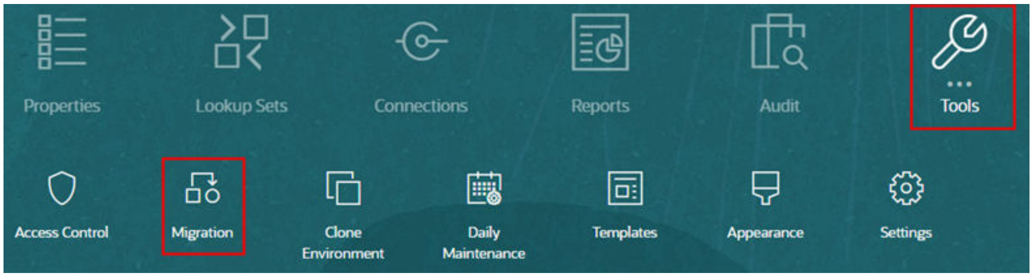
Cost Center - UK Rollup
Corporate Planning - UK application
Cost Center dimension

Step 6: Importing Applications from a Template

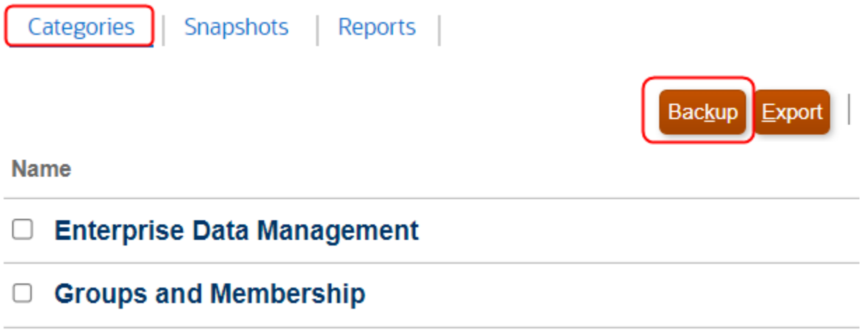
After you finish reviewing and correcting any warnings from the template import preview process, you can proceed with importing the both of the applications from the template file.

1. Before transferring applications over from templates, it is a best practice to take a backup snapshot of your production environment:

   a. In your PRODUCTION environment, click **Tools**, and then **Migration**.

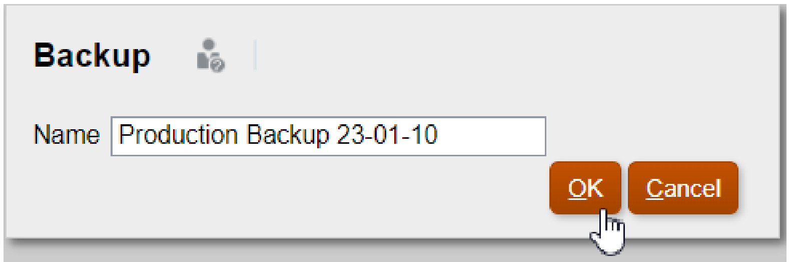


   b. From **Categories**, click **Backup** to create a backup snapshot of your current production environment.



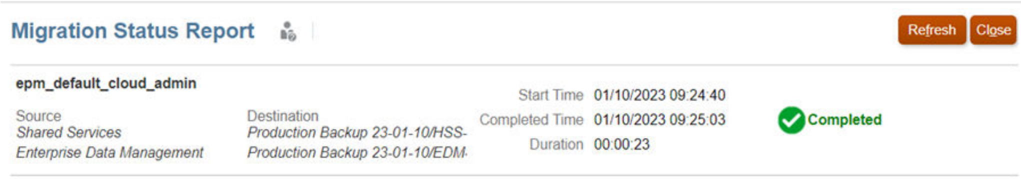
	Name
□	Enterprise Data Management
□	Groups and Membership

   c. Enter a name for the snapshot, and then click **OK**.



Name: Production Backup 23-01-10

   d. Click **Refresh** to update the status of the backup, and then click **Close** to close the window when the backup is complete.



Migration Status Report

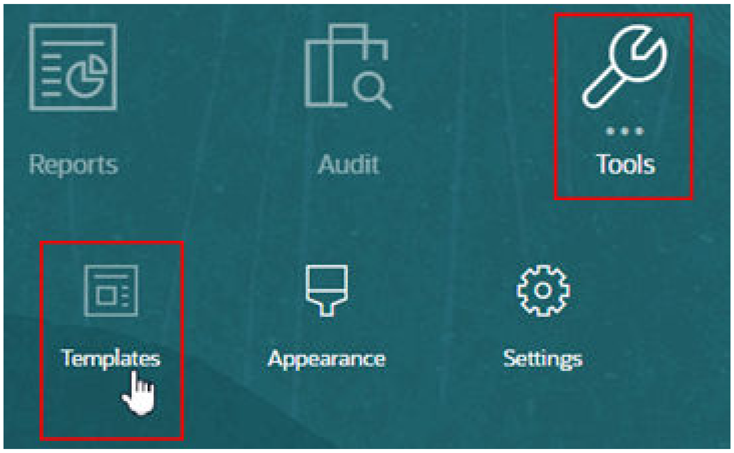
Refresh Close

epm_default_cloud_admin

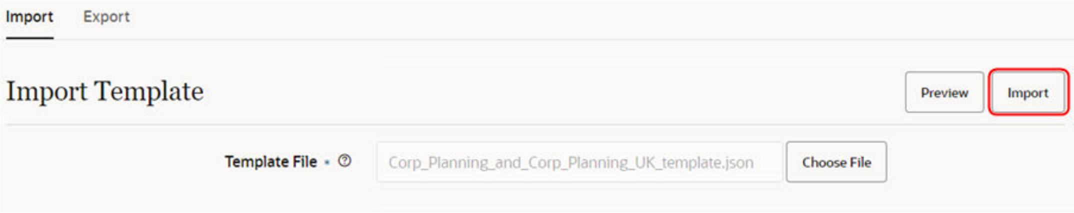
Source	Destination		
Shared Services Enterprise Data Management	Production Backup 23-01-10/HSS- Production Backup 23-01-10/EDM-	Start Time	01/10/2023 09:24:40
		Completed Time	01/10/2023 09:25:03
		Duration	00:00:23

✓ Completed

2. Next, import the applications using the template files that you created. In the PRODUCTION environment, click **Tools**, and then **Templates**.



3. On the **Import** tab:
   a. Click **Choose File**.
   b. Navigate to the export file that was downloaded to your local system.
   c. Click **Import**.



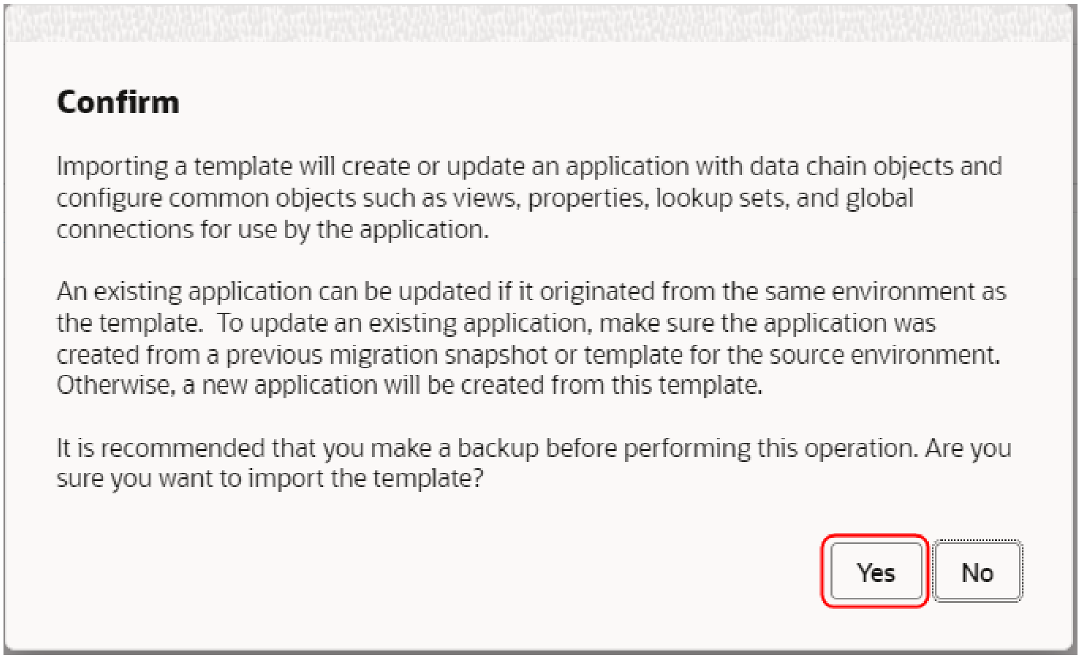
Import Export

Import Template

Preview Import

Template File * ⓘ Corp_Planning_and_Corp_Planning_UK_template.json Choose File

4. Review the confirmation message, and then click **Yes**.



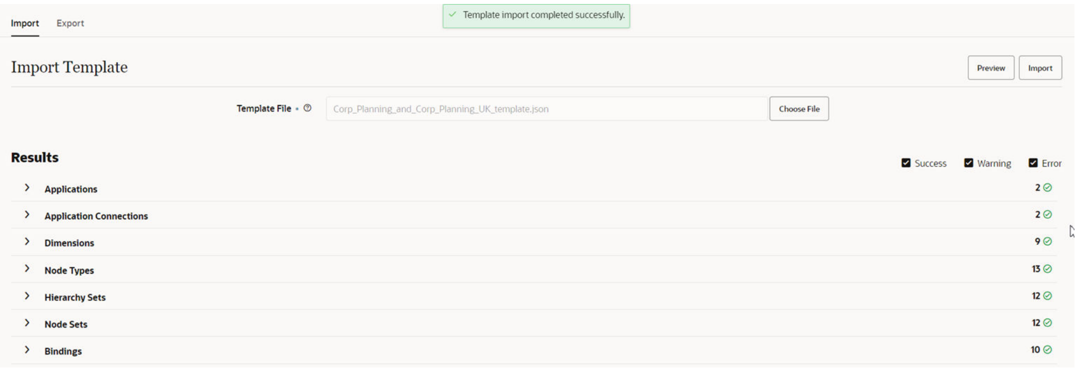
Confirm

Importing a template will create or update an application with data chain objects and configure common objects such as views, properties, lookup sets, and global connections for use by the application.

An existing application can be updated if it originated from the same environment as the template. To update an existing application, make sure the application was created from a previous migration snapshot or template for the source environment. Otherwise, a new application will be created from this template.

It is recommended that you make a backup before performing this operation. Are you sure you want to import the template?

5. Verify that the template was imported successfully.



✓ Template import completed successfully.

Import Template

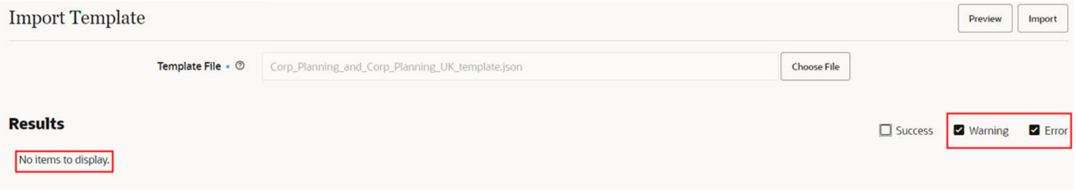
Template File • Corp_Planning_and_Corp_Planning_UK_template.json

Results

☑ Success ☑ Warning ☑ Error

Applications	2 ✓
Application Connections	2 ✓
Dimensions	9 ✓
Node Types	13 ✓
Hierarchy Sets	12 ✓
Node Sets	12 ✓
Bindings	10 ✓

6. Use the **Warning** and **Error** filters to locate any import warnings or errors.



Import Template

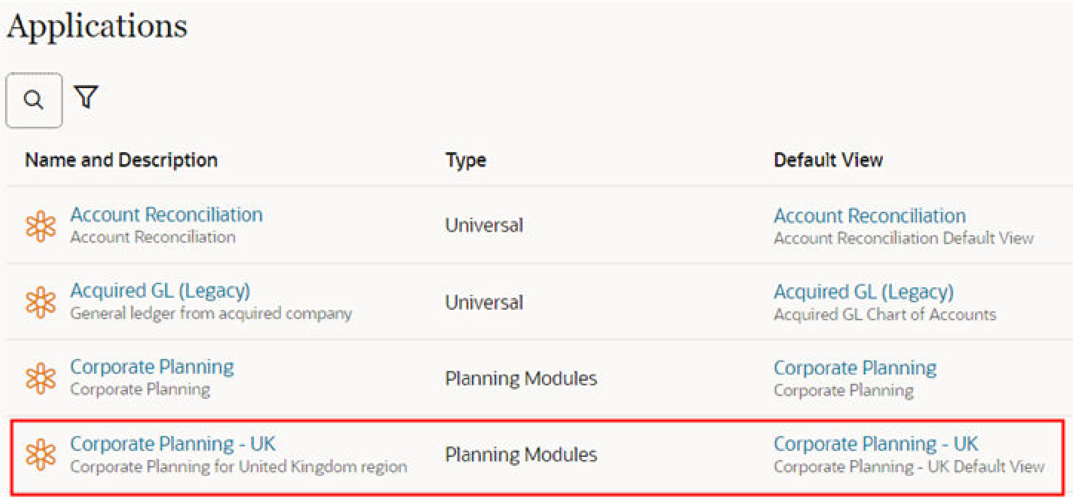
Template File • Corp_Planning_and_Corp_Planning_UK_template.json

Results

☐ Success ☑ Warning ☑ Error

No items to display.

7. Verify that the new application, new default and maintenance views, and new data chain objects exist in your production environment.



Applications

Name and Description	Type	Default View
Account Reconciliation Account Reconciliation	Universal	Account Reconciliation Account Reconciliation Default View
Acquired GL (Legacy) General ledger from acquired company	Universal	Acquired GL (Legacy) Acquired GL Chart of Accounts
Corporate Planning Corporate Planning	Planning Modules	Corporate Planning Corporate Planning
Corporate Planning - UK Corporate Planning for United Kingdom region	Planning Modules	Corporate Planning - UK Corporate Planning - UK Default View

A
Frequently Asked Questions

The following topics provide information for frequently asked questions about Oracle Fusion Cloud Enterprise Data Management. These questions can also include troubleshooting steps for commonly-encountered issues as well as tips and tricks for better performance.

Table A-1 FAQs in Oracle Enterprise Data Management

Topic	FAQs
Matching and Merging	- How are match scores calculated and how do I use them? - Should I use a code or a data source name in a load file? - Can I create a survivorship rule for a registered data source? - When creating match rules, is it better to add multiple criteria to a rule or to create separate rules? - I accidentally accepted a match that I did not mean to. Can I review my previously accepted matches and undo them?
Node Links	When are node links established between nodes?
Expressions	How can I improve the performance of my expressions when they contain a large number of sibling nodes?
Importing and Loading Data	- What is the difference between an import and a load? - When should I use an import versus a load? - What is the difference between Reset, Replace, and Merge modes for imports and loads, and which one should I use?
Property Values in List and Hierarchy Viewpoints	Why are my property values different for the same node between a list and a hierarchy viewpoint?

B
Best Practices

Best practices describe effective ways to use a feature. The following table provides links to the best practices mentioned in this guide.

Category	Best Practice For	See this Section
Configure Applications	Creating a new node type	Creating a Node Type
	Creating a new hierarchy set	Creating Hierarchy Sets
	Creating a new node set	Creating Node Sets
	Assigning cascading permissions	Permission Cascading
	Registering applications using application-specific wizards	Working with Universal Applications
Use Views	Creating alternate viewpoints	Best Practices
	Copying a viewpoint to create a new version	Versioning Best Practices
Govern Data	Using node type converters with subscriptions	Creating, Editing, and Validating Subscriptions
	Making mass changes using a request load file	Making Changes Interactively
Integrate	Validating a viewpoint before you export	Exporting Dimensions Exporting Planning and FreeForm Dimensions Exporting Oracle Financials Cloud General Ledger Dimensions Exporting Financial Consolidation and Close Dimensions Exporting E-Business Suite General Ledger Dimensions
	Transferring an application using a template	Working with Templates

C
Glossary

application

An Oracle Fusion Cloud Enterprise Data Management application is a related set of dimensions, segments, or value sets used to meet a specific set of operational, analytical, or reporting requirements.

application adapter

An application adapter encompasses the registration, bindings, validations, imports, and exports for a specific application type within Cloud EDM.

application type

An application type is a category that defines the product represented by the external application. For example, Planning and Universal are examples of an application type.

approval group

An approval group consists of the users or groups assigned to a policy as approvers. The user configuring the policy determines whether the users and groups can approve the request simultaneously or if they must approve in a specific order.

approval method

An approval method determines if approvals are serial or parallel. If serial, one approver at a time approves in a specified order. If parallel, all approvers can approve in any order and at the same time.

approval policy

An approval policy consists of the settings and groups configured to approve changes to an application, dimension, hierarchy set, or node type. A user with the Owner permission to the policy level (application, dimension, hierarchy set, or node type) can configure approval policies.

assignee

An assignee is the user who currently has the request assigned to them. Only one user can be assigned at a time.

binding

Bindings are created between dimensions and viewpoints in an Oracle Enterprise Data Management Cloud application to ensure that the bound data objects conform to an external application's requirements.

binding keys

Binding keys are created during application registration to control the binding rules and the import and export constraints on a binding for a dimension.

binding type

Specifies whether a binding is a node-type or a hierarchy-type binding. Node-type bindings represent viewpoints with a flat list of nodes that use list node sets. Hierarchy-type bindings represent viewpoints with a structured set of nodes that use hierarchy node sets.

bound data object

A bound data object is an object that is used by the external application. Changes made to a bound object directly impact the external application when the dimension is exported. When you import to an existing dimension, data in the dimension's bound data chain is affected.

commit policy

Commit policies provide a final review stage for a request during which all request items are reviewed and committed by a user specified in the commit policy in order to enforce a segregation of duties.

connection (application-specific)

Enable you to share data from one specific application in Oracle Enterprise Data Management Cloud to one specific external application. Application-specific connections cannot be reused in other applications.

connection (global)

Global connections enable you to share data with external applications using extracts. You can use a global connection in multiple extracts, but each extract can use only one global connection.

constraint

Constraints enable you to enforce specific data rules (such as name uniqueness) across node types and dimensions in an application.

contributor

A contributor is a user or group who has submitted, reassigned, approved, or added an attachment or comment to a request.

data

Data is a generic term used to refer to objects or information when it is not necessary to denote specific details. For example, when you import an application, data is imported.

data access

Data access determines the actions that a user with Participant permission can perform and the properties that they can see and update.

data chain

A data chain consists of the objects used to populate a dimension. The data chain may include one or more node types, one hierarchy set (if working with related nodes), one node set, and one viewpoint.

data manager (role)

As a Data Manager, you are a data steward or analyst. You load data and manage data quality. For example, you can import dimensions, perform comparisons, run validations, and resolve data issues.

default view

The default view is created during registration and is associated with an external application.

dimension

A dimension organizes business information. A dimension usually contains hierarchies of related nodes and can contain nodes from different node types. For example, an application with a Year dimension often includes nodes for each time period, such as quarters and months.

> **Note:**
>
> In some financial applications a dimension may be called a value set, chart of account, or chart of account segment.

dimension type

Dimension types are specific to the application type and define the dimension characteristics based on the external application type. For example, for Financial Consolidation and Close applications the dimension types are Account, Entity, Movement and Custom. The dimension type controls properties, validations, and other characteristics such as binding type (hierarchy or node type).

enterprise data

Enterprise data is organized and managed in Cloud EDM. It may include master data, domains, dimensions, hierarchies, reference data, application settings, metrics, measures, business classifications, and business taxonomies.

export

Exports are application-type specific mechanisms that move data from Cloud EDM to an external target system. Exports are controlled through the application registration, bindings, and binding keys.

export mapping

Exports data for one or more mapping keys.

expression

An expression consists of one or more lines of logic, called statements, that can be used to define custom business logic.

expression term

An expression term contains objects and operators that get evaluated to a value.

external application

An external application is an application such as Planning, residing entirely outside of Cloud EDM.

extract

Extracts enable you to retrieve data from a viewpoint within a dimension into a formatted file or a global connection. Extracts are configurable by users and can use bound and unbound data.

hierarchy

A hierarchy defines relationships between nodes. The hierarchy structure establishes familial relationships between nodes such as parent, child, sibling, descendant, and ancestor.

hierarchy set

A hierarchy set is a group of related nodes with a hierarchical structure which can have one or more top nodes.

implicitly shared nodes

When a node that has descendants occurs in more than one position in a hierarchy, after the initial occurrence of the node all of the other occurrences of that node in the hierarchy are considered implicitly shared nodes.

import

Import is a process in which the administrator imports data from an external application to populate a dimension's nodes, properties, and hierarchy relationships.

inheritance

Provides a way to define a default value for a property based on the position of a node within a hierarchy structure by retrieving the value from its closest ancestor that has a defined value.

invitee

An invitee is a user or group who is currently invited to approve a request.

list

A list is a group of nodes without a hierarchical structure. Lists can contain nodes of one or more types. For example, lists may be used to represent account segment value sets within a general ledger, a list of contacts, or accounts within a front office application.

literals

Literals are constant values that you manually enter in an expression.

location

The location is the full position of a node in a hierarchy. When a node is in multiple places in a hierarchy then each one will have a unique location. The location is the node combined with the full ancestor chain.

lock on commit

A property parameter that specifies that the property can only be updated on a newly added node. If this is enabled, after the request to add a node has been committed the value for the property is locked and can no longer be updated.

lookup set

Lookup sets are sets of key-value pairs that enable you to transform a set of values for properties into a different set of values.

mapping binding

A map binding allows node types to be mapped from dimensions in different source applications to a dimension in a target application.

mapping keys

Specifies the source node types mapped to target node types and defines a location name to export the mapping data.

master data

Master data is a single source of common business data used across multiple applications.

node

A node is an instance of a real-world business entity. Nodes can be used to represent an account, a cost center, a legal entity, a product, and so on. Nodes can be displayed in lists or hierarchies.

node set

A node set is a group of nodes for a particular business purpose. A node set uses a hierarchy set to manage a hierarchy of nodes or node types to manage list of nodes.

node set type

Determines whether a node set is a list-type or hierarchy-type node set. List-type node sets reference one or more node types. Hierarchy-type node sets reference a hierarchy set, which defines the node set's node types.

node type

A node type classifies and manages nodes based on common business characteristics. Node types are associated with dimensions; each node is assigned to a node type. Nodes are identified by a unique name and a description. For example, node name "1000", node description "Cash Account".

node type converter

Node type converters enable you to compare, locate, align, and drag and drop nodes of different node types across two different viewpoints.

notify policy

Notification policies enable application or dimension owners to notify specific users of completed requests submitted by other users.

object

An object is an element in an expression that represents a value.

operator

Operators perform logical comparisons in expressions.

owner (role)

As an owner, you are responsible for a business application or a subject area in your organization and you manage several functional areas in Cloud EDM.

permissions

Permissions secure access to applications, dimensions, data chain objects, and data. There are three permission levels: Owner, Data Manager, and Participant. For the Participant permission, you can also assign data access. See *data access*.

policy level

The policy level is the data object where an approval policy is set. Approval policies can be set at these levels: application, dimension, hierarchy set, and node type.

positional logic

Logic that describes the hierarchical position of a node, such as ancestor, parent, child, or bottom.

property

A property is a characteristic of an object. Nodes have identifying properties such as name and description. Nodes in hierarchies can have node and relationship properties. Node properties describe characteristics regarding the node itself, while relationship properties describe attributes of relationship in context.

property data type

The data storage format for the value of a property (for example, Boolean, date, string, or timestamp).

property default

Properties can be set up to have no default, a specified default or a derived default. This is managed in the property inspector and is setup during application registration based on the application type, dimension type, and property.

property level

Determines where property values are defined. For node-level properties, the defined property values apply to that node across all locations and viewpoints. For relationship-level properties, the defined property values are unique to specific parent-child relationships within a hierarchy set.

property parameters

Base property parameters are the initial configuration of a property from the application registration process. You can change the value of a base parameter for a specific application or node type by adding an override for that parameter at the application or node type level.

rationalization

Rationalization ensures data is represented consistently across multiple applications.

reference data

Reference data are permissible values that are often re-used, such as postal codes and state abbreviations.

registration

Registration is a process in which the administrator specifies the dimensions and properties to use from an external application to create an Cloud EDM application, default view, and data chain. Application registration also configures any other settings required by the application type. For example, Oracle Planning and Budgeting Cloud application registrations specify the plan types used by the application.

relationship

A connection between two nodes in Cloud EDM used for the parent-child organization of nodes in a hierarchy set.

request

A request is the mechanism to update data. Changes to data are organized into request items.

request item

A request item is a group of change actions for a specific node. A request item can contain one or more change actions.

request item action

Operations (such as add, insert, update, move, remove, or delete) that are performed on an item in a request. You can create a filter on a subscription or a policy for a request item action so that, for example, only specific actions generate a subscription request.

request stage

The request stage refers to the position of a request in the approval workflow. A request can be in one of these stages: Submit, Approve, or Closed.

request status

The request status is the status for the request: Draft, In Flight, Pushed Back, Recalled, Completed, or Rejected.

service administrator (role)

As a Service Administrator, you are a technical leader. You create and administer Enterprise Data Management service components. For example, you can manage user security, create and migrate applications, and set up maintenance and backups.

shared nodes

Nodes that exist under different positions within a hierarchy set or viewpoint.

statement

A statement in an expression is a discrete piece of information that controls the logic of an expression. There are three types of statements: If statements, Return statements, and Comments.

subscription

Subscriptions enable you to share data between viewpoints by subscribing a target viewpoint to a source viewpoint. When an update is made to the source viewpoint, a request is automatically generated to make the same change in the target viewpoint.

submitter

The submitter is the user who submitted a request. There is one submitter for a request. This user needs at least Submitter permission on all objects in the request.

template

Templates enable you to store application configurations in an offline file for use in other Oracle Enterprise Data Management Cloud environments.

unbound data object

An unbound data object is an object that is not used by the external application but instead can be used to model and evaluate potential changes to dimensions. Unbound objects are not updated by the import process. They may be indirectly affected by changes to other objects in their data chain.

user (role)

As a User, you are a business user or auditor. You view or modify data and may approve requests or audit changes made by other users. For example, you can browse data through views or submit and approve changes using requests.

validation

Validations preserve data integrity and enforce business logic in Oracle Enterprise Data Management Cloud.

validation level

The validation level is the data object to which a validation is applied in a custom validation. The validation level can be at the node type or hierarchy set.

view

Views give you a collection of lists and/or hierarchies for a particular context or activity, such as entity maintenance. Views consist of one or more viewpoints where you can update data. The default view is created when you register an application. A maintenance view is created for specific business needs.

viewpoint

Viewpoints provide a subset of nodes for you to work with. For example, viewpoints may represent different cost centers which require maintenance across applications such as financial applications and planning applications.

viewpoint type

Determines whether a viewpoint is a list-type or hierarchy-type viewpoint. The viewpoint type is determined by the node set that the viewpoint references.