

Oracle® Data Relationship Management Suite User's Guide



Release 11.2.x
F13692-06
October 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Data Relationship Management Suite User's Guide, Release 11.2.x

F13692-06

Copyright © 1999, 2023, Oracle and/or its affiliates.

Primary Author: EPM Information Development Team

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Documentation Accessibility

Documentation Feedback

1 Revision History

2 Getting Started

About Data Relationship Management Suite	2-1
Accessing Data Relationship Management	2-2
Changing Passwords	2-2
Security	2-3
Navigation Menu	2-3
Troubleshooting Tips	2-4

3 Managing Versions

Data Relationship Management Version Life Cycle	3-1
Version Type and Status	3-2
Creating Versions	3-3
Working with Version Variables	3-3
Creating Version Variables	3-3
Assigning a Version to a Variable	3-4
Deleting Version Variables	3-4
Copying Versions	3-4
Saving Versions	3-5
Loading and Unloading Versions	3-5
Creating As-Of Versions	3-6
Editing Version Properties	3-6
Assigning Validations to Versions	3-7
Assigning Nodes to Domains	3-8

Assigning Domains to Versions	3-9
Deleting Versions	3-9
Version Backup and Restore	3-10

4 Managing Hierarchies

Creating Hierarchies	4-1
Editing Hierarchy Properties	4-1
Enabling and Disabling Node Types	4-4
Opening Hierarchies	4-4
Hierarchy Groups	4-5
Assigning Controlled Node Access Groups	4-6
Assigning Validations	4-7
Deleting Hierarchies	4-7

5 Working with Nodes

About Nodes	5-1
Naming Nodes	5-2
Using Domains	5-2
Adding and Inserting Nodes	5-3
ID Function	5-5
Modeling Nodes	5-5
Copying Node Properties	5-7
Moving and Ordering Nodes	5-7
Relaxed Move Option	5-8
Sorting Nodes	5-8
Custom Sort Order	5-8
Node Deletion Methods	5-9
Merging Nodes	5-10
Deleting Nodes	5-10
Removing Nodes	5-11
Annulling Nodes	5-11
Inactivating Nodes	5-12
Reactivating Nodes	5-12
Assigning Validations	5-13
Orphan Nodes	5-13
Navigating Nodes in Tree View	5-14
Viewing Relationships	5-14
Setting Options in Tree View	5-15
Finding Nodes	5-15

Node Security	5-17
Using Shared Nodes	5-18
Shared Node Properties	5-19
Enabling Shared Nodes	5-19
Adding and Inserting Shared Nodes	5-20
Removing and Deleting Shared Nodes	5-21
Renaming Shared Nodes	5-21
Converting a Shared Node to a Primary Node	5-21
Displaying Shared Nodes	5-22
Shared Nodes and Domains	5-22

6 Managing Properties

Property Categories	6-1
System Properties	6-2
Stats Properties	6-2
Property Value Scope and Origin	6-3
Property Status	6-5
Editing Property Values	6-5
Filtering Properties	6-6

7 Validating Data

Validating Versions	7-1
Validating Hierarchies	7-2
Validating Nodes	7-2
Working With Validation Results	7-2

8 Governance Workflows

Workflow Models	8-1
Working with Requests	8-2
Request Lifecycle	8-9
Creating Requests	8-9
Calculating Request Item Properties	8-11
Validating Request Items	8-11
Submitting Requests	8-12
Claiming Requests	8-12
Unclaiming Requests	8-12
Approving Requests	8-13
Enriching Requests	8-13
Pushing Back Requests	8-14

Escalating Requests	8-14
Rejecting Requests	8-15
Handling Overdue Requests	8-15
Skipping a Workflow Stage	8-15
Promoting Requests	8-16
Splitting Requests	8-16
Committing Requests	8-16
Navigating the Worklist	8-16
Notifications	8-18

9 Using Queries

Query Types	9-1
Operator Descriptions	9-1
Creating Queries	9-3
Working with Query Results	9-6
Managing Queries	9-7

10 Comparing Hierarchies

Compare Types	10-1
Creating Compares	10-1
Working with Compare Results	10-4
Managing Compares	10-6

11 Working with Action Scripts

Action Script Parameters	11-1
Loading Action Scripts From Flat Files	11-9
Loading Action Scripts From a Transaction Log	11-11
Loading Action Scripts From a Node Model	11-12

12 Working with Imports

Import Sections	12-1
Using Substitution Parameters	12-3
Creating Imports	12-3
Running, Copying, Moving, and Deleting Imports	12-11
Viewing Import Results	12-12

13 Blending Versions

Creating Blenders	13-1
Running, Copying, Moving, and Deleting Blenders	13-6
Viewing Blender Results	13-6

14 Working with Exports

Creating Exports	14-1
Using Substitution Parameters	14-3
Export Specific Columns	14-3
Hierarchy Export	14-5
Customization Tags	14-11
Hierarchy XML Export	14-12
Generation Export	14-15
Compare Export	14-20
Difference Export	14-25
EPM Architect Export	14-29
Version Export	14-32
Property List Export	14-37
Property Lookup Export	14-38
Transaction Log Export	14-41
Merge Log Export	14-45
Creating Export Books	14-48
Running, Copying, Moving, and Deleting Exports and Export Books	14-50

15 Job Management

16 Auditing Transactions, Jobs, and Requests

Auditing Transaction History	16-1
Data Actions in Transaction History	16-2
Auditing Requests	16-7
Viewing Job History	16-8

17 Using the Data Relationship Management Batch Client

Securing the Batch Client	17-1
Running the Batch Client	17-2
Command Line and Configuration File Syntax	17-3

Command Line Parameters and Configuration File Keys	17-7
Outputs and Result Codes	17-23

18 Using Data Relationship Management Analytics

Version Lineage	18-2
Using Filters	18-2
Change Management Dashboard	18-3
Viewing Change Metrics	18-4
Growth Analysis Dashboard	18-5
Viewing Growth Metrics	18-5
Request Monitoring Dashboard	18-6
Viewing Request Metrics	18-7
Model Analysis Dashboard	18-8
Viewing Model Metrics	18-9
User Role Assignment Report	18-12
Access Group Membership Report	18-13
Object Access Group Authorization Report	18-13
Hierarchy Access Group Assignment Report	18-14
Workflow Access Group Assignment Report	18-15
User Login Activity Report	18-15
Metadata Object Usage Report	18-16

A Dashboard Metrics

Change Management Dashboard	A-1
Growth Analysis Dashboard	A-2
Request Monitoring Dashboard	A-3
Model Analysis Dashboard	A-5

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

Revision History

The following topics have been updated in this release of the guide:

Topic	Change
EPM Architect Export	Added note that new dimensions that require dimension associations cannot be created via the Oracle Hyperion EPM Architect export.
Command Line Parameters and Configuration File Keys	Added additional information to the description for the <code>CultureName</code> key.
Property Status	Updated description for Remove Duplicates.
Editing Property Values	Updated description for Remove Duplicates option in step 3.
Hierarchy Export Compare Export Version Export	Added a Tip to the Database Columns Options bullet.
Notifications	Various updates to clarify and update notification behavior.
Hierarchy Export Generation Export Compare Export Difference Export Version Export Property List Export Property Lookup Export Transaction Log Export Merge Log Export	Clarified the behavior of the quoted strings option.
Troubleshooting Tips	Added new section "Troubleshooting Tips" to the Getting Started chapter. Added workaround information about pasting into fields.
Filtering Properties	Added a note that filtering of node type properties applies to the Web client only.
Creating Queries	Added note to step 14 that the Name property is always displayed in query results regardless of what properties are selected to display.
Creating Imports	Updated the description for Strip Quoted Strings in step 6.
Import Sections	Add information on how to enable process and stripping of quoted strings to the Preserving Quoted Strings and Formatting section.

2

Getting Started

Related Topics

- [About Data Relationship Management Suite](#)
- [Accessing Data Relationship Management](#)
- [Security](#)
- [Navigation Menu](#)
- [Troubleshooting Tips](#)

About Data Relationship Management Suite

Oracle Data Relationship Management Suite consists of:

- Oracle Data Relationship Management
- Oracle Data Relationship Management Read Only Access
- Oracle Data Relationship Steward
- Oracle Data Relationship Governance
- Oracle Data Relationship Management Analytics
- Oracle Data Relationship Management for Oracle Hyperion Enterprise Planning Suite
- Oracle Data Relationship Management for Oracle Hyperion Financial Close Suite

About Data Relationship Management

Data Relationship Management functions as a hub where reporting structures are maintained, analyzed, and validated before moving throughout the enterprise.

Data Relationship Management does the following:

- Manages business entities, hierarchies, attributes, and mappings across multiple systems
- Validates data relationships and calculates attribute values using business rules
- Enforces referential integrity across all subscribing systems and hierarchies
- Enables users to create and manage alternate views
- Merges independent data sets from different sources and combines them into a master set
- Maintains historical versions for comparative reporting and analysis
- Tracks all hierarchy and attribute changes with a full-featured audit log
- Can serve as the main point-of-entry to update subscribing systems or be used after-the-fact for reconciliation and analysis

 **Caution:**

When you name metadata objects such as exports and compares, do not use a comma in the name.

Accessing Data Relationship Management

To start the Oracle Data Relationship Management client:

1. Select **Start**, then **Programs**, then **Oracle EPM System**, then **Data Relationship Management**, then **Web Client**.
2. Enter your user name and password.
User names and passwords are case-sensitive.
3. Select an application and click **Log On**.

Changing Passwords

To change a password:

1. From the Oracle Data Relationship Management Home page, select **Preferences**.
2. Click **Change My Password**.
3. Type the current password.
4. Type the new password.

 **Note:**

When a user is authenticated natively and the PasswordPolicyEnabled system preference is set to True, a password must contain three of the following elements:

- Uppercase letters
- Lowercase letters
- Numbers
- Special characters

 **Note:**

Otherwise, the password is not restricted unless by an external directory when the user is authenticated via Oracle Hyperion Shared Services.

5. Type the new password again.
6. Click **OK**.

Security

User access is determined by:

- User role assignments — For more information, see the *Oracle Data Relationship Management Administrator's Guide*.
- Version status
 - Working versions—Accessible to all users based on their security privileges
 - Submitted versions—Can be edited by the version owner or a Data Manager
 - Finalized versions—Read only to all user roles
 - Expired versions—Read only to the version owner or a Data Manager role.
- Node access group assignments — Node access groups are used to control user access to hierarchies and nodes. Users can be assigned to specific groups that have permissions to certain nodes within Oracle Data Relationship Management. For example, within the chart of accounts, a group of users may only need to maintain the Asset structure. If a user does not have access to any node in a hierarchy, the hierarchy is not displayed. If a user does not have access to any hierarchies in a version, the version is not displayed.
- Property category assignments — Access can be restricted to certain property categories for versions, hierarchies, and nodes. In most cases, property categories are based on a system basis.

Navigation Menu

Oracle Data Relationship Management features and content are organized into a set of web pages which are accessed from a centralized navigation menu on the Home page. Depending on the roles assigned to a user, a limited set of menu items may be available for selection. Descriptions for each group of web pages accessible from the Home page navigation menu are listed below.

- **Worklist**— The Worklist page is used to create, view, and search for requests using governance workflows to process proposed changes to data. When a request is created or opened, the request is displayed on a separate instance tab. Request level actions are performed on the instance tab for each request.
- **Browse**— The Browse pages are used to browse, search, and edit versions, hierarchies, nodes, and properties. Batch validations are run and their results are returned on the Browse pages. The administration of versions and hierarchies including creating, deleting and assigning access is also handled on these pages. When you browse a hierarchy for a version, each hierarchy opens on a separate instance tab. Hierarchy node level actions are performed on the instance tab for each hierarchy.
- **Query**—The Query pages are used to manage and run property queries and work with the results of a query process. The details and results of each query are displayed on a separate instance tab. The results of each query can be edited so you can make changes to hierarchy nodes and properties without navigating to the Browse pages.
- **Compare**—The Compare pages are used to manage and run compares and work with the results of the comparison process. The details and results of each compare are displayed on a separate instance tab. The results of each compare can be edited so you

can make changes to hierarchy nodes and properties without navigating to the Browse pages.

- **Script**—The Script pages are used to load and run action scripts and view the results of a script which was processed. You can have only one action script (and its results) open at a time. An action script that has been loaded from another source can be edited directly from the Script page.
- **Import**—The Import pages are used to manage and run imports and view the results of an import process. The details and results of each import are displayed on a separate instance tab.
- **Blend**—The Blend pages are used to manage and run blenders and view the results of a blender process. The details and results of each blender are displayed on a separate instance tab.
- **Export**—The Export pages are used to manage exports and books, run export processes, and view the results of an export process. The details and results of each export or book are displayed on a separate instance tab.
- **Audit**—The Audit pages are used to query and view history for transactions, jobs, and external requests. Only one type of history can be selected and viewed at a time.
- **Administer**—The Administer pages are used to administer metadata, workflow, and security for the Data Relationship Management application. The details of each metadata, workflow, or security object are displayed on a separate instance tab.

Troubleshooting Tips

Pasting Into Entry Fields

In some cases, content cannot be pasted from the clipboard by using right click and then **Paste**. To workaround this issue, use Ctrl-V or click **Edit** and then select **Paste** to paste content from the clipboard.

3

Managing Versions

Oracle Data Relationship Management organizes sets of related data into versions. A version is a single, independent set of data that can represent a period of time or a business scenario. Data within each version is categorized into hierarchies, nodes, and properties. All data-related operations require selection of a version.

Separate versions are typically used for the following purposes:

- To maintain historical copies of hierarchies, nodes, and properties for previous periods in time.
- To create new data or model changes to existing data without affecting other users or processes.
- To load data from different sources to be compared, rationalized, and blended.

Data Relationship Management Version Life Cycle

Most organizations use Oracle Data Relationship Management on a cyclical basis that matches their operational or reporting calendar. Within each calendar period, the use of Data Relationship Management follows a predictable pattern:

1. A new Data Relationship Management Working version is created as a copy of the Finalized version from the previous period. The new version may contain multiple hierarchies (for example, for the chart of accounts, the organization structure, and the product structure).
2. Changes are made to the Working version. Validations are automatically performed as users enter or modify hierarchy data.
3. If necessary, bulk changes to hierarchy data are performed using Action Scripts.
4. Near the reporting period deadline, the version status is changed to Submitted and changes are no longer permitted. Validations are performed to ensure the integrity of the data. Compares can be used to identify differences between the current version and the previous Finalized version.
5. When data integrity is assured, the version status is changed to Finalized and no further changes are permitted.
6. The version status for the previous reporting period may be changed from Finalized to Expired, and the version is stored for possible future use in historical analyses or as an audit record.
7. Exports are performed from the Finalized version to send hierarchy data to participating systems. After all exports are complete and have been loaded to the destination systems, all participating systems have consistent hierarchical data as a basis for the period end reporting process.

Existing organizational workflow constraints can be enforced by Data Relationship Management:

- Business rules might require that all new cost centers be approved by Corporate Treasury. In this case, a property can be added to indicate approval, and no nodes are

exported to other systems until the property is changed to approved. Corporate Treasury can be granted access to update only the indicator property. A property query can also be defined to identify indicator nodes.

- Business processes might require that all hierarchy updates be redirected to a dedicated group responsible for implementing such updates. Following review and approval, changes can be entered into a flat file for bulk loading via Action Scripts into Data Relationship Management. This automated approach can significantly reduce potential typing errors.
- More complex business processes that involve the coordination of multiple user inputs and approval before committing changes can be handled using change requests.

Other tasks that are performed on an irregular basis:

- New hierarchies can be established to support an expansion in scope of the participating systems. Hierarchies can be imported from an external source or can be directly created within Data Relationship Management.
- Hierarchies may need to be restructured to comply with changing business needs. Separate versions can be used to isolate these modifications from other production versions used for exporting to subscribing systems.
- Using the Blender feature, newly imported or restructured data in different versions can be combined into the same version with other existing production data.

Version Type and Status

Version information is available from the Browse task on the navigation bar. Each version has the following characteristics:

- Version name
- Version description
- Type
 - Normal—A version that can be edited and its status modified.
 - Baseline—A copy of the normal version at the time it was saved. This version type has a status of Expired.
 - As-Of—A historical view of a normal version based on a time period or transaction ID.
- Status
 - Working—Users can edit versions with this status.
 - Submitted—Only users who are the owner of a version or with a user role of Data Manager can edit versions with this status.
 - Finalized—No one can edit versions with this status.
 - Expired—No one can edit versions with this status.


 **Note:**

Only Data Manager users or version owners can see Expired versions.

- **Saved**—Specifies if the version is saved to the database or not. If not saved, the version is detached from the database and changes are made in memory only and will be lost if the application is stopped.
- **Load Status**—Specifies if the version is loaded into memory and ready to use or initialized but not loaded into memory.

Creating Versions

To create a new, empty version:

1. From **Browse**, click .
2. Enter a name and description for the version, and then click **OK**.

When the version tree refreshes, the new version is displayed in a detached (unsaved) state and you can start adding data to it.

Working with Version Variables

Version variables allow dynamic references to versions, which can change over time. Data Creator, Data Manager, and Application Administrator role users can create version variables to refer to different versions over time. User variables are managed by these users for their own personal use. Standard variables are managed by Data Manager users for public use by all users. System variables are managed by Application Administrators for system integration purposes.

A version may be assigned to multiple variables but any variable may only have one version assigned. A variable may also have no versions assigned but it is not recommended for variables that are saved with user metadata objects such as queries, compares, and exports.



In all areas of Oracle Data Relationship Management, a version variable may be used in the same manner as a version name. Using version variables, you can also:

- Quickly find a version in a large list of versions.
- Save version variables with queries, compares, imports, blenders, and exports.

Creating Version Variables

When creating a version variable, you must enter a label and select an object access level (user, standard, system, or custom group). The label must be unique across all versions and variables. A version variable may be created by adding it to the list of existing variables or by selecting a version and adding a new variable.

To create a version variable:

1. From **Browse**, click .
2. Click .

3. Enter the name and description for the version variable and select an object access.
4. Click **OK**.



Assigning a Version to a Variable

You can assign any Normal version to any accessible version variable. Baseline or As-Of versions cannot be assigned to a variable. Only one version can be assigned to a variable. If a variable currently has a version assigned, then that version is removed and the new version is assigned.

To assign a version to a variable:

1. From **Browse**, right-click a version to assign it to a variable and then select **Set Variables**.
2. Select a variable from the list.



To make multiple changes to version variable assignments:

1. From **Browse**, click  .
2. For any assigned version that you want to change, select the row. In the **Assigned Version** column, select the new version to assign to the variable.
3. In the **Action** column, click  .
4. Click **Close**.

Deleting Version Variables

You can delete any version variable for which you have management rights.

To delete a version variable:

1. From **Browse**, click  .
2. For any version variable that you want to delete, select the row of the variable you want to delete and click  in the **Action** column.
3. Click **Delete the Variable** to confirm the deletion.

Copying Versions

You can copy existing versions to create new versions. A version copy will create a normal, working version. Hierarchies, nodes, and properties are copied to the new version as well as access and validation assignments. Baseline versions, As-Of versions, and transaction history are not copied.

To copy a version:

1. From **Browse**, select a version.
2. Select **Versions**, then **Copy**.
3. Enter a name and description for the version.

4. **Optional:** If the Change Approval system preference is enabled in the version you are copying, you can clear it in the new version.
5. Click **OK**.

When the version tree refreshes, the new version is displayed in a detached (unsaved) state.

Saving Versions

To save a detached version:

1. From **Browse**, select a detached version.
2. Select **Versions**, then **Save**.

When the version tree refreshes, the version is displayed as saved.

Loading and Unloading Versions

Oracle Data Relationship Management works with versions of data loaded into memory on the application server. A version is loaded:

- At the time the version is created
- Explicitly from the application repository by user request
- Implicitly when data is requested from a saved version
- At the beginning of a data import process
- During a version restore process from a backup file

Loaded versions can utilize significant server resources (memory). Each loaded version increases the amount of memory used on the application server by Data Relationship Management. When a version is no longer being used, it is recommended to unload the version to enable other processes to use those resources.

To explicitly load a saved version from the repository:

1. From **Browse**, select a saved version.
2. Select **Versions**, then **Load**.

The Load Status of the version will change from Initialized to Saved after the version is loaded.



Note:

Unloading a version clears it from memory and deletes it if not saved to the database.

To unload a saved version:

1. From **Browse**, select a version.
2. Select **Versions**, then **Unload**.

When the version tree refreshes, the version is displayed with a Load Status of Initialized.

Creating As-Of Versions

An As-Of version is a snapshot of a normal version as of a given point in time. This type of version is created using the baseline version and applying changes from the transaction log up to the specified point in time. Only users who are the owner of a version or who have a user role of Data Manager can create an As-Of version.

You can specify an As-Of version using:

- A date and time
- A date offset from the current date
- A specific transaction ID

When creating an As-Of version using types Transaction Date/Time or Transaction Date Offset (days), the version name includes the date/time and time zone of the client browser that created the As-Of version.

Example: Departments - As-Of: 2014-08-13 10:55:03 (Mountain Daylight Time).

The date/time and time zone in the version name are static to easily identify the origin of the As-Of version. The date/time and time zone are not UTC values and are not recalculated for different client time zones.

When creating an As-Of version using type Transaction ID, the version name includes the specified transaction ID.

Example: Departments - As-Of:1000

To create an As-Of version:

1. From **Browser**, select a normal version.
2. Select **Versions**, then **Create As-Of Version**.
3. Select the As-Of Type: Transaction Date/Time, Transaction Date Offset (Days), or Transaction ID.
4. Perform an action:
 - For **Transaction Date/Time**, enter the date and time for the snapshot and click **OK**.
 - For **Transaction Date Offset (Days)**, enter the number of days offset from the current date and the time of day and then click **OK**.
 - For **Transaction ID**, enter the transaction ID and click **OK**.

When the version tree refreshes, the As-Of version is displayed as a child of its normal version.

Editing Version Properties

Version properties are defined at the version level and are the same for all nodes in all hierarchies. By definition, version properties are always global properties.


Table 3-1 Version Properties and Definitions

Version Property	Definition
Version ID	Unique, internal identifier for the version
Version Name	Name of the version
Version Description	Description for the version
Version Status	Status of the version: Working, Submitted, Finalized, Expired
Version Owner	User name of the user who created the version
Exact Copy Flag	Boolean indicating if the copied version is an exact copy of the source version
Copy Time	Date and time the version was copied
Legacy ID	Internal identifier of the version from which it was copied
Legacy Name	Name of the version from which it was copied
Lineage Version	Name of the version from which lineage should be built
Domains in Use	List of domains to which nodes in the version can be members

To edit version properties:

1. From **Browse**, select a version.
2. Select **Versions**, then **Properties**.
Version properties are displayed on the **Properties** tab of the sidebar.
3. Select a property category from the **Category** list.
4. Click the property row to edit and make changes.

 **Note:**

For simple data types, you can edit the property value within the editor. For more complex data types, click  to access the advanced editor for the property value.

5. Repeat steps 2 and 3 as needed to edit version properties.
6. Click **Save** in the property grid.

Assigning Validations to Versions

To assign validations to a version:

1. From **Browse**, select a version.
2. Select **Versions**, then **Assign Validations**.
Validations are displayed on the **Properties** tab of the sidebar.

3. Click a validation row to assign to the version.
4. From the **Value** list, select the type of validation:
 - **Real-time**—Validation is automatically run at the time of making a hierarchy change or saving property changes. This type of validation will prevent changes from being saved if they are not valid. Real-time validations can only be run at the node level.
 - **Batch**—Validation is run explicitly by a user from the version, hierarchy, or node level. Validation results are returned as a list of nodes and failure messages for each.
 - **Both**—Run validation in real-time and batch.

 **Note:**

Depending on how the validation is configured, only some of the types may be available.

 **Note:**

To remove a validation assignment, change the value to None or select **Remove Value** from the property menu.

5. Repeat steps 2 and 3 as needed to assign validations to the version.
6. Click **Save** in the property grid.

Assigning Nodes to Domains

You can assign nodes to domains at the version level.

To assign nodes to domains:

1. From **Browse**, select a version.
2. From **Versions** select **Domain Membership**.
3. You can select from the following options:
 - **Default Domain only**—Qualified nodes are assigned to the specified domain.
 - **Override Property only**—Qualified nodes are assigned to the domain specified in the override property.

 **Note:**

An override property is a designated global property with a domain name value used for domain name qualification and domain assignment.

- **Default Domain and Override Property**—If both options are specified, the default domain is used only when the override property value at the node is

blank. If the override property value is invalid, then the default domain is not used and an error occurs.

- **Inclusion Filter**—Use with any name qualification options; only qualified nodes in the filter results are assigned to the domain specified in **Override Property** or **Default Domain**
4. Click **OK**.



 **Note:**

After a node is made a member of a domain assigned to the version, you cannot remove the domain from the version unless all of the domain nodes are deleted from the version or the domain itself is deleted.

Assigning Domains to Versions

You can assign multiple domains to a version.

To assign a domain to a version:

1. From **Browse**, select a version.
2. Click .
3. Select **System** from the **Category** drop-down list and select the **Domains in Use** property.
4. Click  in the **Value** column.
5. From **Available**, select domains to assign to the version and move them to the **Selected** list.

 **Note:**

You can use the up and down arrows in the Selected list to determine the order in which the domains are displayed.

6. Click **OK**.

Deleting Versions

Saved and detached versions can be deleted from a Oracle Data Relationship Management application when they are no longer necessary. Deleting a detached version clears it from memory on the application server. For saved versions, the deletion process is handled in two phases:

1. The version is cleared from memory on the application server.
2. Database records for the deleted version are removed from the repository as a scheduled task.

Only normal and As-Of versions can be deleted. When you delete a normal version, all baseline and As-Of versions for the version are deleted as is the transaction history and requests for those versions.

To delete a version:

1. From **Browse**, select a version.
2. Select **Versions**, then **Delete**.
3. Click **OK** to confirm the deletion.

When the version tree refreshes, the version is no longer displayed.

Version Backup and Restore

Versions of data stored within a Oracle Data Relationship Management application may be temporarily archived to a file system or migrated to another application using a backup and restore procedure. One or more versions may be selected and included in a backup file which is written to an external connection defined by an administrator. The backup file may later be restored to the same or a different application. During the restore process, Data Relationship Management identifies whether any incompatibilities exist between the properties stored in the file and the property definitions to which they will be loaded in the target application. Restore From File Warning transactions are logged for property incompatibilities which may be encountered during the restore process.

Caution:

Version backup files can only be restored to a Data Relationship Management application of the same release level. For example, version backup files output from an 11.1.2.4.x application can only be restored in an 11.1.2.4.x application. Database backups of the Data Relationship Management repository should be used to permanently archive application data.

Note:

You cannot modify the binary backup file after creation.

To back up version to a file:

1. From **Browse**, select a version.
2. From **Versions**, select **Backup to File**.
3. From **Connection**, select a connection and enter a filename for the saved version file.
4. The version you selected should already be in the **Selected** list. To include more versions in the saved versions file, move versions from **Available** to **Selected**.
5. Click **OK**.

To restore versions from a file:

1. Log in to the application.
2. From **Browse**, select **Versions**, and then select **Restore from File**.
3. From **Connection**, select a connection.
4. Enter the name of the file, and then click **Open**.
5. Select one or more versions to load into the application.
6. Click **OK**.

4

Managing Hierarchies


A Oracle Data Relationship Management version can contain one or more hierarchies. A hierarchy can represent a business view, reporting structure, or a collection of related items. Hierarchies group and organize data into nodes, relationships, and properties.

Each hierarchy is assigned a top node and other nodes in the version can be added to the hierarchy by defining parent-child relationships between them. All nodes that are descendants of a hierarchy's top node are included within the hierarchy. A node can exist in multiple hierarchies and will always have the same descendants in every hierarchy to ensure synchronization across hierarchies. A node can have the same or different parents in each hierarchy.

Hierarchies provide the main interface for a user when working with Data Relationship Management. Examples of hierarchies include Balance Sheet, Line Of Business, Geographic, Legal, and Management Summary.

Creating Hierarchies

To create a hierarchy:

1. From **Browse**, select a version.
2. On the **Hierarchies** tab, click .
3. Enter a name and description.
4. **Optional:** Select a hierarchy group.

Note:

Hierarchy groups allow related hierarchies to be organized together for easy browsing and selection.

5. Select a top node for the new hierarchy by selecting one option:
 - **New Node**—Enter the name for the new node.
 - **Existing Node**—Select the hierarchy in which the node exists, and then select the node and click **OK**.
6. Provide a description for the top node.
7. Click **OK**.

When the hierarchy tree refreshes, the new hierarchy is displayed.

Editing Hierarchy Properties

Hierarchy properties are defined at the hierarchy level and are the same for all nodes in the hierarchy. By definition, hierarchy properties are always local properties.


Table 4-1 Hierarchy Properties

Hierarchy Property	Definition
Default Display Properties	Default for displaying nodes within the hierarchy
Default Paste Properties	If set, the Use Default Property List option is displayed in the Paste Properties dialog box. When this option is selected, only the properties specified here are pasted. If not set, the option is not displayed.
Default Match By	Default property by which to synchronize nodes for the current hierarchy.
Enable Shared Nodes	Determines if shared nodes are allowed in the hierarchy.
Hierarchy Node Type	Points to the node-level property that is used to determine if a specified node is a certain Node Type. Enables the use of node types for a hierarchy.
Hierarchy URL	URL hyperlink for directly accessing the hierarchy from an external resource
Hierarchy Description	Hierarchy description
Hierarchy Group	Displays the hierarchy group to which the hierarchy is assigned
Hierarchy ID	Unique, internal identifier for the hierarchy
Hierarchy Name	Hierarchy name
Hierarchy Owner	The user name of the hierarchy owner
Hierarchy Sort Order	Integer value that determines the order of display for hierarchies within a version or hierarchy group
ID Length Leaf Property	Used with the ID feature to determine the total length of the ID (excluding the prefix and including leading zeros). Points to the node-level property that is used to specify the length for the specified leaf node. If not set, then the length of the next sequential ID is used without leading zeros.
ID Length Limb Property	Used with the ID feature to determine the total length of the ID (excluding the prefix and including leading zeros). Points to the node-level property that is used to specify the length for the specified limb node. If not set, then the length of the next sequential ID is used without leading zeros.
Prefill Leaf Property	Points to the node-level property that is used to pre-fill node name when creating new leaf node
Prefill Limb Property	Points to the node-level property that is used to pre-fill node name when creating new limb node


Table 4-1 (Cont.) Hierarchy Properties

Hierarchy Property	Definition
Standard Hierarchy Sort	Points to a node-level property to determine default sort order within the hierarchy
Top Node	Name of the top node for the hierarchy
Top Node ID	Node ID of the top node
User Display Properties	Determines how hierarchy nodes display for the current user Note: For Anonymous profile users, this property is used for the current session only.
User Hierarchy Sort	Points to a node-level property to determine sort order within the hierarchy for the current user Note: For Anonymous profile users, this property is used for the current session only.
User Match By	Property by which to synchronize nodes in the selected hierarchy for the current user Note: For Anonymous profile users, this property is used for the current session only.
User Paste Properties	If set, the Use default Property List option is displayed in the Paste Properties dialog box. When this option is selected, only the properties specified in the hierarchy-level properties are pasted. If not set, the option is not displayed. Tip: If this property and Default Paste Properties are both set, the user property takes precedence over the default property. Note: For Anonymous profile users, this property is used for the current session only.

To edit hierarchy properties:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Select a hierarchy, and then click .
4. Select a property category from the **Category** list.
5. Click the property row to edit and make changes.

 **Note:**

For simple data types, edit the property value within the editor. For more complex data types, click  to access the advanced editor for the property value.

6. Repeat steps 3 and 4 as needed to edit hierarchy properties.
7. Click **Save** in the property grid.

Enabling and Disabling Node Types

Node types can be enabled or disabled for any hierarchy in a version using the `HierarchyNodeType` property. This property must point to a version, hierarchy, or node-level property whose value matches the name of a node type.

For more information, see "Managing Node Types" in *Oracle Data Relationship Management Administrator's Guide*.

To enable node types for a hierarchy:

1. Select a version and hierarchy.
2. From **Hierarchies**, select **Properties**.
3. In the property grid, select the **HierarchyNodeType** property.
4. Choose a node-level property which will return the name of a node type.
5. Click **Save** in the property grid.

To disable node types for a hierarchy:

1. Select a version and hierarchy.
2. From **Hierarchies**, select **Properties**.
3. In the property grid, select the **HierarchyNodeType** property.
4. From **Property**, select **Remove Value**.
5. Click **Save** in the property grid.

Opening Hierarchies

To open a hierarchy:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Right-click a hierarchy or hierarchy group, and then select **Open**.

Caution:

Opening a hierarchy group opens all of the hierarchies within that group.

Example 4-1 Auto Size All Columns

When a hierarchy window is first opened, the specified View By columns are displayed using an equivalent percentage of total width available. On the hierarchy window, the Auto Size All Columns toolbar button can be used to update the width of all displayed columns based on the maximum length property value in each column. The resize is based only on visible nodes on the current page. Any changes to column width,

whether done manually or via the Auto Size All Columns toolbar button, are persisted as long as the window is open.

Hierarchy Groups

Hierarchy groups allow related hierarchies to be organized together for easy browsing and selection. Hierarchies may be grouped using one or more hierarchy group properties. A hierarchy can only be associated with a single group for each hierarchy group property. The core Hierarchy Group property may be used for default grouping purposes. Additional hierarchy group properties may be added to handle alternate grouping requirements. When browsing hierarchies, use the Group By drop-down list to select a different hierarchy group property to use for grouping.

Creating Hierarchy Groups

To create a hierarchy group:

1. From **Browse**, select a version.
2. Select **Hierarchies**, then **New**, then **Hierarchy Group**.
3. Enter a name and description.
4. Select hierarchies from **Available** to include in the hierarchy group.
5. Click **OK**.

When the hierarchy tree refreshes, the new hierarchy group is displayed.

Note:

If you did not add at least one hierarchy to the new hierarchy group, the group does not display in the tree. At any time, you can assign hierarchies to an existing group.

Assigning Hierarchies to Hierarchy Groups

To assign a hierarchy to a hierarchy group:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Right-click a hierarchy and select **Assign to Group**.
4. Select hierarchies from **Available** to include in the hierarchy group.
5. Click **OK**.

When the hierarchy tree refreshes, the hierarchy is displayed within the assigned hierarchy group.

 **Note:**

You can set the value of a hierarchy group property in the property grid. You can also drag and drop a hierarchy to a different group to reassign it.

Assigning Controlled Properties

Global, inheriting properties must be assigned to a hierarchy that will control from where property values are inherited.

To assign controlled properties to a hierarchy:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Select a hierarchy that will control a global inheriting property.
4. Right-click the hierarchy, select **Assign Control**, and then **Properties**.
A list of global inheriting properties are displayed.
5. Select properties and move them from **Available** to **Selected**, and then click **OK**.

 **Note:**

Use the Category list to select a property category to ease navigation.

Assigning Controlled Node Access Groups

You use node access groups to control user access to hierarchies and nodes. Node access groups use inheritance to assign security to a node and its descendants. Global node access groups must be assigned to a hierarchy that will control from where access is inherited.

To assign a controlled node access group to a hierarchy:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Right-click a hierarchy, select **Assign Control**, and then **Node Access Groups**.
A list of global inheriting access groups are displayed.
4. Select groups and move them from **Available** to **Selected**, and then click **OK**.

 **Note:**

Use the Category list to select a user category to ease navigation.

Assigning Validations

Validations are tests to ensure that hierarchy rules are observed. They help enforce business rules.

To assign a validation to a hierarchy:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Right-click a hierarchy, and then select **Assign Validations**.

Available validations for the hierarchy are displayed on the **Properties** tab of the sidebar.

4. Click a validation row to assign to the hierarchy.
5. From the **Value** list, select the type of validation:
 - **Real-time**—Validation is automatically run at the time of making a hierarchy change or saving property changes. This type of validation will prevent changes from being saved if they are not valid. Real-time validations can only be run at the node level.
 - **Batch**—Validation is run explicitly by a user from the version, hierarchy, or node level. Validation results are returned as a list of nodes and failure messages for each.
 - **Both**—Run validation in real-time and batch.

 **Note:**

To remove a validation assignment, change the value to None or select **Remove Value** from the Property menu.

6. Repeat steps 4 and 5 as needed to assign validations to the hierarchy.
7. Click **Save** in the property grid.

Deleting Hierarchies

When you delete a hierarchy, the nodes within the hierarchy are not deleted. If the nodes do not exist in any other hierarchy within the version, then they become orphans. Click the Orphans tab to view the list of orphans.

To delete a hierarchy:

1. From **Browse**, select a version.
2. **Optional:** View a subset of hierarchies by selecting an option from the **Group By** drop-down list.
3. Right-click a hierarchy and select **Delete**.
4. Click **OK** to confirm the deletion.

When the version tree refreshes, the hierarchy is no longer displayed.

5

Working with Nodes

Related Topics

- [About Nodes](#)
- [Adding and Inserting Nodes](#)
- [Modeling Nodes](#)
- [Copying Node Properties](#)
- [Moving and Ordering Nodes](#)
- [Node Deletion Methods](#)
- [Assigning Validations](#)
- [Orphan Nodes](#)
- [Navigating Nodes in Tree View](#)
- [Finding Nodes](#)
- [Node Security](#)
- [Using Shared Nodes](#)

About Nodes

Master or reference data records used to describe, qualify, or summarize transactional data in a subscribing system are managed in Oracle Data Relationship Management as nodes. For example, within a hierarchy that represents an organizational structure, a node might represent a department or a cost center. Nodes in a version can have properties called global node properties. Nodes in a hierarchy can have hierarchy-specific properties called local node properties.

Within a version, a node may be a part of more than one hierarchy. A node can have many user-defined properties that store information about the node and control the use of the node within the information systems of an organization.

The following terms are used to define the position of a node and behavior of the node within a hierarchy:

Table 5-1 Node Terminology

Term	Definition
Leaf	A node that cannot have children
Limb	A node that can have children
Child	A node directly below another node (if B is directly below A, B is a child of A)
Parent	The node directly above another node (in the previous example, A is the parent of B)

Table 5-1 (Cont.) Node Terminology

Term	Definition
Descendant	A node below a specified node (including children and children of children)
Ancestor	All nodes between a node and the top of the hierarchy (including the parent, the parent of the parent, and so on)
Sibling	All nodes that share a parent node in a hierarchy
Orphan	A node not assigned to a hierarchy
Global Node	A node within a version.
Local Node	A node within a hierarchy.

Naming Nodes

Review these considerations when naming nodes.

- Do not use commas, the shared node delimiter, or the hierarchy delimiter in node names. Using commas in node names can cause problems when working with certain properties that are comma-delimited, such as Ancestor List, Child List, and client-created node lists.
- Do not use the characters defined in system preferences for the shared node delimiter and the hierarchy delimiter as they can cause unpredictable behavior.
- If you use the query wildcard characters asterisk (*) and underscore (_) in a node name, the node cannot be easily searched for unless a derived formula is used to search for them using the Pos function.

Using Domains

A node is uniquely identified by its name within a version and no two nodes can share the same name in the same version. By default, nodes with the same name from different systems can exist in different versions. This situation may become problematic when the nodes from these versions are combined into the same version if the intent is to keep managing them as separate nodes for referential integrity purposes.

A domain may be used to preserve integrity and ensure consistency of node names and descriptive properties across multiple versions for nodes of a common type. Examples of domains include Accounts, Entities, Departments, Products, Employees, and Projects. Domains are created by administrators and associated with versions by Data Manager role users or version owners. Nodes may be assigned to a domain in a particular version at the time of creation or later using the Domain Membership feature.

Domain nodes must follow the behavior specified for their domain. Nodes that are members of a domain cannot be renamed. Domains may be configured to:

- Use a prefix or suffix to qualify node names
- Restrict the deletion of nodes
- Restrict the changing of limb/leaf designation

Adding and Inserting Nodes

You can include a node in a hierarchy in two ways:

- Create a new node. Reviewing the considerations for naming nodes in [Naming Nodes](#).
- Insert an existing node not currently in the hierarchy

To create a node and add it to a hierarchy:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.

4. From **Nodes**, select **New**.
5. Select **Leaf Node** or **Limb Node**, depending on the type of node you are adding.
6. **Optional:** Select a domain from the drop-down list. The domain option in the **New Node** dialog box is displayed only if domains are assigned to the version. Some important considerations when selecting a domain:
 - If the node name entered does not exist, it is added as a member of the selected domain with the domain qualifier text appended to the name.
 - If the exact domain qualifier text is entered in the Name field, the qualifier text is not appended a second time.
 - If the node name entered exists in the domain but not in the current version, you can add the domain node to the version.
 - If the node name exists in the domain, the Domain Node Name and Inactive Date fields are populated with the values of the existing domain node.
 - If the node exists in the domain and in a version, the Copy Global Properties From option is available to copy the domain node's global properties from another version.
 - If a domain is not selected from the list and the node name entered includes text that qualifies it for a domain assigned to the version, the node cannot be added unless it is added to the domain at the same time, or was previously added to the domain in another version
7. Enter a unique name for the node. If the name entered matches an existing domain node, the name, description, and inactive date of the domain node are displayed for reference purposes.

 **Note:**

If you are adding a node to a part of the hierarchy tree that has repeated node names, you can click ID to automatically generate a node name. See [ID Function](#).

8. Enter a new description for the node.

 **Note:**


Descriptions are required for domain nodes, but are optional for non-domain nodes.

9. Select whether to add the node as a child or sibling of the selected node.
10. Click **OK**.

To insert an existing node into a hierarchy:


1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.

4. From **Nodes**, select **Insert**.
5. Select the location of the node:
 - Existing Hierarchy
 - Orphan List
6. Do one of the following:
 - If you selected **Existing Hierarchy** in step 5, do the following:
 - a. Select a version.
 - b. Select a hierarchy.
 - c. Select the node to insert by doing one of the following:
 - Enter text in **Find** and click  to locate the node.

 **Note:**

You can use the asterisk (*) as a wildcard character.

If multiple nodes are located, click  to find the next instance of the search item.

- Expand the Nodes tree to locate the node.
 - If you selected **Orphan List** in step 5, select a node from the list.
7. Select to insert the node as a child or sibling of the selected node.
 8. If you are inserting a node from a different version than the current version, do the following:
 - a. Select one of these options for copying node properties:
 - Do Not Copy Properties
 - Copy Global Properties

- Copy Local and Global Properties
 - b. If you want to include the node's descendants in the copy, select **Include Descendants**.
 - c. Select **Use These Settings as the Default** if you want to use the same settings for additional nodes that you copy.
9. Click **OK**.

ID Function

The ID function enables you to automatically generate node names using sequential numerical IDs and node prefixes. For example, a set of nodes can be created where each of them have a name like **Org000xxx**, where:

- **Org** is the node prefix and can be any desired character string.
- **000xxx** is the sequential ID and increments by 1 starting at 1.

These system preferences must be enabled for the ID function to work:

- AllowNextIDGeneration must be set to True.
- AllowNextIDKeyCreation determines the level of user that can create new ID keys.

To use the Next ID function, type a prefix in the node name field and click ID.

- If the prefix already exists in the system, then an ID value is returned that is one greater than the last ID previously generated.
- If the prefix has not been used before, then an ID value of 1 (with the appropriate number of leading zeros) is returned.

The total length of the ID (excluding the prefix and including any leading zeros as necessary) is determined as follows:

- The hierarchy properties **IDLengthLimbProp** and **IDLengthLeafProp** can be set to point to node-level properties that specify the desired ID length for the specified node.
- If the **IDLengthLimbProp** and **IDLengthLeafProp** properties are not set, then the length of the next sequential ID is used without leading zeros.

Modeling Nodes

You can model an existing node to create a new node with the same properties and in the same hierarchies as the node being modeled.


To create a node by modeling a node:




1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.
The top node of the hierarchy is displayed in a node tree.
4. Select a node to model.
5. On the **Shortcuts** menu, select **Script** and then select **Model After**.

The Source tab is populated with information from the node you selected.

6. On the **Properties** tab, from **Copy Properties**, select the properties to be copied from the node.
 - **None**—No properties are copied.
 - **Overridden**—Only overridden property values are copied.
 - **Selected**—Only the properties you select are copied.

 **Note:**

For **Selected**, click  to display a property selector from which to select properties.

- **All**—all properties are copied.
7. Do the following:
 - Select the **Global Properties** tab to view the Global property values that will be included in the Action Script based on your selection in Step 4.
 - Select the **Local Properties** tab and select a hierarchy to view the local property values that will be included from each hierarchy based on your selection in Step 4. Make changes as necessary.
 8. On the **Target** tab, do the following:
 - Select the target version for the new node.
 - Enter a name for the new node.
 - Enter a description for the new node.
 9. Click **Load**.
 10. When the load completes, review the script actions and parameter values. You can:
 - Click  to modify a row.
 - Select **Script**, then **Substitute Versions** to change a version. Select the version to be replaced and the version to replace it with and click **OK**.
 - Click a column header to sort rows alphabetically.
 - Click  to filter one or more columns.
 - Click  to switch between displaying properties by label or by name.
 11. Select the check box in the **Process** column next to each row to be run.

 **Note:**

From the Script menu, use Select All to process all actions. You can select a subset of actions from a script and only process those actions.

12. Click  to run the Action Script.

The results of each script action are displayed in the Status and Result Value columns. The script and the results can be saved to a file using the Download option on the Script menu.

Copying Node Properties

You can copy properties from one node to another, including copying and pasting properties to and from nodes in different versions.

To copy node properties:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.

4. Select a node from which to copy properties.
5. From **Nodes**, select **Copy Properties**.
6. Locate and select the node to which you want to paste the properties.

 **Note:**

If you are pasting properties to a node in a different version, open the other version and locate and select the node.

7. From **Nodes**, select **Paste Properties**.
8. In the Paste Properties dialog box, select the type of properties to copy:
 - Overridden Properties Only
 - All Properties
 - Selected Properties—If you select this option, you will select the properties to copy.
9. Click **OK**.

Moving and Ordering Nodes

You can move a node within a hierarchy or to a different hierarchy:

- Moving a node within a hierarchy removes the node from its original parent and places it under the new parent.
- Moving a node to a different hierarchy is treated like a copy. The node is left unchanged in its original hierarchy. This function is identical to inserting an existing node.

To move a node:

1. Select the node to move.
2. Drag and drop the selected node to the new location.
3. If you moved the node under a different top node, select an action:
 - **Put as Child**—To insert the node as a child of the node.

- **Put as Sibling or Reorder**—To insert the node as a sibling.

Relaxed Move Option

To facilitate a move that requires a node to be removed from an alternate hierarchy, you can use the Relaxed Move option. If the System Preference `AllowRelaxedMove` is set to `True`, the Relaxed Move is allowed. If the system preference is set to `False` or does not exist, then the normal integrity constraints are enforced.

When moving a node, the Relaxed Move enables the new parent in the main hierarchy to take precedence over any conflicting parental relationships for this node in other hierarchies.

Sorting Nodes

In Oracle Data Relationship Management, nodes within a hierarchy are sorted according to the following default rules:

1. Limb nodes are sorted before leaf nodes.

**Note:**

This is optional and controlled by the `SortLimbsFirst` system preference.

2. Nodes are sorted by name in ascending order.

Custom Sort Order

Nodes can also be sorted in any explicit custom sort order. The sort order can be defined by any property that has a property data type of `Sort`.

To sort a hierarchy according to the value of a node-level sort order property, the hierarchy-level property `StandardHierSort` must be set to point to the appropriate node-level property that contains the sort order.

A custom sort order can also be used that applies only to a specific Oracle Data Relationship Management user ID, enabling users to customize their display of Data Relationship Management data without impacting other users of the same hierarchy. This is controlled by the hierarchy-level property `UserHierSort`. If this property has been set to point to a node-level property that has a data type of `Sort Order`, then this sort order takes precedence over both the `StandardHierSort` property and the default sort order.

If there is no value assigned to the hierarchy-level property `StandardHierSort` nor to `UserHierSort`, then default sorting rules are used.

The value of a custom sort order property can be set in any of the following ways:

- Sort order values can be loaded into a node-level sort order property when data is initially imported into Data Relationship Management.
- Explicit values can be defined for the sort order property for any node. This property accepts an integer value.

- If a sort order property has been selected for a hierarchy's StandardHierSort or UserHierSort properties, use the Take/Put options or drag and drop to custom order nodes within a hierarchy.
- A node-level custom sort order can also be calculated using a derived Sort property.

Node Deletion Methods

The methods for deleting nodes are described below.

Table 5-2 Node Deletion Methods

Action	Result
Destroy	Deletes a node from the version and overrides any required merge logic if enabled Note: The Destroy option is only available from the Orphans list. Note: If the node is a domain node whose domain prevents deletion, you cannot delete the node.
Delete	Deletes the node from all parents in all hierarchies. Note: If the node is a domain node whose domain prevents deletion, you cannot delete the node.
Delete & Merge	Deletes the node from all parents in all hierarchies and allows you to select a node to use in place of the deleted node. Note: This option is only available if the UseMerge system preference is enabled. Note: If the node is a domain node whose domain prevents deletion, you cannot delete the node.
Annul	Removes the node from all hierarchies where it has the same parent, and also deletes the node if it is then an orphan <ul style="list-style-type: none"> • For example, if a node has the same parent in all hierarchies in which it exists, then Annul removes the node from all hierarchies and then delete it because it is an orphan. • If the node does not have the same parent in all hierarchies, the Annul produces the same result as Remove. Note: If the node is a domain node whose domain prevents deletion, you cannot delete the node.
Annul All Below	Removes all the children of the selected node. If any of the children become orphans, they are also deleted. Note: If the node is a domain node whose domain prevents deletion, you cannot delete the node.
Remove	Removes the node from the current parent but not from other parents in other hierarchies. The node is not deleted from the version and is available for reinsertion.

Table 5-2 (Cont.) Node Deletion Methods

Action	Result
Inactivate	Flags the node as inactive so that it can be filtered from an export. When a domain node is inactivated in a version, you can provide a domain inactive date value. The default domain inactive date is the current date.
Inactivate & Merge	Flags the node as inactive and allows you to select a node to use in place of the inactivated node. Note: This option is only available if the UseMerge system preference is enabled.

Merging Nodes

Subscribing systems supported by Oracle Data Relationship Management may have transactions that refer to a deleted node. When you delete or inactivate nodes, you can specify other nodes in Data Relationship Management to use in place of a deleted or inactive node to handle this situation. The merge list can be used to maintain and provide these node relationships to an external process to map transaction data appropriately using the Merge Log export type.

Note:

The merge options (**Delete & Merge** and **Inactivate & Merge**) are available only if the UseMerge system preference is enabled. The merge options are not available when deleting or inactivating **shared** nodes

Deleting Nodes

To delete a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.

4. Select a node.
5. From **Nodes**, select **Delete**.

 **Note:**

If the UseMerge system preference is enabled, you can select **Delete & Merge**, then select a node to use in place of the deleted node. If the ReqMerge system preference is enabled, then you must specify a merge node.

6. Select **Delete Node** to confirm node deletion.

When the version tree refreshes, the node is no longer displayed.

Removing Nodes

Removing a node removes it only from the current parent. If the node has other parents, those relationships are not affected. If the node has no other parents, it becomes an orphan node.

To remove a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.
The top node of the hierarchy is displayed in a node tree.
4. Select a node.
5. From **Nodes**, select **Remove**.
6. Select **Remove Node** to confirm node removal.

When the version tree refreshes, the node is no longer displayed.

Annuling Nodes

When you annul a node, the node is removed from the parent, and, if it becomes an orphan, the orphan node is deleted. For example, if a node is annulled in one of several hierarchies where it exists, then the node is just removed from that hierarchy and other hierarchies where it had the same parent. If a node is annulled from its parent in the only hierarchy it exists in, then Annul results in the node being deleted.

You can use the Annul All Below command to annul multiple nodes. Annul All Below used on a node that has descendants results in the entire branch being removed from the hierarchy and all nodes that do not exist in another hierarchy being deleted from the version.

To annul a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.
The top node of the hierarchy is displayed in a node tree.
4. Select a node.
5. From **Nodes**, select an action:
 - **Annul**—To annul the selected node

- **Annul All Below**—To annul all nodes below the selected node

 **Note:**

This option is available only if you select a limb node.

6. Select **Annul Node**.

When the version tree refreshes, the node is no longer displayed.

Inactivating Nodes

You can inactivate leaf nodes or limb nodes that have no active children.

The Inactivate and InactiveChanges system preferences control which users can inactivate a node and make changes to inactive nodes.

To inactivate a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.


4. Select a node.
5. From **Nodes**, select **Inactivate**.

 **Note:**

If the UseMerge system preference is enabled, you can select **Inactivate & Merge**, then select a node to use in place of the inactivated node.

6. **Optional:** Enter a **Domain Inactive Date** for the node and click **OK**.

When the version tree refreshes, the node is grayed out and displays an inactive

status icon  .

Reactivating Nodes

To reactivate a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.

The top node of the hierarchy is displayed in a node tree.

4. Select an inactive node.
5. From **Nodes**, select **Reactivate**.

When the version tree refreshes, the node is enabled and the inactive icon is removed.

Assigning Validations

To assign a validation to a node:

1. From **Browse**, select a version.
2. Select a hierarchy.
3. From **Hierarchies**, select **Open**.
The top node of the hierarchy is displayed in a node tree.
4. Select a node.
5. From **Nodes**, select **Assign Validations**.
6. On the **Properties** tab, select a validation and change its value from **None** to **Batch** or **RealTime**.
7. Click **Save**.


Orphan Nodes

An orphan node is defined as a node that exists in a version but is not assigned to a hierarchy within the version. Orphans may exist pending a cleanup to completely remove them, or they may be awaiting reassignment to hierarchies in the version.

Although they are not associated with any hierarchy, orphan nodes can have global relationships to other nodes in the version. It is possible for an orphan node to have descendants that are not orphans. To avoid this, review the list of orphan nodes regularly.

Viewing Orphan Nodes

To view orphan nodes:

1. Select the **Orphans** tab.
For each orphan node, the number of descendants is displayed.
2. Select a node.
3. Perform an action:
 - Select **Related** and then **Descendants** or **Children** to view a node's relationships.
 - Select **Node** and then **Properties** to view a node's properties.
 - Click  to search for an orphan node by name, description, or number of descendants.
 - Select **Node** and then **Take/Copy** to copy the selected node to the clipboard.
 - Select **Download** and then choose a format in which to download the list of orphan nodes.

Deleting Orphan Nodes

You can delete or destroy orphan nodes. If you select destroy, the node is deleted immediately regardless of any merge requirements. If you delete an orphan node, the node and all of its relationships to other nodes in the version are also deleted.

To delete orphan nodes:

1. Select the **Orphans** tab.
2. Select one or more nodes.
3. From **Node**, select **Delete** or **Destroy**.

 **Note:**

If the UseMerge system preference is enabled, you can select **Delete & Merge**, then select a node to use in place of the deleted node.




4. Confirm the Delete or Destroy command.

When the orphan list refreshes, the node is no longer displayed.

Navigating Nodes in Tree View

Use the following toolbar buttons to navigate in tree view.

Table 5-3 Toolbar buttons

Toolbar Button	Description
	Navigates to the previously selected node
	Navigates to the parent of the selected node.
	Expands the tree to display all nodes of the specified level below the selected node

Viewing Relationships

Use the options on the **Related** menu of the Tree toolbar to view different types of relationships for a selected node to other hierarchy nodes. Nodes matching the type of relationship selected are displayed on the **Relationships** tab.

- **Descendants**—View the descendants of the selected node.
- **Children**—View the children of the selected node.
- **Ancestors**—View the ancestors of the selected node.
- **Siblings**—View the siblings of the selected node.
- **Links**—View all hierarchies in the version in which the selected node exists.
- **Locate**—Locates the selected node in all open hierarchy tabs.
- **Match**—Locates nodes with a matching property value in all open hierarchy tabs.

 **Note:**

To perform the match, the value for the Default Match By hierarchy property must be populated or you can use a User Match By property by selecting Options, Match By, and then User Property.

Setting Options in Tree View

Use the items in the **Options** menu to determine how nodes are displayed and pasted into the tree:


- **Hide Inactive Nodes**—Hide inactive nodes from the tree view.
- **Put As**—Select one of these options to determine how copied nodes are pasted into the tree:
 - **Child of Target Node**—Copied node is pasted as a child of the selected node.
 - **Sibling of Target Node**—Copied node is pasted as a sibling of the selected node.
 - **Prompt for Destination**—You are prompted to select whether the copied node is pasted as a child or sibling of the selected node.
- **Put from Other Versions**—When pasting a node from another version, select one of the options for copying properties.
 - **Do Not Copy Properties**—Do not copy node properties.
 - **Copy Global Properties**—Copy and paste only global properties for the node.
 - **Copy Local and Global Properties**—Copy and paste both local and global properties for the node.
 - **Include Descendants**—Include the descendants of the copied node.
- **View From**—Displays the tree from either the Top Node or the Current Node.
- **View By**—Determines the columns that display in the tree view.
 - **Default**—Displays the node name and description columns.
 - **User Properties**—Select properties to add columns to the tree view.
- **Match By**—Determines how to match nodes.
 - **Default**—Matches nodes to the value in the Default Match By property for the hierarchy. If no value exists, the match does not return any results.
 - **User Property**—Select a property by which to match nodes.

Finding Nodes

You can search for nodes within any hierarchy in a selected version. Wildcard and exact match searches can be performed. You can optionally search for all instances of a node in the hierarchy by selecting to include implicitly shared nodes in the search. The implicit shared nodes find option is only available for searching within a single hierarchy, and not at the version level. Search results can be returned as a list of nodes or located in a hierarchy tree.

Orphan nodes are not included in these searches. You can use a global query to perform a search for all nodes within a version.

To find a node in a version:

1. From **Browse**, select a version.
2. Select the **Find Nodes** tab.
3. Click  and then select how you want to search:
 - **Find By**—Search for a node by name or description.
 - **Find In Domain**—Search for a node in a specified domain.

The search text is applicable to the **Domain Node Name** value only, not the Core.Abbrev.

 **Note:**

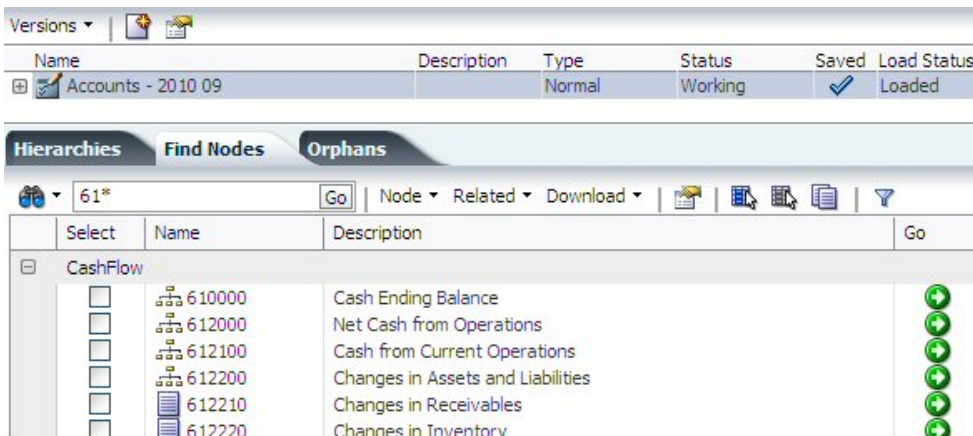
You can use the Find By and Find in Domain options together.

4. Enter text for which to search.







 **Note:**


You can use the asterisk (*) as a wildcard character. You can also use the backslash (\) as an escape character for exact match searches.

5. Click **Go**.




Name	Description	Type	Status	Saved	Load Status
Accounts - 2010 09		Normal	Working	<input checked="" type="checkbox"/>	Loaded

Select	Name	Description	Go
<input type="checkbox"/>	CashFlow		
<input type="checkbox"/>	610000	Cash Ending Balance	
<input type="checkbox"/>	612000	Net Cash from Operations	
<input type="checkbox"/>	612100	Cash from Current Operations	
<input type="checkbox"/>	612200	Changes in Assets and Liabilities	
<input type="checkbox"/>	612210	Changes in Receivables	
<input type="checkbox"/>	612220	Changes in Inventory	


6. Click  to navigate to a node within the hierarchy.

To find a node in a hierarchy:

1. From **Browse**, select a version.
2. Select a hierarchy, and then click .

 **Note:**

You can open multiple hierarchies in which to find nodes.


3. Click  and select how you want to search and return results:
 - **Include Implicitly Shared Nodes**—Select to include all implicitly shared nodes in the search. Implicitly shared nodes will only be displayed when this option is selected.
 - **Find By—Name or Description**
 - **Find in Domain**—Select a domain to search for the node in only that domain.
The search text is applicable to the **Domain Node Name** value only, not the Name/Core.Abbrev.
 - **Show Results—Locate in Tree or Show as List**
4. Enter text for which to search.


 **Note:**

You can use the asterisk (*) as a wildcard character.

5. Click **Go**.

 **Note:**

If you selected to show results in the hierarchy tree, you can move to the next result by clicking .

6. Select one of the found nodes and from **Related**, select **Locate**.
The node is displayed on the **Relationship** tab of the side bar. Above the node name is the hierarchy and version in which it is located.
7. Click  to navigate to a node within the hierarchy.

Node Security

Oracle Data Relationship Management users can be assigned different levels of access to nodes within and across hierarchies. The Limb and Leaf Access properties for a node indicate your level of access. Node security settings can only be assigned by a Data Relationship Management user with access management privileges. For more information, see the "User Permissions" section of the *Oracle Data Relationship Management Administrator's Guide*.

Using Shared Nodes

Shared nodes are multiple instances of a node within the same hierarchy. The global properties for shared nodes automatically reference the values of the primary node and can only be changed at the primary node. Local properties are not shared with the primary node and can be specific to each shared node.

Table 5-4 Types of Shared Nodes

Node	Description
Primary Node	The primary instance of a node in a hierarchy and the version. A shared node should not exist in a version where the primary does not exist.
Explicit Shared Node	Duplicate instances of a node that are explicitly created during manual maintenance or import processes. These nodes are defined with a special naming convention to distinguish them from the primary instance of a node in a hierarchy. Explicitly shared nodes, created during manual maintenance and import processes, share global property values with the primary node
Implicit Shared Node	Descendants of explicit shared nodes that are shared as a result of an ancestor being shared. These nodes are read-only and cannot be modified.

You can:

- Automate the alternate naming of the shared nodes
- Create system-level properties that map shared nodes
- Display the structure below a shared node
- Move between shared nodes and the primary node
- Remove and delete primary shared nodes
- Rename the primary shared node (thus renaming all the shared nodes)
- Find and display a shared node list to be used for navigation and analysis purposes

Only the following areas of Oracle Data Relationship Management work with Shared nodes:

- Browse Hierarchy in tree view
- View Shared nodes using the Related, Shared Nodes menu in tree view
- Export using the Include Implicit Shared Nodes option

Shared Node Properties

The following read-only, node properties are created automatically when a node is shared and provide information about primary and shared nodes

 **Note:**

The Shared Info property category must be manually assigned to users in order to see the shared node properties. For more information, see the "Managing Users" section of the *Oracle Data Relationship Management Administrator's Guide*.

Table 5-5 Shared Node Properties

Property	Description
Primary Node	Indicates whether a node has shared nodes that point to it
Shared Node	Indicates whether the node is a shared node
Primary Node Name	The primary name portion of the shared node name. For non-shared nodes, this property returns the node name in its entirety.
Primary Node Has Descendants	Indicates whether the primary node has structure below it. Only populated on shared nodes.
Primary Node Descendants	List of descendants of the primary node. Only populated on shared nodes where the primary node has structure below it.
Primary Node Ancestors	List of ancestors of the primary node. Only populated on shared nodes.
Shared Map	Associated nodes property that maps the shared and primary nodes. This property is populated for shared and primary nodes.
Missing Primary Node	Indicates whether the shared node points to a primary node that does not exist
Primary Not in Hierarchy	Indicates whether the primary node is not in the same hierarchy

Enabling Shared Nodes

To use shared nodes in an application, you must enable the `SharedNodeMaintenanceEnabled` system preference. For any hierarchy in which you want to share nodes, you must enable the `EnableSharedNodes` hierarchy property. These settings are disabled by default.

**Note:**

If shared nodes are enabled, the default shared node system preferences (delimiter, unique string, zero-pad length) should not be changed.

For information on configuring system preferences, see the *Oracle Data Relationship Management Administrator's Guide*.

Adding and Inserting Shared Nodes

You can use the add or insert menu options to create a shared node. You can create a shared node by doing a Take/Put across hierarchies, when the node already exists in the target hierarchy. You can create the same result by dragging/dropping across two hierarchies in a side-by-side compare result.

**Note:**

Before creating a shared node, make sure you have completed [Enabling Shared Nodes](#).

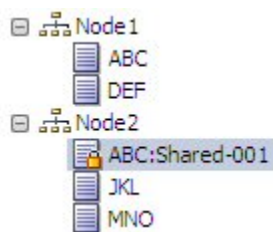
To add a shared node:

1. Select the node where you want to insert a shared node.
2. From **Nodes**, select **Add**.
3. If the shared node is a leaf, select **Leaf Node**.
4. Enter the name of the existing node that you want to add.
5. Select If Node already exists, Insert the Node.
6. For Add As, select Child or Sibling.
7. Click **OK**.

To insert a shared node:

1. Select the node where you want to insert a shared node.
2. From **Nodes**, select **Insert**.
3. For **Insert From**, make sure that **Existing Hierarchy** is selected.
4. Select a hierarchy.
5. From the **Nodes** list, select the node to share.
6. For **Insert As**, select **Child** or **Sibling**.
7. Click **OK**.

In this example, node ABC is a primary node under Node1 and a shared node under Node2.



Removing and Deleting Shared Nodes

When you remove a shared node, the system removes the shared node only and the primary node remains. If the primary node is removed and a shared node also exists, the shared node is removed and the primary is moved to the parent of the shared node that was removed. If multiple shared nodes exist, one of the shared nodes becomes the primary node.

When you delete a primary node from a version, the primary node and all of its explicit shared nodes are deleted.

To remove or delete a shared node:

1. Select a shared node.
2. From **Nodes** select **Delete** or **Remove**.
3. Select **Delete Node** or **Remove Node**.

When the hierarchy tree refreshes, the shared nodes are no longer displayed.

Renaming Shared Nodes

A rename operation can only take place on a primary node. When you rename a primary node, the system renames the primary node and the shared nodes.

To rename a shared node:

1. Select the primary node.
2. In the property grid, select the Name value field.
3. Change the name and click **Save**.


When the hierarchy refreshes, the primary node name and the shared node names are updated with the new name.

Converting a Shared Node to a Primary Node

After working with different hierarchies that have shared nodes the situation can arise where a hierarchy has a shared node in it but the primary node does not exist in the hierarchy. In this scenario, you can convert the shared node into the primary node in one operation.

To convert a shared node to a primary node:

1. Select the shared node.
2. From **Nodes**, select **Convert to Primary**.

When the hierarchy refreshes, the selected shared node is displayed with the primary node icon .

 **Note:**

If the primary node already exists in the hierarchy and you want the primary node to be in the location of the shared node, then you must manually swap the nodes

Displaying Shared Nodes

To display the primary, explicit, and implicit shared nodes for a node:

1. Select a node.
2. From **Related**, select **Shared Nodes**.

Primary and shared nodes are displayed on the **Relationships** tab.

Shared Nodes and Domains

It is important to understand the following information about shared nodes and domains:

- The Domain Membership menu option is not be available for shared nodes (explicit or implicit).
- If a primary node is added to a domain, its shared nodes reflect the same domain attributes.
- Children of primary nodes have their implicit nodes displayed as read-only and reflect the domain attributes of the primary node.
- When the Domain Membership process is run from the version level, shared nodes do not display as added to the domain, but display as an exception in the list, for example, Node XXX is a shared node and can't be directly assigned to a domain. Shared nodes inherit their domain attributes from their primary node

6

Managing Properties

Properties are attributes of a version, hierarchy, or node. Core properties manage attributes that are used for standard product functionality. Custom properties can be added by the application administrator to manage additional attributes that are necessary to support subscribing systems and business processes.

Properties are displayed for a version, hierarchy, or node on the Properties tab.

Property Categories

Version, hierarchy, and node properties are organized by category. Each property category represents a set of logically related properties such as those belonging to a particular external system (for example, Oracle Hyperion Planning or Oracle Essbase) or a specific functional area within the company. The application administrator can create as many property categories as needed. Oracle Data Relationship Management includes default property categories.

Table 6-1 Default Property Categories

Category	Description
System	Properties related to a node's identifying characteristics, such as ID, name and description. The only change that can be made to this category is assigning the read-only flag for individual users. Users with read-only access cannot edit values in the category, but can view them. Note: Properties cannot be assigned to this category.
Stats	Properties that provide statistical information about the node such as number of children, number of siblings, and so on This category is available for nodes only.
Validation	Validations assigned for the node; one property for each validation This category is available for nodes only.
Leaf Access	Node security groups and their leaf access levels for the node; one property for each group This category is available for nodes only.
Limb Access	Node security groups and their limb access levels for the node; one property for each group This category is available for nodes only.

You can have different levels of access to different property categories. Categories that can be viewed are available in the drop-down list. Categories that can be edited allow

modification of properties that support overridden values. The Validation, Leaf Access, and Limb Access categories are available only to users with the Data Manager role.

System Properties

System properties typically displayed at the node level are listed in the following table.

Table 6-2 System Properties

Property	Definition
Description	Description of node
Display String	Value of combined properties used for viewing nodes in a hierarchy. Controlled by the Default Display By and User Display By properties.
Domain Name	Name of the domain to which the node is a member
Domain Node Inactive Date	Specified date for which the domain node is declared inactive
Domain Node Name	Name of the node within its domain. This value does not include the domain qualifier
Domain Node Description	Most recent description defined for a domain node in any version where it exists
Leaf	<ul style="list-style-type: none"> • If True, node is a leaf node and cannot have children • If False, node is a limb node and may have children
Leaf Access	Indicates level of access to node for current user (only applies if node is leaf)
Limb Access	Indicates level of access to node for current user (only applies if node is limb)
Name	Unique name of node within a version
Node Approved	Boolean flag to indicate that node has been in a Finalized version
Node ID	Unique numeric identifier for the node generated by the system
Node URL	URL hyperlink for directly accessing the node from an external resource

Stats Properties

Statistical properties displayed at the node level are listed in the following table.

Table 6-3 Stats Properties

Property	Definition
# Children	Number of nodes, one level beneath
# Descendants	Total number of nodes beneath (at all levels)

Table 6-3 (Cont.) Stats Properties

Property	Definition
# Leaf Children	Number of child nodes that cannot have children
# Leaf Descendants	Number of descendants that cannot have children
# Leaf Siblings	Number of siblings that cannot have children
# Limb Children	Number of children that may have children
# Limb Descendants	Number of descendants that may have children
# Limb Siblings	Number of siblings that may have children
# Links	Number of other hierarchies in which the node exists
# Siblings	Number of nodes which have the same parent as this node
Added By	User who created the node
Added On	Date and time the node was created
Ancestor List	List of ancestor nodes from parent node up to the top node of a hierarchy
Last Changed By	User who last changed the node
Last Changed On	Date and time the node was last changed
Level	Numeric level of node within hierarchy where top node level = 1
Linked	Flag indicating whether the node exists in at least one other hierarchy
Links	List of all other hierarchies in which the node exists (excluding current)
Node Changed	True or False indicating whether the node was changed since creation of the version
References	List of all hierarchies in which the node exists (including current)

Property Value Scope and Origin

The property grid displays two icons to the left of each property. The first icon indicates the scope of the property and if it can inherit values. The second icon indicates the origin of the property value.

Table 6-4 Property Scope Icons


Property Scope	Description
	The property value is constant across all hierarchies within a version.

Table 6-4 (Cont.) Property Scope Icons










Property Scope	Description
Local 	The property value is specific to the hierarchy in which it exists.
Global Inherited 	The value of the property is global and inherited from an ancestor node in the controlling hierarchy for the property.
Local Inherited 	The value of the property is local and inherited from an ancestor node in the current hierarchy.

Table 6-5 Property Origin Values

Property Origin	Description
Default 	The value of the property is populated by a default value (as set by the application administrator.) Default values are used when: <ul style="list-style-type: none"> The property is non-inheriting and you have not entered a value. The property is inheriting and no ancestor value is found.
Inherited 	The value of the property is being supplied by an ancestor node. (The name of the ancestor node and the hierarchy from which the inherited value originates is displayed in the Status column.)
Overridden 	The value of the property was set by a user at this specific node.
Derived 	The value of the property is derived (calculated) from other property values.
Locked 	The value of the property has been locked for the current node and cannot be changed in any descendant node that inherits this property's value. If this property value is viewed in a descendant node, its status is indicated with the Static icon.
Static 	The value of the property has been inherited from an ancestor node, whose value has been locked: <ul style="list-style-type: none"> As a result, the value cannot be changed in the descendant node The name of the ancestor node and the hierarchy from which the inherited value originates is shown in the Status column of the Property window

Property Status

The Status column in the Property Grid provides the name of the ancestor node and the hierarchy in which an inherited value originates. When editing properties, this column also displays status messages that reflect how a value has been edited. The following table describes property statuses.

Table 6-6 Property Status

Property status	Definition
Clear All Below	A request has been made to Clear All Below for the selected property. Clears the value on all descendant nodes which have overridden their inherited values.)
Locked	The classification of the property has been changed to Locked.
Modified Value	The value has been modified.
Remove Duplicates	A request has been made to Remove Duplicates for the selected property. Removes any equivalent value from the descendants of the node.
Remove Value	The value has been removed; only applicable to Overridden property values.
Unlocked	The classification of the property has been changed to Unlocked and the value can be changed in descendant nodes that inherits the property.

Editing Property Values

Properties that are grayed out in the property editor are read only and cannot be modified. A property value cannot be edited if any of these conditions exist:

- The property has a derived value. For example, a property indicating the total number of child nodes is derived by Oracle Data Relationship Management and cannot be edited.
- The user does not have appropriate security rights to edit the value.
- The value of the property has been inherited from an ancestor node, whose value has been locked.

To edit property values:

1. Select a property in the property grid.
2. Click the **Value** field for a property and make changes.

 **Note:**

You can edit only properties that are not grayed out.

3. **Optional:** From **Property**, select an action from the Property menu:

- **Copy**—Copy a property value.
- **Remove Value**—Removes an overridden property value
- **Remove Duplicates**—For a normal, defined property, removes any value equal to the default value of the node from its descendants. For an inheriting property, removes any value equal to the inherited value of the node from its descendants. For a read/write derived property, removes any value equal to the derived value of the node from its descendants.
- **Clear All Below**—Clears the value on all descendant nodes which have overridden their inherited values and replaces the value with the default value from the inheritance rules
- **Lock**—Prevents changes to the value in inheriting descendant nodes.

 **Note:**

Only inheriting property values can be locked.

If prior to applying the Lock, a descendant node inherited value was overridden, the Lock does not affect that node. To affect all descendants, use the Lock and Clear All Below option.

- **Lock and Clear All Below**—Prevents changes to the value in inheriting descendant nodes, including any descendant nodes where the inherited value was overridden.
4. Take an action:
- Click **Save** to save your changes.
 - Click **Cancel** to exit without saving changes.

Filtering Properties

By default, all node-level properties are available for all nodes in an application. If node types are enabled for a hierarchy, the properties displayed for a node in the hierarchy are automatically filtered to only those applicable to the node type. You can filter the properties displayed for a category by label or value using a filter row. Data Manager role users can disable the node type filter on properties to view all properties for a node without needing to disable node types for the entire hierarchy.

 **Note:**


Filtering of node type properties applies only to the Oracle Data Relationship Management web client and does not apply when using the Data Relationship Management batch client or API.

To filter properties by label or value:

1. In the property grid, click  , and then click **Show Filter Row**.

2. In the **Label** or **Value** text box, enter text by which to filter and then press **Enter**.

To disable property filtering by node type,

1. In the property grid, click  .
2. **Select Show All Properties.**

7

Validating Data

Versions, hierarchies, nodes and their properties are checked for data integrity using Oracle Data Relationship Management validations. Validations can operate in real time, in batch, or in both modes. The modes in which a validation will run can be restricted based on the validation level and selected modes of operation specified at the time of creation, as well as the selected modes specified at the time of assignment to a version, hierarchy, or node.

Real-time validations run at the time a change is being made to a hierarchy node and will prevent the change from being made if the validation fails. Real-time validations must be previously assigned at the version, hierarchy, or node level in order to run. The validation will run for the current node and its parent node. If the nodes being validated are only located in the hierarchy being changed, then only validations for that hierarchy will run. If the nodes being validated are also located in other hierarchies, real-time validations will also run for any global changes that affect those other hierarchies.

Batch validations are run explicitly by a user at the version, hierarchy, or node level. You can run a selected set of validations or all validations that have been assigned to data. Version batch validations can run for all hierarchies and nodes within the version. Hierarchy batch validations can run for all nodes within a hierarchy. Node batch validations can run for the current node only. Export batch validations can run for the hierarchies and nodes associated with an export profile.

Both real-time and batch validations that run at the node level can be filtered using node types. If node types are enabled for a hierarchy, the validations that will run for nodes in that hierarchy will be limited to only those validations that are assigned to the node type of each node. For more information on node types, *Oracle Data Relationship Management Administrator's Guide*.

Validating Versions

To run version batch validations:

1. From **Browse**, select a version.
2. From **Versions** select an action:
 - **Validate**, then **Selected** to select the validations to run.
 - **Validate**, then **Assigned** to immediately run the batch validations assigned to the version.
3. If you selected **Validate-Selected**, select the validations to run and click **OK**.

Note:


You can select **Assigned** to select all of the assigned validations.

Validation results for versions and hierarchies are displayed on a separate tab on the Home page. Results are grouped by validation level. Hierarchy node results are displayed on a separate tab when a hierarchy is opened.

Validating Hierarchies

To run hierarchy batch validations:


1. From **Browse**, select a version and a hierarchy.
2. From **Hierarchies**, select an action:
 - **Validate**, then **Selected** to select the validations to run.
 - **Validate**, then **Assigned** to immediately run the validations assigned to the hierarchy.
3. If you selected **Validate-Selected**, select the validations to run and click **OK**.

 **Note:**

You can select **Assigned** to select all of the assigned validations.

Validating Nodes

To run node batch validations:

1. From **Browse**, select a version and a hierarchy.
2. Click .
3. From **Nodes**, select an action:
 - **Validate**, then **Selected** to select the validations to run.
 - **Validate**, then **Assigned** to immediately run the validations assigned to the node.
4. If you selected **Validate-Selected**, select the validations to run and click **OK**.

 **Note:**

You can select **Assigned** to select all of the assigned validations.

Working With Validation Results

Navigating Version and Hierarchy Results in List View

Batch validation results are displayed in a list on the Validation Results tab of the Home page. Validation failure messages are grouped by version and hierarchy. You can navigate from hierarchy validation results to node-level validation results.

Navigating Node Results in List View

Nodes that failed batch validations are displayed in a list on the Validation Results tab for an open hierarchy. The failure messages for each node can be viewed by clicking the plus sign next to the row for the node. You can directly edit the properties for selected nodes from the validation results.

From the Node menu, you can select these options:






- **Take** — Copies a node to the Clipboard from which you can insert it, move it, reorder it, or paste its properties somewhere else.
- **Go to Node** — Takes you to the selected node in the tree view
- **Show Filter Row** — Opens a blank row that you can use to filter the list
- **Properties** — Displays the node properties on the Properties tab

Navigating Node Results in Tree View

On the Tree tab of an open hierarchy, nodes that failed batch validations are displayed with red exclamation marks on the left side of the row for each node.

You can use the following toolbar buttons:

Table 7-1 Tree View Toolbar Button

Toolbar Button	Description
	Navigates to the parent of the selected node.
	Navigates to the previously selected marked node.
	Navigates to the next marked node.
	Expands the tree to display the marked nodes.
	Clears the marked nodes.

Downloading Node Results

From the List and Tree views, you can download the validation results to the following file types:

- Portable Document Format (PDF)
- Rich Text Document (RTF)
- Spreadsheet (XLS)



Note:

In List View, you can also download to a CSV file.

To download query results in list view:

1. From **Download**, select a file type.
2. Click **Save**.
3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.

Clearing Validation Results

Batch validation results can be removed from display by clearing them at the version or hierarchy level.

To clear validation results for a version and all hierarchies in the version:

1. Select a version.
2. From **Versions**, select **Clear Validation Results**.

Validation results are no longer displayed for any version, hierarchy, or node within the version.

To clear validation results for a hierarchy and all of its nodes:

1. Select a version and hierarchy.
2. From **Hierarchies**, select **Clear Validation Results**.

8

Governance Workflows

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. Through data governance, organizations can exercise positive control over the processes and methods used by their data stewards and data custodians to handle data.

In Oracle Data Relationship Management, the Oracle Data Relationship Governance module addresses the governance challenge in two critical ways:

- It enables the organization to configure change request management processes that allow business users—in different lines of business, divisions, and functions—to contribute towards the creation, management, and automation of processes associated with authoring and managing changes to master data.
- It allows data stewards and custodians to identify issues related to data quality—missing, erroneous or non-standard information—and route them to stakeholders within the organization who are best able to fix or address the issues based on data access privileges.

Governance workflows enable data stewards to orchestrate the entry, validation, and approval of data changes using repeatable business processes which result in high quality data, policy compliance, user collaboration, and change awareness across the business community.

Workflow Models

A workflow model controls the user tasks, stages of workflow, and types of data involved to govern a particular set of changes to data in Oracle Data Relationship Management. Each request uses a workflow model to determine:

- The workflow tasks to be performed
- The workflow stages to be completed
- The users groups who can work with the request
- The approval method and sequence
- How and when users are notified about the request
- The versions, hierarchies, and types of nodes available to the request
- The length of time that the request should take to complete

Workflow Tasks

A workflow task represents a change action (Add, Update, Insert, Move) to be performed along with a set of properties to be displayed, edited, and required during the task. Workflow tasks are defined separately from workflow models and can be used by multiple models to ensure consistency of common tasks performed in different business processes.

Workflow Stages

A workflow model consists of multiple workflow stages. When a stage is defined for a workflow model, the stage type defines the level of participation for users in that stage of the workflow. A workflow model has:

- A single Submit stage which is defined by a list of allowable workflow tasks that can be used to submit changes
- An optional series of Approval or Enrichment stages, each consisting of a single workflow task
- A single Commit stage used for final approval and commitment of all workflow tasks performed

Working with Requests

Requests are used to initiate changes to be completed, approved, enriched, and committed by other Oracle Data Relationship Management users using governance workflows. A request represents a collection of changes for a specific set of data to be performed after the request is validated, approved, and committed. Each request uses a single workflow model to control which users participate in the request, when they participate, and their type of participation.

Change Requests

A change request can be submitted by a governance user who has Submit access to both a data set and a workflow model that is configured to be used with the data set. Users who are granted Submit access to data and a workflow model are typically the users who will be initiating change requests for that data to be governed using that model.

Remediation Requests

Remediation requests are used to initiate some corrective action to be taken by another user. A remediation request can be submitted by any interactive user who also is a governance user and has been granted Read interactive access and Submit workflow access to a data set and a workflow model. A remediation request is typically created from the results of another operation such as a batch validation, query, or comparison. Nodes from those results can be added to a remediation request using the node Clipboard.

Request Items

Request items are used to perform workflow tasks within the context of a governance request. Each request item uses a single workflow task which must be specified when the item is added to a request. The workflow task defines the types of changes to be performed for a hierarchy node in the Data Relationship Management version used by the request.

A request may consist of one or more request items. During Submit or Enrich workflow stages, new request items can be added to a request to perform new tasks. During Enrich stages, request items added in previous stages can be modified or overridden using an Update task assigned to the workflow stage.

Request items can be added manually or loaded from a source file. The order of request items in a request is important because it controls the order that the changes

in the request get applied to the target version for validation and commitment purposes.

Request items can only be deleted during workflow stages which have a matching workflow task to the request item.

Creating a Source File

You can add items to a request from a source file. Items are loaded to the current workflow stage, for a single hierarchy and workflow task.

Source files must be in a tabular, delimited format. The first record in the file is used to identify the request item property to which each field will be loaded. Only editable properties for the selected workflow task will be loaded from the file. Fields which are not mapped to editable properties for the request item will be ignored during the load. The Name property is required for every request item loaded from a source file.

Some guidelines for creating a request item source file are:

- For Insert tasks, Name is the node to be inserted, Parent Node is the target parent, and the hierarchy you select in the Load Request Items dialog box is the target hierarchy.
- You can specify all data values for a request item in separate fields of the same record in a flat file.
- You must define which fields in the file correspond to which properties of the workflow task for a request item using column header record in the file (using property labels).
- You can specify property labels for workflow task properties in the column header record in any order.
- The property label match is case-insensitive.
- The Name property is required for all records in source file.
- Any editable property for the selected workflow task can be loaded from a source file.
- Columns where header record value does not map to an editable property for the workflow task will be ignored during the file load process.
- Use a blank value for a field where you do not want to supply a value for the property. In cases where you want to override a property with a null value, use the Blank Value Indicator option when loading the file.
- If a task property is defined with a custom property label, then use the same custom field label in the file.

Source File Example

```
#New Financial Accounts
Name,Parent Node,Description,Account Type,Start Date,End Date,Allow Posting
6000,5000>Total Employee Expenses,Expense,,,N
6200,6000,Compensation Expenses,,,,N
6210,6200,Vacation Paid,,1/1/2015,12/31/2015,Y
6220,6200,Employee Benefits,,1/1/2015,<blank>,Y
```

Item Details

Property details are displayed for a selected request item in a request. The properties which are available for display and edit are controlled by the workflow task for each request item or the task for the current workflow stage. For request items where new hierarchy nodes are being created, a single column of proposed new values is displayed. For request items which

refer to existing hierarchy nodes, two columns of values (Current and New) are displayed. The Current Values column displays existing property values while the New Values column displays changed property values. Modified properties are marked for easy identification. Calculated property values are displayed as read only. Properties are tagged with validation failures for quick resolution. To remove a property update from a request item, click the **Revert to Previous Value** link.

Renaming Request Items

You can rename a request item using the **Rename** link provided under the Name property in the request item details. The rename feature changes the name of the request item and synchronizes the new name to other request items which refer to it as a parent or use a node data type property. You can rename a node at any workflow stage where the Name property is editable.

Note:

If you want to change the node name back to the original name, use the Rename link and re-enter the previous name. The **Revert to Previous Value** button is not available for the Name property.


Task and Property Instructions

Instructions for request items can be viewed on the Request page. Instructions may be defined for workflow tasks and their properties. In the instructions area, you can view the instructions defined for the request item task. You can choose to show or hide the task instructions. In the Item Details section, you can view instructions defined for properties of the request item task. Instructions are displayed above the request item property value.

To view task and property instructions:

1. From the Worklist, select a request and open it.
2. Select a request item to view details of the item.
3. In the **Instructions** area, view the instructions defined for the task.

Note:

You can click **Hide** to hide the task instructions. If the instructions are hidden, click  to show them.

4. In the **Item Details** section, view Property instructions which are displayed above the property field.

Request Actions

When you are working with a request, action buttons in the upper right corner of the page header allow you to perform actions on the request. Certain actions will only be available depending on the situation. The following table describes the actions and provides the corresponding action button.

Table 8-1 Action Buttons















Action Button	Description
Save 	Saves the request to the Data Relationship Management repository in its current state without validating or approving the request This action is available for new requests and claimed requests.
Export to Excel 	Exports the request items in an Excel (.xls) format.
Export to CSV 	Exports the request items in an Comm-separated (.csv) format
Submit 	Validates the request items in the request and if successful, assigns the request to the next stage in the workflow path This action is available for draft requests or requests that have been pushed back to the submitter.
Calculate 	Calculates request item details based on proposed changes in the request
Validate 	Validates proposed changes for request items in the request This action checks required values and runs batch validations configured for the task for the request item or the current workflow stage. Validation failures are returned for correction. Requests can be validated if: <ul style="list-style-type: none"> • The request is in the Draft workflow status and has at least one request item • The request is in the Claimed workflow status
Copy 	Copies request items to a new request To copy a request, you must have access to submit the type of request you have selected.
Claim 	Claims the request for the active workflow stage. A submitted request can be edited, pushed back, escalated, approved, or rejected only by a user who has claimed the request. When a user claims a request, the request is locked and cannot be claimed by any other user. The user must unclaim the request to release it to be claimed by another user.
Unclaim 	Removes the claimed lock from the request but does not validate or assign the request to a different stage, user, or group If a user chooses to push back, escalate, approve, or reject the request, the request is automatically unclaimed.

Table 8-1 (Cont.) Action Buttons

Action Button	Description
Pushback 	Pushes back the request to a previous user in the current stage or a previous stage to correct some element of a request or ask for more information about the request The user pushing back the request must provide a comment about why the request has been pushed back.
Escalate 	Escalates the request to Data Manager role users Users may escalate a request for any number of reasons; for example, if they are unsure about what to do with a request or if they have access issues. Data Managers may resolve issues by adding comments to the request, changing the request items, modifying node access group assignment to hierarchies and nodes, or rejecting the request.
De-escalate 	Reassigns the request to the next node access group for the current workflow stage
Approve 	Approves and commits the changes in the request to the target version
Reject 	Rejects the request A submitted request may be rejected by any approving user in the workflow path or a Data Manager during escalation. Rejecting a request terminates the workflow for the request. A rejected request cannot be resubmitted. To resubmit the request items in the rejected request, the original submitting user or another Submit stage participant can copy the rejected request to a new draft request, edit the request items in the new request, and then submit the new request.

Request Comments

Comments entered by participating users of a request can be viewed on the Comments tab. Assigned users can add new comments as needed.

To add a comment to a request:

1. Open a request from the Worklist.
2. On the **Comments** tab, click  , enter a comment, and then click **OK**.


Request Attachments

Users who are creating a draft request or who have claimed a submitted request can upload and attach an electronic file to the request. Files up to 8 MB in size can be attached to requests during any stage where the request can be modified. Users can


view and download attachments for any request to which they have access via their user roles and workflow node access group membership.

Request attachments can be deleted from a request by the user who added the attachment or by a user with the Data Manager role.


To add an attachment to a request:

1. Open a request from the Worklist.
2. On the **Attachments** tab, click , browse to and select the file to attach. Optionally, enter a comment, and then click **OK**.

To download a request attachment:

1. Open a request from the Worklist.
2. On the **Attachments** tab, click , select to open the file or save the file, and then click **OK**.


To delete a request attachment:

1. Open a request from the Worklist.
2. On the **Attachments** tab, click , and confirm the deletion.

Request Participants

Users who participated in a request can be viewed on the Participants tab. A selected user can be contacted if more information about their participation is required. You can view users who have participated in a request, the stage they participated in, the workflow action taken, and the time and date the action was taken.

To email request a request participant:

1. Open a request from the Worklist.
2. On the **Participants** tab, click  next to the participant whom you want to contact.
3. Create email content, and then click **Send**.

Request Activity

All activities taking place for a request are recorded in the request activity. These activities include user activities such as submitting, approving, and enriching requests and system initiated activities such as assigning and committing requests. User comments are also recorded as request activities. Each activity includes a timestamp and name of the user if applicable. Activities are displayed with the most recent activity at the top of the list.

To view user and system activity for a request:

1. Open a request from the Worklist.
2. Click the **Activity** tab.

Workflow Tags

Workflow tags allow governance users to enable special handling of the request by a governance workflow. The following workflow tags can be added to a request that can be edited:

- **Due Date**—Identifies a user-defined date for which changes in a request should be committed. The Due Date for a request establishes the date at which point the request gets marked as overdue if not yet committed. A separate worklist view is available for Overdue requests.
- **Urgent**—Marks the request as high priority or time sensitive. A separate worklist view is available for Urgent requests.

To add a workflow tag:


1. Open a request that can be edited.
2. On the right side of the request next to **Workflow Tags**, click **Add**.
3. Select the workflow tags to add:
 - Due Date

 **Note:**

Make sure to select a date from the calendar.

- Urgent
4. Click **OK**.

To remove a workflow tag:

1. Open a request that has workflow tags assigned.
2. On the right side of the request, under Workflow Tags, click  next to the workflow tags to remove from the request.

Workflow Path

The workflow path for a request identifies the stages of workflow to be completed, the active stage for the request, the completion status of previous stages, and the approval count for the active stage. The workflow path enables all participating users to understand how long a request may take, how many approvals may be involved, and where a request is positioned in the overall approval process. The workflow path is automatically updated as users perform actions such as submit, approve, enrich, push back, or reject a request.

The workflow path for a request is determined by:

- The workflow model specified for the request
- The request items in the request
- The hierarchy nodes associated with the request items
- The workflow user groups and their access levels to the hierarchy nodes
- The workflow stages configured for the workflow model
- The current stage for a request
- The status of each stage


Request Lifecycle

The lifecycle of a request is determined by its workflow model. The workflow model controls the stages of workflow, the tasks which can be performed, and the users who can participate in the request. During its lifecycle, a request must go through at least two workflow stages (Submit and Commit) but, depending on the model, it may need to go through additional approval or enrichment stages before it can be committed.

Creating Requests



A request can be created by any governance user or data manager who is assigned to the initial Submit stage of a workflow model. The creating user can add request items to the request to perform tasks available for the workflow stage. After request items have been added, the request can be saved as a draft request.

To create a request from the Worklist:

1. From the Worklist page, click  **New Request** and select a Workflow Model to use for the request.
2. Select a target version for the request.

Note:

If the Workflow Model uses a version variable, then the version is preselected and displayed as a read-only label and cannot be changed.

3. Click  to manually add items or click  to load items from a source file. For information on source files, see [Creating a Source File](#).
4. If you selected to manually add items to the request, continue to step 5. If you are loading items from a source file, configure the following options, click **OK**, and continue with step 7:
 - **Device**—Select **Client File** or **Server File**. If you selected Server File, select the **Connection**. Click **Browse** to select the source file.
 - **Delimiter**—Select the delimiter used in the source file.
 - **Character Encoding**—Select the encoding option used by the source file.
 - **Strip Quotation Marks**—Select to remove single and double quotation marks around values in the source file.
 - **Task**—Select a workflow task for the request items.
 - **Hierarchy**—Select a hierarchy to use for all request items being loaded from the source file.
 - **Allow Blank Overrides**—Select for blank values in the source file to be treated as blank value overrides for each item according to the **Blank Value Indicator**. If not selected, then blank values in the source file are ignored.
 - **Blank Value Indicator**—Define a text value to indicate that the current property value for an existing node referenced by the request item will be overridden with a

blank value. If no indicator is specified, then any blank column value will be interpreted as a blank value override.

 **Note:**

To upload items to a request from a file, specify the following properties in the file header, and in each line in the file to upload:

- For Add actions: Name, Parent, Description
- For Insert actions: Name, Parent
- For Move actions: Name, Parent

5. Depending on the scenario, the **Add Items** dialog box displays one or more of the following fields. Select valid information for the request item:
 - **Task**—Select a task from the list.
 - **Hierarchy**—Select a hierarchy to limit searching and browsing to a specific hierarchy.
 - **Search**—Enter a text value and click Search. Search results are displayed in the Available list.
 - **Browse**—Browse the structure of the selected hierarchy or perform a search. Search results are found in the hierarchy structure.
 - **Items**—Select from a list of nodes used for existing request items. This list allows a request to contain multiple request items for the same node.
 - **Clipboard**—Select from a list of nodes that were taken and copied to the Clipboard from a Browse, Query, or Compare page.
 - **Available List, Selected List**—Move nodes to the Selected List.
6. Click **OK**.
7. In the **Item Details** section of the page, define new property values for the request item.

 **Note:**


Required values are marked with an asterisk.


8. **Optional:** Click **Rename** to rename a node.
9. Click **Update** to temporarily save changes for the selected request item. Click **Cancel** to discard the changes made to the selected request item.

 **Note:**

Changes made to a request item are implicitly saved when you perform another request item action such as selecting a different request item, moving a request item, or adding new request items. Changes to a request item cannot be discarded using Cancel after you have navigated away from the request item. In this case, the only way to discard those changes is to close the request without saving it.

 **Tip:**

Properties that have been updated are marked as modified. Click  to remove a property update from the request item.

10. Repeat steps 3-8 to add items to the request.
11. Click , enter a title for the request, and then click **OK**.

Calculating Request Item Properties

Request items may have properties whose values are calculated instead of manually defined or loaded from a file. These properties are marked with a Calculated Value icon. Property values are calculated for all request items in a request.

To calculate request item properties:

1. Add request items to a request.
2. Define property values for the request items which may be used by calculated properties.
3. Click **Calculate** for the request.
4. View the calculated property values for each request item.

Validating Request Items

Request items are validated at each workflow stage a request enters during its lifecycle. The validations run for each request item are based on the configuration of the workflow model for the request. All request items must pass validations for a stage before it can proceed to the next stage in the workflow path.

To validate request items:

1. Add request items to a request.
2. Define property values for the request items.
3. Click **Validate** for the request.
4. View validation failures for each request item and their properties.
5. Make corrections to the request items.
6. Click **Validate** again until the request passes validations.


Submitting Requests

After a request has been created and saved, the request can be submitted for approval, enrichment, and commitment by other governance users using the workflow model of the request. At least one request item must be added to a request, all required values must be provided, and validations must pass successfully for all request items. After the request has been submitted, the request is moved to the next workflow stage in the workflow model and the request can no longer be edited by the submitting user. Governance users are able to track the progress of requests they submit from the Worklist page.

Caution:

Do not submit a request against a detached version.

To submit a request:


1. Create a request or open a drafted request.
2. Click .

Submitted requests proceed through approve, enrich, and commit stages where they are assigned to users for approval. When a request enters a stage, request items in the request are matched to user groups configured for the stage and users in the matching groups have the request assigned to them in that stage. A request can be assigned to multiple users and groups for the same stage concurrently but only one user can claim the request at a time.

Claiming Requests

When a request is assigned to a workflow stage, all user groups associated with the stage have the request assigned to them. An assigned request must be claimed by a user before any other action can be performed on it. The user who claims an assigned request makes it unavailable for any other assigned user to claim it. While the request is claimed, the user can perform any operations made available to them based on the active workflow stage. If a user approves, pushes back, escalates, or rejects the request, the request is automatically unclaimed and the user no longer can take action on the request for that workflow stage. If a user has completed their tasks but will not approve, push back, escalate or reject the request, they have the option to unclaim the request.

To claim a request:

1. From the Worklist, open a submitted request.
2. Click .

Unclaiming Requests

Unclaiming a request makes the request available for any other assigned user to claim for the active workflow stage. The unclaim option is available to the user who currently

has claim to the request and to users with the Data Manager role. This user can unclaim a request that is currently claimed by another user.

To unclaim a request:

1. Form the Worklist, select the request to unclaim.
2. Click **Unclaim**.


 **Note:**

If a request exceeds the claim duration configured for the workflow model, then the request is automatically unclaimed to make it available for other assigned users to claim.

Approving Requests

Governance users who have proper workflow data access to all request items and are associated with a workflow model for a request will be assigned as available approvers. Every participant who claims a request becomes an actual approver of the request. The approval of a request signifies that the user has performed the tasks assigned to them during their workflow stage and agrees to allow the request to proceed toward being committed. All request items in a request must have required values provided and pass active validations for the workflow stage. A user successfully completes their participation for a stage by approving the request. Users for an Approve workflow stage may provide their approval but are not able to modify request items in the request.

To approve a request:

1. From the Worklist, open a claimed request.
2. Review request items in the request.
Select a request item to review item details.
3. Click .


Enriching Requests

Enrichment stages are used to update request items that were added during submission, to add request items for making local changes to the same nodes in different hierarchies, or to make changes to entirely different nodes that are related. An Enrichment stage can only have a single workflow task. If a request is claimed by a user for an Enrichment workflow stage, the request items can be edited based on the active workflow task assigned to the stage. If the active workflow task only permits the update of existing request items added by users in other workflow stages, the item details can be edited for all request items that are accessible based on the user's workflow access rights to the data. If the active workflow task for the stage performs any other type of data change such as adding a new value or maintaining hierarchy relationship, new request items can be added to the request but request items added in previous stages cannot be modified.

After a request has been enriched, changes to the request should be saved and validated. In some cases, request items will have detail values that are calculated and cannot be edited.

To view the calculated values based on the changes in the request, the request must be calculated.


To enrich a request:

1. From the Worklist, open a claimed request.
2. Add or update request items.
3. Click .

Pushing Back Requests

An approver may push back a claimed request to a previous workflow stage to question the purpose, accuracy, or incompleteness of the request. During the push back, the approver selects the stage to which the request will be assigned and is required to provide a comment to identify the reason for pushing back the request. After the request is assigned to the previous stage, assigned users may claim the request to modify it and resubmit or approve it. If the request has additional request items that were added in a later stage to which the approving user does not have proper workflow data access, details for the request item will not be displayed. After the previous participant addresses the reason for the push back and resubmits or approves, the request may return to the approver who pushed it back originally.

To push back a request:

1. From the Worklist, open a claimed request.
2. Click .
3. Select the Workflow Stage to which to push back the request.
4. Enter a comment to explain the push back and click **OK**.

Example 8-1 Reapproval

In situations where a request is pushed back to a previous stage that has already been completed and approved by previous users, the request may need to be resubmitted or reapproved by users who have already participated. Depending on which stage the request is pushed back to, reapproval by subsequent stages may be required or optional. The reapprove option for each workflow stage in a workflow model controls this behavior for requests using that model.


Escalating Requests

In rare cases, an approver may not be able to perform the tasks required for their workflow stage. This situation may happen if the workflow model or data access are not configured correctly to handle a specific type of request or the user is not sure of how to proceed. The user has the option of escalating a claimed request to Data Manager role users and users who are members of workflow node access groups configured for the Commit stage of the workflow model to involve them in resolving the situation.

When escalated, the request is assigned to these users and normal approvers for the stage cannot claim the request to participate. While escalated, data managers and members of the assigned workflow node access group can modify the request, make direct edits to data related to the request, change workflow data access for participating users, or contact an application administrator to alter the workflow model.

If these users permit the request to proceed, the request may be de-escalated to be reassigned to the original user groups for the workflow stage.

To escalate a request:

1. From the Worklist, open a claimed request.
2. Click .
3. Enter a comment to explain the escalation and click **OK**.

Example 8-2 Automatic Escalation


When a request is assigned to the next stage in the workflow path, Oracle Data Relationship Management matches user groups associated with the stage in the workflow model to the user groups with appropriate workflow access to the data for the request items in the request to determine which users can participate in the request for the stage. This intersection will typically result in at least one user per stage, but in some cases there may be no matching users for a particular stage.

A request cannot be promoted to the next stage if any request items do not have required values or fail validations configured for the stage, or if the stage requires approval. In this case, the request is automatically escalated for the stage.

Rejecting Requests

Any approver of a request, or a data manager during escalation, has the option of rejecting the request to prevent it from being committed. The approver may reject a request for any reason including duplication, inaccuracy, or incompleteness of changes being requested. During the rejection, a comment is required to describe the reason for the rejection. After a request is rejected, the workflow model is cancelled and users are no longer able to perform any action on the request. A rejected request may be later copied and submitted as a new request if applicable.

To reject a request:

1. From the Worklist, open a claimed request.
2. Click .
3. Enter a comment to explain the rejection and click **OK**.

Handling Overdue Requests

If the age of a request exceeds the request duration configured for the workflow model or the due date configured for the request, then the request is marked as overdue. A separate worklist view is available to identify Overdue requests assigned to you. Review the overdue requests and take action.

Skipping a Workflow Stage

Requests can skip workflow stages which are deemed unnecessary for their request items. The ability to enter or skip a stage is controlled by the stage condition configured in the workflow model. When a stage is skipped, the request immediately moves to the next stage in the workflow path.

Promoting Requests

For Approve, Enrich, and Commit stages, a request will enter a stage even though no approving user groups can be found for the request items in the request. In this situation, the request can be promoted to the next stage in the workflow model if the current stage does not require approval and the request items pass validations for the stage. The stage is marked as promoted in the workflow path for the request.

Splitting Requests

Requests containing multiple request items may be split into separate requests during a workflow stage to allow unrelated items to follow different workflow paths. Some items must enter a workflow stage to be approved or enriched, while other items can skip the stage and proceed to the next one. When a request is split, request items that meet the stage condition are moved to a separate, submitted request using the same workflow model. The new request enters the workflow stage for approval. Items that do not meet the stage condition remain in the original request and the stage is skipped.

Committing Requests

Users assigned to the final Commit stage in the workflow model for a request must provide their approval to commit the changes in the request to the target version of data. During the Commit stage, all required values for all request items must be provided and all validations must pass. An approver can edit any request item to address remaining issues or make final adjustments. A committing user must approve all request items in a request. After the request is ready to be committed, the approver provides their approval to initiate the commit operation. If there are multiple approvers for the Commit stage, the last approval will trigger the commitment of changes in the request.

Note these transaction history details for committed requests:

- For auditing purposes, the final approving user of a request in the Commit stage is assigned to the transactions created in transaction history for the committed request items.
- In cases where a request may skip or be promoted in the Commit stage, the version or hierarchy owner is assigned to the transactions in transaction history.

Navigating the Worklist

The Worklist is a central location for interacting with change and remediation requests. From the Worklist, governance users may submit change requests or review and participate in requests assigned to their user group. Data managers may submit remediation requests to governance users or participate in escalated requests submitted by them.

Worklist Views

The Worklist includes a set of views which enable governance users to filter displayed requests based on how the user is able to participate. Each worklist view displays a count of the requests available in that view. Select a worklist view from the menu on

the left side of the Worklist page to display requests available in the view. Worklist views are:

- **Assigned**—Requests assigned to the user or a group in which the user is a member
- **Urgent**—Requests assigned to the user or group and marked as Urgent
- **Overdue**—Requests assigned to the user or group and marked as Overdue
- **Claimed**—Requests assigned to the user or group and claimed by the current user
- **Submitted**—Requests submitted by the current user
- **Drafted**—Requests saved by the user in Draft status but not submitted
- **Participated**—Requests to which the user has been added as a participant (previously submitted or claimed)
- **Notified**—Requests to which the user was notified (and may or may not have participated in)
- **All**—All requests currently in the system (available only for Data Manager role users)

Viewing Requests by Age



Requests displayed in the Worklist can be filtered using a request age filter. The request age filter limits the display of requests to Today, This Week, Last 30 days, Quarter-to-Date, Year-To-Date, or All.

Searching for Requests

Governance users and data managers can find existing requests using a keyword search. An asterisk can be used for wildcard searches. A workflow user can search in one of the following areas of a request:

- Request header
- Request items
- Workflow tags
- Comments

To search for requests:

1. On the Worklist page, click  , **Search In**, and then select a request area to search:
 - Request—Searches the request header for the specified text
 - Request items—Searches request items for the specified text
 - Workflow tags—Searches workflow tags for the specified text
 - Comments—Searches comments for the specified text
2. Enter text to search for, and then click .

Previewing Requests

When a request in the Worklist is selected, a preview of the request is displayed below the Worklist. The request preview displays:


- Request header
- Request items

- Request activity

Opening Requests

You must open a request to view the request details or to take action on a request displayed in the Worklist.

To open a request:

- From the Worklist, double click the request.
- From the request preview, click .


Deleting Requests

You may delete draft workflow requests and submitted workflow requests that have not been committed. The request submitter and users with the Data Manager role may delete:

- Draft workflow requests
- Workflow requests that have been pushed back to the Submit stage

Only users with the Data Manager role may delete workflow requests that have been submitted.

To delete a request:

1. From the Worklist, select a request to delete.
2. In the request preview, click .
3. Click **Delete the Request** to confirm the deletion.

Notifications

Notifications include both Web client alerts and e-mail notifications and are used to notify governance users and Data Managers of request activities for requests to which they are associated. The configuration of stages for the workflow model for a request controls whether users are notified of activities taking place in a particular stage and which users get notified.



Note:

Users do not receive notifications for actions they perform.

Select from these Notify options for each stage:

- **None**—No users are notified of actions performed for this workflow stage.
- **Assignees**—Users who belong to any workflow node access group currently assigned to the request are notified when Assign, Approve, Commit, or Reject actions occur.

Assignees are only notified if they are members of a workflow access group assigned to the stage with a Notify setting of either Assignees or Assignees and Participants.

- **Participants**

- When Commit or Reject actions occur, users who have submitted or claimed the request are notified.
- When Approve or Promote actions occur, users who have submitted the request are notified.

Participants are only notified if they are members of a workflow node access group assigned to the stage with a Notify setting of either Participants or Assignees and Participants.

- **Assignees and Participants**—Assignees and participants are notified.

Notifying Users

The following table lists actions that trigger notifications and the recipients of the notifications based on the Notify setting of each stage.

Table 8-2 Workflow Alerts

Workflow Action	Assignees	Submitter	Participants	Notify Users
Assign	X			
Approve	X	X		X
Promote		X		X
Escalate	X			X
Reject	X		X	X
Commit	X		X	X

 **Note:**

Notify Users are users who are members of a workflow node access group assigned to a stage with only Notify access to request items. They are only notified if the Notify setting is either Assignees or Assignees and Participants. If the Notify option is None or Participants, then these users are not notified

Web Client Alerts

When a user needs to be notified of request activity, the user receives an alert in the Oracle Data Relationship Management Web client. When a user receives an alert, an Alerts link will appear at the top of any Web page. The Alerts link displays a count of alerts to be viewed. Accessing the Alerts link will display the Alerts dialog where the detailed list of alerts can be viewed. If a user currently has an active session in the Web Client, the Alerts link will immediately reflect alerts that have been sent to them.

To access alerts:

1. At the top of the Web page, click **Alerts**.
2. Click on an alert to view detailed information.

 **Note:**

You can clear all alerts by click **Clear All**.

 **Note:**

When clicking the link to open an Alert, the Request instance page opens showing the current state of the request. All activity for the request is displayed regardless of when the Alert was issued.

Example 8-3 Email Notifications

Governance users may also be made aware of request activities using email notifications. This type of communication is useful to alert users who do not have an active user session or who may not frequently access Data Relationship Management. An email notification will be sent when a Web Client alert is raised.

- The subject of the email will indicate the request and the activity that took place.
- In the body of the email, users can view the request header, request items, and recent request activities.
- To view property details for request items, view all request activities, or take action on a request, a hyperlink is available to directly navigate the user to the request in the Web client after being authenticated.

9

Using Queries

Oracle Data Relationship Management enables you to query nodes in versions and hierarchies based on property values. From the Query task on the Home page, you can create, open, and manage queries. You can open multiple queries in separate tabs, but only one query can be in focus at a time. You can also run queries from a selected data object being browsed.

Within Data Relationship Management, query supports a number of operators such as Like, In, Length Equal, and so on. A list of criteria can be created (similar to the WHERE clause in a conventional SQL statement) that can be run against a set of nodes. The query returns a list of nodes that meet the defined criteria. Queries can comprise multiple criteria that can be grouped together using *AND* or *OR* to form complex queries.

Property queries may be saved so users can perform recurring analyses without having to redefine the query criteria each time. The Data Manager can create "standard" queries that are available for use by all other users.

Typical uses of queries:

- Locating nodes that meet certain criteria while browsing hierarchies
- As a filtering mechanism in exports and comparisons
- As the logic for a custom validation used to ensure data integrity

Query Types

Query types:

- **Global Queries** are performed against all nodes in a version. Because the global query runs against a set of nodes without reference to any specific hierarchy, only version and global node properties may be included in the criteria.
- **Local Queries** are defined and performed against a node and its descendants within a hierarchy. They can be run from the top node of a hierarchy or from a node within the hierarchy. Local queries can reference any property including version, hierarchy, global node, and local node properties.

Operator Descriptions

Table 9-1 Operator Descriptions

Operator	Description	Data Type Restrictions
Above	The specified value of the property is used by an ancestor node.	
Below	The specified value of the property is used by a descendant node.	

Table 9-1 (Cont.) Operator Descriptions

Operator	Description	Data Type Restrictions
Contains	The comma-delimited list value of the property contains the specified item in the list.	Strings only
Equal	The value of the property is equal to the specified value.	
Greater Than	The value of the property is greater than (but not equal to) the specified value.	
Greater Than, Equal	The value of the property is greater than or equal to the specified value.	
Has Characters	The value of the property has one or more of the specified characters.	
Has Not Characters	The value of the property does not have any of the specified characters.	
In	The value of the property is in the specified comma-delimited list of values.	Strings only
Is Alpha	The value of the property consists of alphabetic characters only.	
Is Alpha Numeric	The value of the property consists of alphabetic or numeric characters only.	
Is Blank	The value of the property is blank (the specified value field is unavailable).	
Is Defined	The value of the property is a defined (overridden) value.	
Is Not Alpha	The value of the property does not consist of alphabetic characters only.	
Is Not Alpha Numeric	The value of the property does not consist of alphabetic or numeric characters only.	
In Not Blank	The value of the property is not blank.	
Is Not Numeric	The value of the property is not numeric.	
Is Numeric	The value of the property is numeric.	
Length Equal	The length of the property value is equal to the specified value.	Strings only
Length Greater Than	The length of the property value is greater than (but not equal to) the specified value.	Strings only
Length Greater Than, Equal	The length of the property value is greater than or equal to the specified value.	Strings only

Table 9-1 (Cont.) Operator Descriptions


Operator	Description	Data Type Restrictions
Length Less Than	The length of the property value is less than (but not equal to) the specified value.	Strings only
Length Less Than, Equal	The length of the property value is less than or equal to the specified value.	Strings only
Length Not Equal	The length of the property value is not equal to the specified value.	Strings only
Less Than	The value of the property is less than (but not equal to) the specified value.	
Less Than, Equal	The value of the property is less than or equal to the specified value.	
Like	The value of the property is like the specified value (including wildcards). The asterisk (*) and percent sign (%) can be used as wildcards.	Strings only
Not Above	The specified value of the property is not used by an ancestor node.	
Not Below	The specified value of the property is not used by a descendant node.	
Not Contains	The comma-delimited list value of the property does not contain the specified item in the list.	Strings only
Not Equal	The value of the property is not equal to the specified value.	
Not In	The value of the property is not in the specified comma-delimited list of values.	Strings only
Not Like	The asterisk (*) and percent sign (%) can be used as wildcards.	Strings only


Creating Queries

When you create or open a query, it is displayed on a named tab.

The query wizard is used to define the parameters and criteria for the query. The wizard content is organized across Source, Style, Filters, and Columns tabs to guide input. Although the wizard tabs are ordered to gather input in a sequence, you can navigate between any tabs.


To create a query:

1. On the Home page, select **Query**.
2. Click .
3. **Optional:** Create substitution parameters for the query to pass in user-defined values to the query filter at run time.

- a. Click .
- b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
4. From the **Source** tab, select a version or version variable for the query.
You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
5. Perform an action:
 - For a global query, go to the next step.
 - For a local query, click , select a hierarchy and a top node and click **OK**

 **Note:**

To change the hierarchy/top node combination, click **Clear** and then repeat the steps for a local query.

6. On the **Style** select how the query results are displayed:
 - List — Nodes that match the criteria are returned in a dynamic list.
 - Mark Nodes — Nodes that match the criteria are marked in the hierarchy tree.
 - Both — Nodes that match the criteria are returned in a dynamic list and marked in the hierarchy.

 **Note:**

For global queries, the results can be returned only in a list.

7. **Optional:** If you selected Mark Nodes or Both, select Expand to Marked Nodes to automatically open and expand hierarchies with marked nodes.

 **Note:**



This option is applicable only to local queries.

8. On the **Filter**, click **Add** to insert a criteria row.
9. Select a property and **operator**, enter a value or substitution parameter for the criteria row, and press Enter to save the row.

 **Note:**

Substitution parameter formatting is <%ParamKey%>, where <% and %> are the identifiers that note a runtime substitution variable and ParamKey is the name of the parameter to be used for substitution for example, <%AccountType%>. You can include parentheses to group criteria and to define whether to include inherited values for the property. For operator descriptions see [Operator Descriptions](#).

 **Tip:**

Click  to modify a row. Click  to delete a row. Click the arrows to move a row.

10. From **Include Options**, select the nodes to include in the query results:
 - Match Only — Returns only the nodes that match the criteria
 - Match and Ancestors — Returns all nodes that match the criteria, plus their ancestors
 - Match and Descendants — Returns all nodes that match the criteria, plus their descendants
11. **Optional:** If you selected **Match and Ancestors** or **Match and Descendants** in step 9, by default, the **Inclusive** option is selected and includes the node matching the criteria in the result.
12. **Optional:** If you selected **Match and Ancestors** in step 9, you can select **Include ancestors above query top node** to include ancestor nodes that are above the top node.

The **Include ancestors above query top node** option includes additional ancestors only if the top node specified for the query did not include all ancestors (Match and Ancestors option). Thus, this option has no effect on descendants (Match and Descendants option).
13. Select the **Columns** tab and choose a Category.
14. Select the properties to display as columns in the query results and move them from **Available** to **Selected**.


 **Note:**

The Name property is always displayed in query results regardless of what properties are selected to display.


Column selections are runtime parameters only and are not saved with the query.

 **Tip:**

Use the arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

15. **Optional:** Select **Include Criteria Properties** to automatically include properties referenced in the filter criteria in the query results.
16. Do any of the following:
 - Click  to save the query.

You are prompted to define a name, description, and an object access group for the query. Select a custom object access group or one of the following:

 - **User**—Personal objects that are available only to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard queries.
 - **System**—Restricted objects that are available only to Application Administrator role users to view and run.
 - Click  to run the query.

Working with Query Results

Depending on the display options selected for the query, the results are displayed in a list, marked in the tree, or both. You can edit properties for nodes in a list of results for local queries. Structural changes and property edits are allowed for local query results displayed as marked nodes in the hierarchy tree. Global query results are read only.

Navigating in List View

Query results displayed in a list only show nodes that meet the query criteria. From the Node menu, you can select these options:







- **Take** — Copies a node to the Clipboard from which you can insert it, move it, reorder it, or paste its properties somewhere else.
- **Go to Node** — Takes you to the selected node in the tree view
- **Show Filter Row** — Opens a blank row that you can use to filter the list
- **Properties** — Displays the node properties on the Properties tab

Navigating in Tree View

Query results displayed as marked nodes in a tree will show the entire hierarchy tree with a blue checkmark next to each node that meets the query criteria.

You can use the following toolbar buttons:

Table 9-2 Tree View Toolbar Buttons

Toolbar Button	Description
	Navigates to the parent of the selected node.
	Navigates to the previously selected marked node.
	Expands the tree to display all nodes of the specified level.
	Navigates to the next marked node.
	Expands the tree to display the marked nodes.
	Clears the marked nodes.

Downloading Query Results

From the List and Tree views, you can download the query results to the following file types:

- Portable Document (PDF)
- Rich Text Document (RTF)
- Spreadsheet (XLS)



Note:

In List View, you can also download to a CSV file.

To download query results in tree view:

1. From **Download**, select a file type.
2. Click **Save**.
3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.



Managing Queries

You can open, copy, and delete saved queries.

Opening Queries

You can open and modify queries. You must open a query before you can run it.

To run a query:

1. Select the **Query** tab and choose a query.
2. Click  to open the selected query.
3. Click the tabs to modify the query parameters, and then click  to run the query.

Copying Queries

To copy a query:

1. Select the **Query** tab and choose a query.
2. From **Queries**, select **Copy**.
3. Enter a new name and description for the query.
4. From **Object Access Group**, select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard compares.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
5. Click **OK**.

Moving Queries

You can move a query from one object access group to another.



Note:

The move action is only available to users with Manage Object permission for the group.

To move a query:

1. Select the **Query** tab, and then choose a query.
2. From **Queries**, select **Move**.
3. From **Object Access Group**, select a different group, and then click **OK**.

Deleting Queries

To delete a query:

1. Select the **Query** tab, and choose a query.
2. From **Queries**, select **Delete**.



Note:

If the query is used by another object, you get a prompt with a list of objects that use the query. You can choose to continue with the deletion or cancel it.

10

Comparing Hierarchies

Oracle Data Relationship Management allows you to compare hierarchies in the same version or across two different versions. From the Compare task on the Home page, you can create, open, and manage compares. You can open multiple compares in separate tabs, but only one compare can be in focus at a time. You can also run compares from a hierarchy that is being browsed.

Compare Types

Visual Compare

The visual compare displays the hierarchy trees side-by-side without running a comparison process.

Structure Compare

The structure compare identifies nodes that are similar or different between two versions of the same hierarchy or two points of different hierarchies.

A structural compare returns nodes from each hierarchy that are similar or different based on the type of comparison being performed. The results can be displayed as two lists, two marked trees, or both.

Property Compare

The property compare identifies differences in property values between two versions of the same hierarchy or two points of different hierarchies. You can select properties to compare for each node that is common between the hierarchies. The results can be displayed as two lists, two marked trees, or both.

Rename Compare

The renamed node compare is the same as the structure compare, but it treats renamed nodes as being the same for comparison purposes.

For example, if Hierarchy 1 in a compare includes nodes A, B and C, and hierarchy 2 contains the same nodes but with node C renamed to CC, then the renamed node compare does not identify any differences between the two hierarchies. A structure compare regards nodes C and CC as different if the differences option is selected.



A renamed node compare returns nodes from each hierarchy that are similar or different based on the type of comparison being performed. The results can be displayed as two lists, two marked trees, or both.

Creating Compares

When you create or open a compare, it is displayed on a named tab.

The compare wizard is used to define the parameters and criteria for the compare. The wizard content is organized across Source, Style, Filters, and Columns tabs to guide input. Although the wizard tabs are ordered to gather input in a sequence, you can navigate between any tabs.


To create a compare:

1. On the Home page, select **Compare**.
2. Click .
3. **Optional:** Create substitution parameters for the compare to pass in user-defined values to the query filter at run time. See
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
4. Click **Source** and select the version or version variable for each hierarchy to compare.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
5. For each version, click , and from the **Select Node** dialog box, select a hierarchy and a top node, and click **OK**.
6. Select the **Style** tab and choose the compare type:
 - Visual
 - Structure
 - Property
 - Rename
7. If you selected a visual compare or a property compare, continue to the next step. If you selected another compare type, do the following:
 - Under the compare type, select whether to display differences or similarities.
 - From **Join**, select a join property.



By default, compares map nodes from each hierarchy that have the same name. You can also join nodes based on a different property. For example, if a property named SortCode has a unique key in the specified hierarchies, then this property is used to identify differences between nodes having the same SortCode in each hierarchy.
 - From **Result Display**, select a compare result to display.
 - List — Nodes that match the criteria are returned in a dynamic list.

- Mark Nodes — Nodes that match the criteria are marked in the hierarchy.
 - Both — Nodes that match the criteria are returned in a dynamic list and marked in the hierarchy.
 - **Optional:** If you selected Mark Nodes or Both, select Expand to Marked Nodes to automatically open and expand hierarchies with marked nodes.
8. Select the **Filter** tab and do one of the following:
- From **Inclusion Filter**, select a saved query.
 - Define a filter criteria for the compare by doing the following:
 - a. Click **Add** to insert a criteria row.
 - b. Select a property and **operator**, enter a value or substitution parameter for the criteria row, and press Enter to save the row.

 **Note:**

Substitution parameter formatting is <%ParamKey%>, where <% and %> are the identifiers that note a runtime substitution variable and ParamKey is the name of the parameter to be used for substitution for example, <%AccountType%>. You can include parentheses to group criteria and to define whether to include inherited values for the property. For operator descriptions see [Operator Descriptions](#).

 **Tip:**

Click  to modify a row. Click  to delete a row. Click the arrows to move a row.



9. From **Limb/Leaf Filter**, select nodes to include.
10. Select the **Columns** tab and choose a Category.
11. Select the properties to display as columns in the compare results and move them from **Available** to **Selected**.

 **Note:**

Column selections are runtime parameters and are not saved with the compare.

 **Tip:**

Use the arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

12. **Optional:** Select **Include Compare/Filter Properties** to automatically include properties referenced in the filter criteria for Property compares.
13. Do any of the following:
 - Click  to save the compare.
You are prompted to define a name, description, and an object access group for the compare. Select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard compares.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
 - Click  to run the compare.

Working with Compare Results

Depending on the display options selected before running the compare, the results are displayed in a list or a tree or both and are displayed side by side.

Navigating in List View

Compare results displayed in a list only show nodes that meet the compare criteria. From the Node menu you can select these options:

- **Take** — Copies a node to the Clipboard from which you can insert it, move it, reorder it, or paste its properties elsewhere.
- **Go to Node** — Takes you to the selected node in the tree view
- **Show Filter Row** — Opens a blank row that you can use to filter the list
- **Properties** — Displays the node properties on the Properties tab

Navigating in Tree View

On the Tree tab, marked nodes are displayed with blue check marks on the left side of the row.

Compare results displayed as marked nodes will show both hierarchy trees with a blue checkmark next to each node that meets the compare criteria.

Table 10-1 Tree View Toolbar Buttons







Toolbar Button	Description
	Navigates to the parent of the selected node.
	Navigates to the previously selected marked node.
	Expands the tree to display all nodes of the specified level.

Table 10-1 (Cont.) Tree View Toolbar Buttons

Toolbar Button	Description
	Navigates to the next marked node.
	Expands the tree to display the marked nodes.
	Clears the marked nodes.

Moving Nodes Between Versions

On the Compare results page, you can copy a node from one version to another by dragging the node.

To move a node to another version:

1. Select the node to move and drag it to the new location.
2. In the **Put Node** dialog box, select an option:
 - **Put as Child** — Adds the selected node as a child node.
 - **Put as Sibling or Reorder** — Adds the node as a sibling node or reorders the node within the hierarchy.
3. Select an option for copying node properties:
 - Do Not Copy Properties
 - Copy Global Properties
 - Copy Local and Global Properties
4. **Optional:** Select **Include Descendants** to include the node's descendants in the copy.
5. **Optional:** Select **Use These Settings as the Default** to use the same settings for additional nodes that you copy.
6. Click **OK**.

Downloading Compare Results

From the List and Tree views, you can download the compare results to the following file types:

- Portable Document (PDF)
- Rich Text Document (RTF)
- Spreadsheet (XLS)



Note:

In List View, you can also download to a CSV file.

To download compare results in tree view:

1. From **Download**, select a file type.

2. Click **Save**.
3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.



Managing Compares

You can open, copy, and delete saved compares.

Opening Compares

You can open and modify existing compares. You must open a compare before you can run it.

To open and run a compare:

1. On the Home page, select **Compare**, and then select a compare to open.
2. Click .
3. Select the tabs to modify the compare parameters, and then click  to run the compare.

Copying Compares

To copy a compare:

1. On the Home page, select **Compare**, and then select a compare to copy.
2. From **Compares**, select **Copy**.
3. Enter a new name and description.
4. From **Object Access Group**, select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard compares.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
5. Click **OK**.

Moving Compares

You can move a compare from one object access group to another.



Note:

The move action is available only to users with Manage Object permission for the group.

To move a compare:

1. On the Home page, select **Compare**, and then select a compare to move.

2. From **Compares**, select **Move**.
3. From **Object Access Group**, select a different group, and then click **OK**.

Deleting Compares

To delete a compare:

1. On the Home page, select **Compare**, and then select a compare to delete.
2. From **Compares**, select **Delete**.

 **Note:**

If the compare is used by another object, you get a prompt with a list of objects that use the compare. You can choose to continue with the deletion or cancel it.

11

Working with Action Scripts

Action Scripts allow you to process a bulk set of incremental changes in an automated fashion. Each record within the script represents a separate action to be performed and is processed individually from other actions. Actions of different types can be grouped together in the same script. Action Scripts are particularly useful when you need to perform the same set of actions for multiple versions, hierarchies, or nodes. You can leave the Action Script page while a script is running and return to the page later to view the results. You can also view results of the Action Script.

Caution:

When running Action Scripts, the regional settings of your client session are used to define the expected format for date and time values for the Action Script. If these settings are not detected, then the regional settings for the server are used. For example, if the regional setting for your session specifies that date should be formatted as dd.mm.yyyy, then the data given to the Action Script should be in that format. Otherwise, the information does not display or displays incorrectly.

Action Script Parameters

The parameters used by each action in each flat file column are as follows:

Table 11-1 Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
Activate Activates the specified node in the hierarchy	Version	Hierarchy	Node				

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
Add Adds the specified node (or domain node) to the specified hierarchy (This action can only be used for a node that does not already exist within the specified version)	Version	Hierarchy	Node	Parent node	Leaf property	Description Required if adding a domain node	Domain Required if adding a domain node
AddDomain ToVersion Adds a domain to a version	Version	Domain					
AddHier Adds a hierarchy to a version	Version	Hierarchy	Description	Top Node			
AddInsert Adds the specified node (or domain node) if it does not already exist in the specified version, otherwise the node is inserted	Version	Hierarchy	Node	Parent node	Leaf property Required only if adding a node	Description Required if adding a domain node	Domain Required if adding a domain node
AddOrphan Adds an orphan node (or domain node) to a version	Version	Node	Leaf Property	Description Required if adding a domain node	Domain Required if adding a domain node		
Annul Annuls the specified node	Version	Hierarchy	Node				

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
AnnulAllBelow Annuls all nodes following the specified node	Version	Hierarchy	Node				
AssignHierValidations Assigns validations at the hierarchy level	Version	Hierarchy	Validation				
AssignNodeToDomain Assigns node to domain	Version	Node	Domain				
AssignVersionValidations Assigns validations at the version level	Version	Validation					
ChangeHierTopNode Changes the top node for a specified hierarchy	Version	Hierarchy	New Top Node				
ChangeProperty Updates the value of the specified property of the specified node	Version	Hierarchy	Node	Property label or name Controlled by the Property References By Label option.	Value		

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
ClearProps Below Clears the values of the specified property in all of specified node's children (Corresponds to the <i>Clear All Below</i> command that is available to users within Oracle Data Relationship Management)	Version	Hierarchy	Node	Property label			
Delete Deletes the specified node from the version, and merges it with the merge node if one is specified	Version	Hierarchy	Node	(Merge node)			
DeleteHier Deletes a hierarchy from a version	Version	Hierarchy					
DeleteOrphan Deletes an orphan node	Version	Node					
Destroy Destroys the specified node from the version	Version	Hierarchy	Node				

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
Inactivate Inactivates the specified node in the hierarchy, and merges it with the merge node if one is specified	Version	Hierarchy	Node	(Merge node)	(Domain Node Inactive Date)		
Insert Inserts the specified node into the specified hierarchy (This action can only be used for nodes that already exist in another hierarchy within the specified version)	Version	Hierarchy	Node	Parent node			
LockProp Locks or unlocks the specified property of the specified node	Version	Hierarchy	Node	Property label	Is Locked (0=false, 1=true)		
Move Moves the specified node within the hierarchy	Version	Hierarchy	Node	Destination parent node			
NodeExists Returns True if the specified node exists in the specified version, otherwise returns False	Version	Node					

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
NodeExistsInHier Returns True if the specified node exists in the specified hierarchy, otherwise returns False	Version	Hierarchy	Node				
NodeHasAsParent Returns True if the specified parent node is the parent of the specified node, otherwise returns False	Version	Hierarchy	Node	Parent node			
PropCompare Returns True if the value of the specified property of the specified node equals the value listed in the flat file, otherwise returns False	Version	Hierarchy	Node	Property label	Value		
PropQuery Returns the value of the specified property of the specified node in the Action Script result set	Version	Hierarchy	Node	Property label			

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
Rem Indicates a remark/ comment row in the flat file (Any arbitrary text can be entered on other columns of this row for documentati on purposes —it is ignored by the Action Script process)	Remark	Remark	Remark	Remark	Remark		
Remove Removes the specified node from the hierarchy	Version	Hierarchy	Node				
RemoveDo mainFromV ersion Removes a domain from a version	Version	Domain					
RemoveDu pProps Removes (clears) duplicate properties for the specified node (Correspond s to the <i>Remove Duplicates</i> command that is available to users within Data Relationship Managemen t)	Version	Hierarchy	Node	Property label			

Table 11-1 (Cont.) Action Script Parameters

Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
RemoveHierarchyPropValue Removes (clears) the value of a hierarchy property	Version	Hierarchy	Property label				
RemoveProperty Removes (clears) the value of the specified property of the specified node (The property itself is not deleted because many other nodes may be utilizing this property)	Version	Hierarchy	Node	Property label			
RemoveVersionPropValue Removes (clears) the value of a version property.	Version	Property label					
SetControlledProperties Sets the controlling hierarchy for one or more properties.	Version	Hierarchy	Property List				
UpdateHierarchyPropValue Updates the value of a hierarchy property	Version	Hierarchy	Property label	Value			

Table 11-1 (Cont.) Action Script Parameters


Action Name	Param1	Param2	Param3	Param4	Param5	Param6	Param7
UpdateVersionPropertyValue Updates the value of a version property	Version	Property label	Value				

Loading Action Scripts From Flat Files

Action Scripts read rows from the specified flat file. Each flat file row defines an action to be performed in Data Relationship Management. Each row is read and the appropriate actions are performed sequentially. A status message is returned for each row. Details are displayed in the Return Value column.

To load an Action Script from a flat file:



1. On the Home page, select **Script**.
 2. From **Source Type** select **File**.
 3. Click **Browse** and select the flat file to use for the Action Script.
 4. Select from these options:
 - **Column Order**—Use the arrows to customize the order of the six required columns from the flat file.
 - **Character Encoding**—Select one of the encoding options.
 - **Delimiter**—Select the column delimiter that is used in the flat file.
 - **Strip Quoted Strings**—Select to remove quotation marks.
 - **Property References By Label**—Select to use property labels instead of names.
- ▲ Caution:**

If you reference a property that does not have a unique label, the system will require additional information in order to select the appropriate property when the script is being loaded.
- **Defaults**—Click to reset all options to the default settings.
 5. Click **Load**.
 6. When the load completes, review the script actions and parameter values. You can:
 - Click  to modify a row.
 - Select **Script**, then **Substitute Versions** to change a version. Select the version to be replaced and the version to replace it with and click **OK**.

- Click a column header to sort rows alphabetically.

 **Note:**


The script is still processed in the order in which it was loaded. This order is indicated by the Order column.

- Click  to filter one or more columns.
- Click  to switch between displaying properties by label or by name.

7. Select the check box in the **Process** column next to each row to be run.

 **Note:**

From the Script menu, use Select All to process all actions. You can select a subset of actions from a script and only process those actions.

8. Click  to run the Action Script.

The results of each script action are displayed in the Status and Result Value columns. The script and the results can be saved to a file using the Download option on the Script menu. See [Downloading Action Script Results](#).

Downloading Action Script Results

You can download the action script results to the following file types:

- Action Script File (TSV)
- Comma-separated file (CSV)
- Tab-separated file (TSV)
- Portable Document Format (PDF)
- Rich Text Document (RTF)
- Spreadsheet (XLS)

Action script results can be filtered before downloading. For example, you can filter the Status field using "Failure" to show only lines that failed to load.

To download action script results:

1. From **Script**, select **Download**, then select a file type.
2. Click **Save**.
3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.

Loading Action Scripts From a Transaction Log

You can load an Action Script from the transactions that are returned by a transaction log query. Action scripts created from the transaction log are often used to automate changes made manually in one version to another version.




To load an action script from a transaction log:

1. On the Home page, select **Script**.
2. From **Source Type** select **Transaction Log**.
3. On the **Source** tab, do the following:
 - Select the source version, hierarchies, and nodes.
 - Select whether to return system transactions, data transactions, or both.

 **Note:**


If you select Data Transactions, you can select a version, hierarchies and nodes.

4. On the **Filters** tab, select from these options:
 - **Levels** — Click to specify action levels to filter:
 - **Core** — Includes actions that indicate a change to version data (not metadata objects).
 - **Logged** — Includes actions that indicate system activity occurred but did not result in additions, changes, or updates to data.
 - **Loss** — Includes transaction history records that store data that has been lost due to a core action, such as deleting a node.
 - **Result** — Includes actions that are associated with, and are a result of, a Core action.
 - **Admin** — Includes actions that affect metadata objects (not version related data).
 - **Actions** — Click to specify transaction log action types to filter.
 - **Properties** — Click to specify a list of properties to filter.
 - **Users** — Click to specify users to filter.
 - **Include Child Nodes** — Select to include child nodes for nodes returned by the query.
 - **Include Shared Nodes** — Select to include shared nodes for nodes returned by the query.
 - **From Date** — Select and enter a start date.
 - **To Date** — Select and enter an end date.
 - **Filter to Current Session** — Select to filter transactions to those performed during the current session.

- **From Transaction** — Select and enter a start transaction ID.
 - **To Transaction** — Select and enter an end transaction ID.
 - **Max Records** — Enter a value to limit the number of transactions displayed.
5. Click **Load**.
 6. When the load completes, review the script actions and parameter values. You can:
 - Click  to modify a row.
 - Select **Script**, then **Substitute Versions** to change a version. Select the version to be replaced and the version to replace it with and click **OK**.
 - Click a column header to sort rows alphabetically.
 - Click  to filter one or more columns.
 - Click  to switch between displaying properties by label or by name.
 7. Select the check box in the **Process** column next to each row to run.

 **Note:**


From the Script menu, use Select All to process all actions. You can select a subset of actions from a script and only process those actions.

8. Click  to run the Action Script.
The results of each script action are displayed in the Status and Result Value columns. The script and the results can be saved to a file using the Download option on the Script menu.

Loading Action Scripts From a Node Model

You can load an Action Script from the model of an existing node. This type of script allows you to create a new node, insert it into multiple hierarchies, and define property values based on the relationships and attributes of the existing node.

To load an action script from a node model:


1. On the Home page, select **Script**.
2. From **Source Type** select **Node Model**.
3. On the **Source** tab, do the following:
 - a. Select the source version.
 - b. Do one of the following to select a node:
 - Click  select a node, and click **OK**.
 - Type a node name in **Node To Model** and click **Fetch**.



 **Note:**

Fetch also retrieves property values for the selected node.

- c. Use the check boxes to select instances of nodes to include in the Action Script.
4. On the **Properties** tab, from **Copy Properties**, select the properties to be copied from the node model.
 - **None**—No properties are copied.
 - **Overridden**—Only overridden property values are copied.
 - **Selected**—Only the properties you select are copied.


 **Note:**

For **Selected**, click  to display a property selector from which to select properties.

- **All**—all properties are copied.
5. Do the following:
 - Select the **Global Properties** tab to view the Global property values that will be included in the Action Script based on your selection in Step 4.
 - Select the **Local Properties** tab and select a hierarchy to view the local property values that will be included from each hierarchy based on your selection in Step 4. Make changes as necessary.
 6. On the **Target** tab, do the following:
 - Select the target version for the new node.
 - Enter a name for the new node.
 - Enter a description for the new node.
 7. Click **Load**.
 8. When the load completes, review the script actions and parameter values. You can:
 - Click  to modify a row.
 - Select **Script**, then **Substitute Versions** to change a version. Select the version to be replaced and the version to replace it with and click **OK**.
 - Click a column header to sort rows alphabetically.
 - Click  to filter one or more columns.
 9. Select the check box in the **Process** column next to each row to be run.

 **Note:**

From the Script menu, use Select All to process all actions. You can select a subset of actions from a script and only process those actions.

10. Click  to run the Action Script.

The results of each script action are displayed in the Status and Result Value columns. The script and the results can be saved to a file using the Download option on the Script menu.

12

Working with Imports

Using Oracle Data Relationship Management, you can import data extracted from external systems, relational database tables and views, or manually created by users using a multi-section, text file format. Imports are always performed on new, empty versions created as part of the import process. This approach allows you to verify that data was imported correctly and resolve any issues before saving the data to the database or merging with data in other versions. After you verify the data, you can use the blender to combine imported data with data originating from other sources.

Caution:

When running imports, the regional settings of your client session are used to define the expected format for date and time values that are being imported. If these settings are not detected, then the regional settings for the server are used. For example, if the regional setting for your session specifies that date should be formatted as dd.mm.yyyy, then the data given to the import should be in that format. Otherwise, the information does not display or displays incorrectly.

Imports can be customized and saved by users with the roles: Application Administrator, Data Manager, and Data Creator.

Import Sections

An import consists of up to five sections. Each import section corresponds to a set of source data of a specific granularity. All records for an import section are expected to be the same number of columns. Import sections in source files are identified with a unique section header composed of a prefix, label, and suffix. An import file can consist of multiple sections of the same type but all sections of the same type must have the same number of columns. For database sources, each import section is mapped to a single database table or view.

Table 12-1 Import Sections

Import Section	Description
Version	Property values of the new version This import section is only required to allow for the name or description of the new version created by the import to be different for each run or to load version property values. <i>Required column: Version Name</i>

Table 12-1 (Cont.) Import Sections

Import Section	Description
Hierarchy	Hierarchies to be created by the import process, their top nodes, and property values for each hierarchy This section is required to relate nodes to each other using hierarchy relationships. <i>Required columns: Hierarchy Name, Top Node</i>
Node	Nodes to be created by the import process and global node property values for each node Nodes that are not related to hierarchies will be imported into the new version as orphans. <i>Required column: Node Name</i>
Relationship	Parent-child relationships between nodes in hierarchies and global or local node property values for the child node This section is required to relate nodes using hierarchy relationships. Nodes that are directly or indirectly related to a top node of a hierarchy being imported will be inserted into the hierarchy unless the insert would result in duplicate instances of the node. <i>Required columns: Parent Node Name, Node Name</i>
Hierarchy Node	Local node property values specific to a specified hierarchy The Hierarchy import section is required when this section is being used. <i>Required columns: Hierarchy Name, Node Name</i>

Example 12-1 Import File Example

```
[version]
My Version
[hier]
Balance Sheet,BALSHEET
[node]
1000,Assets
2000,Liabilities
3000,Equity
[relation]
BALSHEET,1000
BALSHEET,2000
BALSHEET,3000
```

Preserving Quoted Strings and Formatting

The import process can preserve quoted strings and tab and carriage return formatting. In the import file, for any section where you want to preserve the quoting and formatting, enclose the entire section in quotes. For quoted items within the section, you must provide an extra set of quotes. Quoted string processing is enabled

by the Strip Quoted Strings" parameter in the import. This option must be selected to enable the processing and stripping of quoted strings.

For example, if you want to preserve the formatting and quoted strings in this section:

```
IF (@ISMBR("Plan") AND @ISMBR(&NextYear))
    "Units"->"P_TP1"->"YearTotal" / "Avg Order Size" / "Close Rate" * "Travel
    %";
ELSEIF (@ISMBR("Forecast") AND @ISMBR(&NextYear))
    "Units"->"P_TP1"->"Rolling" / "Avg Order Size" / "Close Rate" * "Travel %";
ENDIF
```

You would add quotes like this:

```
"IF (@ISMBR("Plan") AND @ISMBR(&NextYear))
    ""Units""->"P_TP1"->"YearTotal" / ""Avg Order Size"" / ""Close Rate"" *
    ""Travel %"";
ELSEIF (@ISMBR("Forecast") AND @ISMBR(&NextYear))
    ""Units""->"P_TP1"->"Rolling" / ""Avg Order Size"" / ""Close Rate"" *
    ""Travel %"";
ENDIF"
```



Note:

If the section uses single quotes, add an extra set of single quotes to those items.

Using Substitution Parameters

Substitution parameters can be defined for an import profile to pass in user-defined values to the import process at runtime. The following import options allow the use of substitution parameters:

- Orphan Hierarchy Name
- Orphan Hierarchy Description
- Orphan Hierarchy Top Node Name
- Orphan Hierarchy Top Node Description
- Orphan Hierarchy Unique Name Prefix
- Skip Nodes During Import



Substitution parameter formatting is <%ParamKey%>, where <% and %> denote a substitution parameter and ParamKey is the name of the parameter to be used for substitution. For example, <%TopNode%>.

Creating Imports

The import wizard is used to define the parameters and criteria for the import. The wizard content is organized across Source, Style, Filters, Columns, and Target tabs to guide input.

Although the wizard tabs are ordered to gather input in a sequence, you can navigate between any tabs.

To create an import:

1. On the Home page, select **Import**.
2. Click .
3. **Optional:** Create substitution parameters for the import to pass in user-defined values to the query filter at run time.
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
4. On the **Source** tab, select the import source from the **Device** list.
 - If you selected **Client File**, click **Browse** and navigate to and select the import file.


 **Note:**

The client file is not saved with the import and will need to be re-selected for subsequent runs.

- If you selected **Server File**, select the connection and a file name.

 **Note:**

A server file can come from a network file system or FTP directory.

- If you selected **Database**, do the following:
 - a. From **Connection**, select the external connection to the database.
 - b. Click  to test the connection.
 - c. For each import section, select the database table or view from which to import.
 - d. **Optional:** Select **Clear Imported Records After Processing** to clear database records which were loaded from the source connection when the import successfully completes.

 **Note:**

The Clear Imported Records After Processing option applies to the source tables only. It does not apply to the control tables used for filtering purposes.

- e. Go to step 6.
5. Select from the following **Import Sections** options:
 - **Prefix Delimiter** — A character that is displayed at the beginning of a section heading.
 - **Suffix Delimiter** — A character that is displayed at the end of a section heading. The delimiter suffix is optional.
 - Select section headings that are in the import source and enter text for the section heading.

 **Note:**

All import sections are selected by default. Deselect sections that are not in the import source.

6. Select **File Format** options:
 - **Character Encoding** — Select one of the encoding options.
 - **Strip Quoted Strings** — Select to process quoted strings in the import file and to remove quotation marks during data import.

 **Note:**

Single (') and double quotation marks (") are removed. If both types of quotation marks are used at the same time, only the outer set of quotations is stripped.

- **Fixed Width or Delimited** — For Delimited, enter the delimiter character. For Fixed Width, enter fixed width column values in Columns Options of the **Columns** tab.
 - For client or server file: **Skip First Rows in File** — Enter the number of rows to skip from the beginning of the file.
 - For client or server file: **No Section Header in File** — Select to import one section only. Available only if one import section is selected. Select if the source file does not contain a section header.
7. Select the **Style** tab and choose options from these sections:
 - **Node Options**
 - **Determine Leaf Nodes at the End of Import Process**—Select to specify that, at the end of the import process, the import automatically sets the leaf property to True for any node that has no children.

 **Note:**

This option requires a second pass of the imported data and may require additional time to process. If this option is not selected, all nodes will be imported as limb nodes unless the Leaf property is specified as a column in the Node or Relationship section of the imported data.

- **Enforce Invalid Node Name Characters**—Select to validate node name based on the InvName system preference.
- **Relationship Sorting—Sort Relationships**—Select to enable the import to sort the nodes based on the order in which they are displayed in the relationship section of the source file or database table, and complete both of the following actions:
 - **Sort Property to Populate**—Select the node property in which to store the sort order.
 - **Hierarchy to be Sorted** — Enter the hierarchies to which to apply the sort. Separate hierarchies with commas or enter them on separate lines.

 **Note:**

All hierarchies are sorted by default unless specified here.

- **Hierarchy Options**—Specify the following information for all hierarchies created during the import:
 - **Node Type Property to Assign**—Select a node type property which will be assigned to the hierarchy node type property of every hierarchy created from the import.
 - **Validations to Assign**—Select validations to assign to hierarchies created from the import.
 - **Duplicate Delimiter**—Enter the character used to delimit base hierarchy names from their unique qualifier names when duplicates are encountered during the import.
- **Domain Options**—Specify how domain names are qualified in the import by selecting an option:
 - **None** — Domain names are not used
 - **Select a Domain** — Qualifies all nodes in the domain

 **Note:**

If you select this option, select a domain name from the drop-down list.

- **Select Import Column** — Qualifies nodes based on the selected columns. See steps 9-11.

 **Note:**

Qualifying node names appends the qualifier text of the specified domain; it does not assign the node to a domain.

- **Duplicate Handling**

- **Append Unique Text to the End of the Node Name** — Select to append the unique portion of the name to the end of the original node name.

 **Note:**

If you select this option, use the Unique String, Delimiter, Unique ID Seed, and Zero Pad Length parameters to create a naming convention for duplicate nodes. The unique portion of the name is added to the end of the original node name. If not selected, the unique portion of the name is prefixed to the beginning. For example, 001NodeA, 002NodeA.

- **Unique String** — Enter a string to be used to form part of the new unique node name.
- **Delimiter** — Enter a delimiter character to be used to separate the original node name from the unique portion of the name.

- **Unique ID Seed** — Select a starting point for this ID.

For example, if NodeA occurs three times in a hierarchy and the unique ID seed is 1, the original node is not changed, and the other node names are replaced with NodeA1 and NodeA2. This example ignores the other Duplicate settings in order to illustrate the use of unique ID seed.

- **Zero-Pad Length** — Enter the minimum number of digits required in the unique ID portion of the duplicate name.

For example, a value of 3 for zero-pad length creates duplicate nodes named NodeA001, NodeA002, and so on. A value of 2 creates duplicate nodes named NodeA01, NodeA02, and so on.

- **Associated Property** — Select a property to enable duplicate nodes to be grouped so they can be easily queried after the import is complete.

If an associated property is set, the property is populated with the node name of the original node in all duplicate nodes. This option must refer to a global, associated group (AscGroup) property.

- **Duplicate Error Mode** — Select a mode to determine the severity with which the import processor treats a duplicate:

- * **Generate Warning** — Warning message is added to the log for each duplicate.

- * **Generate Error** — Error message is added to the log for each duplicate.

 **Note:**

Processing stops if the number of errors exceeds the Max Errors value (on Target tab).

- **Orphan Options**—Select how orphans generated by the import are handled.

 **Note:**

Two types of orphans are used when creating hierarchies from orphan nodes:

Orphans that are not a child of any other orphan node and have no children (stranded orphans)

Orphans that are not a child of any other orphan node and have children (top node orphans)

- **Create Base Orphan Hierarchy**—Select to create a hierarchy of stranded orphans and enter these parameters for the hierarchy:
 - * **Name**—Enter the name for the orphan hierarchy.
 - * **Description—Optional:** Enter a description for the orphan hierarchy
 - * **Top Node Name**—Enter the name for the top node of the orphan hierarchy.
 - * **Top Node Description—Optional:** Enter a description for the top node of the orphan hierarchy.
 - **Create Orphan Branch Hierarchies**—Select to create orphan branch hierarchies from top node orphans and enter these settings:
 - * **Max # To Create**—Enter the maximum number of orphan hierarchies to create.
 - * **Unique Name Prefix—Optional:** Enter a prefix which will be appended to the hierarchy name.
 - * **Hierarchy Property for Parent**—Enter a valid hierarchy property. Select a hierarchy level property which will be populated with the value in the Parent field in the Relationship import section if defined. This option is used with the Skip Nodes During Import parameter to create multiple hierarchies from one larger hierarchy. This property can then be used by the export to output a Parent value for top nodes of those hierarchies.
8. On the **Columns** tab, from the **Section** list, select an import section.
 9. On the **Select Columns** tab, select the properties to be populated by the import for the import and move them from **Available** to **Selected**.

 **Note:**

Use the Category list to select a property category to ease navigation.

A special column called [Ignore Column] is available as a placeholder for any column in the input source that should not be processed during the import.



 **Tip:**

For client and server file imports, use the arrows to the right of the Selected list to position the column based on the order in which the column exists in the source file. The first item in the list is the first column, the second item in the list is the second column, and so on.

10. On the **Column Options** tab, you can map a column to hierarchies. Double-click a column name row and select from these options:
 - Select **Apply Value to All Hiers** if values in the column should be applied to all hierarchies. By default, this option is selected.
 - In the **Hierarchy Listing** field, enter the name of each hierarchy to which the column applies. The property value is set only in the hierarchies designated in the list.




 **Note:**

The listed hierarchies must match the names of hierarchies specified in the import source.

- If you selected **Fixed Width** on the **Source** tab, enter a value in the **Fixed Width** column.
11. If you selected the **Select Import Column** domain name qualification method in step 7, the **Domain Options** tab is displayed. You can configure these sections:
 - Hierarchy — Top node column
 - Node — Node column
 - Relation — Child and Parent columns
 - Hierarchy Node — Node column
 12. If you are importing from database tables, the **Database Options** tab is displayed. For each import section, the following columns are displayed:
 - **Column**—Displays the name of the Oracle Data Relationship Management property being populated.
 - **Source Field**—Click  in the **Action** column to select a field from the source database table for the section. Click  to save the changes.




 **Note:**




A source field may be mapped to more than one Data Relationship Management property.

- **Field Type**—Displays the data type of the field from the source database table.
 - **Length**—Displays the length of the field from the source database table (if applicable).
 - **Order**—Click  in the **Action** column and select a field from the source database table used to order the data before import. Click   to save the changes.
13. On the **Filters** tab, do any of the following:
- Skip blank values for certain property types
 - Skip default values for certain property types
 - Skip Nodes During Import—These nodes are ignored during the import process.

 **Note:**

It is recommended to skip blank and use default values to minimize the size of the imported version.

14. If you are importing data from database tables, the following database filters are available:
- **None**—No filtering of database records occurs during import.
 - **Control Table**—Specifies the use of a source database table for controlling the filter. Records existing in the control table will determine which records from the import section tables will be imported.
 - a. From **Filter Table**, select the source database table used for filtering.
 - b. From **Filter Key Field**, select the field from the specified filter table used for filtering data from the import tables.
 - c. In the **Filter Join Fields** section, click  and select a field for each import section table to be used for filtering. Click   to save the changes.
 - **Inclusion List**—Specifies literal values used for filtering records from import section tables.
 - a. In the **Inclusion List** section, click **Add** and enter the literal values to be used as a filter for the tables for all import sections that have a Filter Join Field specified.

- b. In the **Filter Join Fields** section, click  and select a field for each import section table to be used for filtering. Click  to save the changes.
15. On the **Target** tab, enter a name for the new version that will be created by the import.
16. Make selections from these options:
 - **Max Errors**—Select a value for the maximum number of errors that can occur during processing before the import stops.
 - **Assign to Variable**—Select a version variable to which to assign the imported version.
 - **Save Version To Repository**—Select to save the new version to the repository.
17. Click  to save the import.



You are prompted to define a name, description, and an object access group for the import. Select a custom group or one of the following:

- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager and Application Administrator users can create and modify standard imports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

Running, Copying, Moving, and Deleting Imports

You can run, copy, move, and delete imports. You must open an import before you can run it.

To run an import:

1. On the Home page, select **Import** , and then select an import to run.
2. Click .
3. Click the tabs to modify import parameters, and then click  to run the import.

To copy an import:

1. On the Home page, select **Import** , and then select an import to copy.
2. From **Imports**, select **Copy**.
3. Enter a new name and description for the import.
4. From **Object Access Group**, select a custom group or one of the following:
 - **User**—Personal objects that are available only to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard imports.
 - **System**—Restricted objects that are available only to Application Administrator role users to view and run.
5. Click **OK**.

To move an import:

1. On the Home page, select **Import**, and then select an import to move.
2. From **Imports**, select **Move**.
3. From **Object Access Group**, select a different group and click **OK**.

To delete an import:

1. On the Home page, select **Import**, and then select an import to delete.
2. From **Imports**, select **Delete**.

Viewing Import Results

After an import is run, results are available at the bottom of the screen.

To download import results:

1. From **Download**, select a file type:
 - Download as CSV
 - Download as PDF
 - Download as RTF
 - Download as XLS
2. Click **Save**.
3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.

13

Blending Versions

Oracle Data Relationship Management enables you to combine elements of two different versions into the same version.

With Blenders, you can perform these actions:

- Process adds, moves, or deletes to an existing hierarchy
- Process activations and promotions (leaf to limb)
- Act upon any combination of hierarchies
- Create new hierarchies
- Map top nodes from a source hierarchy to nodes in a target hierarchy.


Blenders can be customized and saved by users with the roles: Application Administrator, Data Manager, and Data Creator.

Creating Blenders

When you create or open a blender, it is displayed on a named tab.

The blender wizard is used to define the parameters and criteria for the blender. The wizard content is organized across Source, Style, Filters, Properties, and Target tabs to guide input. Although the wizard tabs are ordered to gather input in a sequence, you can navigate between any tabs.

To create a blender:

1. On the Home page, select **Blend**.
2. Click .
3. On the **Source** tab, select source and target versions or version variables for the blender.
4. Perform an action:
 - Select **Blend all Hierarchies** to blend all hierarchies in the selected versions.

Note:

If you select this option, the **Use Profile Defaults For All Hierarchy Sets** option is required.

- Deselect **Blend all Hierarchies** and click **New**.
 - a. On the **Source** tab, select the hierarchy and top node mappings to blend.
 - b. On the **Target** tab, do one of the following:

- To map the hierarchy and top node from the source version to the same hierarchy and top node in the target version, select **Specified** from the drop-down list, and then select **Use Same as Source**
 - To map the hierarchy and top node from the source version to a different hierarchy and top node in the target version, select **Selected** from the drop-down list, and then select the hierarchy and top node in the target version.
- c. **Optional:** On the **Domain Options** tab, select a domain property.
 - d. Click **OK**.

 **Note:**

You can select **Use Profile Defaults For All Hierarchy Sets** to use default values for all hierarchy/top node sets. You set the profile defaults for properties on the **Properties** tab. If you deselect this option, you have the option of choosing which hierarchies should use the defaults and which should not when you select hierarchy/top node pairs.

5. On the **Style** tab, do the following:
 - Select from the following structure options:
 - **Process Structure** — To blend structural elements, such as nodes and hierarchy relationships.
 - **Allow Hierarchy Creation** — To create hierarchies in the target version that exist only in the source version.

 **Note:**

If this option is not selected and there are hierarchies that exist in the source but not in the target, then an error occurs.

- **Allow Leaf Promotion** — to promote a node with children in the source to a limb in the target.

 **Note:**

If a node in the source has children, but its corresponding node in the target is a leaf node (an outermost node with no children allowed), then this option enables the Blender to promote the node to limb status. The child nodes from the source are then added to the limb node in the target.

- Select from the following property options:
 - **Process Properties** — To blend property values
 - **Disable Realtime Validations** — To turn off the real-time validations on the target version prior to blending. This setting only applies if the target

version is detached. If the target version is saved (not detached) then validations are still enabled regardless of this setting.

 **Note:**

Real-time validations can prevent changes from being propagated to the target.

- **Process Version Validation Assignments** — To process version-level validation assignments from the source to the target
- In the **Domain Options** section, specify how domain names are assigned by selecting an option:
 - **None**—Domains names are not used.
 - **Select a Domain**—Specify a single domain to assign to all nodes.

 **Note:**

If you select this option, select the domain to assign to all nodes.

- **Select a Domain Property**—Select a domain holding property from the drop down list.
 - **Optional:** Select **Assign Domains to Target Version**.
6. On the **Filter** tab, select from these options on the Profile Defaults tab or the Hierarchy Overrides tab.

 **Note:**

If you selected Use Profile Defaults for All Hierarchy Sets in step 3, make selections on the Profile Defaults tab. If any hierarchy sets exist and any are set to NOT use defaults, then the Hierarchy Overrides tab is enabled. Provide overrides for each hierarchy set on this tab.

- **Process Inserts**—To insert any nodes existing only in the source hierarchy to the target hierarchy.
- **Process Removes**—To remove any nodes existing only in the target hierarchy according to the setting in the Remove Mode list and the Remove Indicator list.

If you selected **Process Removes**, select one of the following Remove Modes:

- **Remove**—Node is immediately removed during blending
- **Mark**—Node is marked for later removal

When the Remove Mode is set to Mark, the Remove Indicator option designates which property contains the value showing that the node is flagged to be removed. The remove indicator property must be created as a local, Boolean property. For information on creating properties, see the *Oracle Data Relationship Management Administrator's Guide*.

- **Process Moves**—To move any node which has a different parent in the source to that parent in the target.
 - **Process Activations**—To process node activations (inactivate and reactivate) from the target to the source.
 - **Process Hierarchy Validation Assignments**—To process the hierarchy level validation assignments from the source to the target.
7. On the **Properties** tab, from **Property Selection**, select one of the following ways to blend properties:
 - **All Excluding Val/Access**—All properties are blended; excludes properties used for node-level validations and for node access group leaf and limb-level settings.
 - **None**—No properties are blended.
 - **Selected**—The selected properties are blended.
 8. From **Property Propagate Mode**, select one of the following modes to determine how the property is populated in the target.

 **Note:**

If you selected Use Profile Defaults for All Hierarchy Sets in step 3, make selections on the Profile Defaults tab. If you did not select to use profile defaults, make selections on the Hierarchy Overrides tab.

- **Force**—The property is assigned for every node in the target, whether that property is defined at that node, is inherited, or derived.
 - **Difference**—Populates the property for any node where the value is different, regardless of how that property is populated in the target.
 - **Defined**—The property is blended only when it is specifically defined for the node in the source.
9. **Optional:** Select **Propagate Property Locks** if you want inheritance locks in the source to be blended to the target.
 10. On the Target tab, select the Target version:
 - **Use Selected Target**—To blend changes into the selected target version.
 - **Copy to New Version**—To blend changes into a copy of the selected target version.

 **Note:**

If you select this option, you must specify a name and description for the new version.

11. Enter a value for **Max Iterations**.

 **Note:**


This value determines the maximum number of attempts the Blender performs in reconciling the Source and Target versions. This prevents the Blender from running indefinitely in the case of irresolvable conflicts. The recommended value is 3.

12. Enter a value for **Max Errors**. **Note:**

This value sets the maximum number of errors that can occur during processing before the Blender stops without completion.

13. Make selections for the following blender status properties options:

- **Clear Values**—to set all Blender Status Property values to False during processing.
- **Populate Values**—to set each Blender Status Property to True during processing.
- **Version Blended**—The name of the property that will be set to TRUE if the version has been blended. This must be a defined, boolean, version level property.
- **Node Moved**—The name of the property that will be set to TRUE if the node was moved by the blend. This must be a defined, boolean, local node level property.
- **Node Created**—The name of the property that will be set to TRUE if the node was created by the blend. This must be a defined, boolean, global node level property.
- **Local Node Changed**—The name of the property that will be set to TRUE if the node had local properties changed by the blend. This must be a defined, boolean, local node level property.
- **Node Assigned to Domain**—The name of the property that will be set to TRUE if the node has a domain assigned.
- **Hierarchy Blended**—The name of the property that will be set to TRUE if the hierarchy has been blended. This must be a defined, boolean, hierarchy level property.
- **Global Node Changed**—The name of the property that will be set to TRUE if the node had global properties changed by the blend. This must be a defined, boolean, global node level property.
- **Node Inserted**—The name of the property that will be set to TRUE if the node was inserted by the blend. This must be a defined, boolean, node level property.
- **Leaf Promoted**—The name of the property that will be set to TRUE if the node was promoted from a leaf to a limb by the blend. This must be a defined, boolean, global node level property.

14. Click  to save the blender.

You are prompted to define a name, description, and an object access group for the blender. Select a custom group or one of the following:



- **User**—Personal objects that are only available to an individual user to view and run.

- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard blenders.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

Running, Copying, Moving, and Deleting Blenders

You can run, copy, move, and delete saved blenders. You must open a blender before you can run it.

To run a blender:

1. On the Home page, select **Blend** and then select a blender to run.
2. Click .
3. Click the tabs to modify the blender parameters, and then click  to run the blender.

To copy a blender:

1. On the Home page, select **Blend** and then select a blender to copy.
2. From **Blenders**, select **Copy**.
3. Enter a new name and description for the blender.
4. From **Object Access Group**, select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard blenders.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
5. Click **OK**.

To move a blender:

1. On the Home page, select **Blend**, and then select a blender to move.
2. From **Blenders**, select **Move**.
3. From **Object Access Group**, select a different group and click **OK**.

To delete a blender:

1. On the Home page, select **Blend** and then select a blender to delete.
2. From **Blenders**, select **Delete**.

Viewing Blender Results

After a blender is run, results are available at the bottom of the screen.

To download blender results:

1. From **Download**, select a file type:

- Download as CSV
 - Download as PDF
 - Download as RTF
 - Download as XLS
2. Click **Save**.
 3. Browse to the directory where you want to download the document, enter a name for the document, and click **Save**.

14

Working with Exports

Oracle Data Relationship Management allows you to export information in a variety of formats to meet the different needs of each system and user. Multiple exports can be grouped and run together using Books. From the Export task on the Home page, you can create, open, and manage exports and books. You can open multiple exports in separate tabs, but only one export can be in focus at a time. You can also run exports from a selected data object being browsed.

Creating Exports

Oracle Data Relationship Management provides several export types to output data in different formats.

The following table lists the export types for Data Relationship Management.

Exports	Export Type	Description
Hierarchy Exports	Hierarchy	Outputs nodes and properties from hierarchies. This export can be used to create a parent-child format or a basic list of hierarchy nodes. Usage Frequency: High Output Formats: File, Database table
	Hierarchy XML	Outputs nodes, their relationships, and properties from hierarchies in an XML format. Usage Frequency: High Output Formats: File
	Generation	Creates a generation- or level-based format for nodes in hierarchies. Outputs a record for each node, its ancestor nodes, and their properties. This export type can be used to create a format where hierarchy relationships are represented as columns. Usage Frequency: Medium Output Formats: File

Exports	Export Type	Description
Comparison Exports	Compare	Compares two hierarchies and outputs a record for each node that meets the comparison criteria. This export can be used to provide incremental changes to target systems. Usage Frequency: Medium Output Formats: File, Database table
	Difference	Compares two hierarchies and outputs a record for each difference found. This export can be used for reporting incremental changes to users. Usage Frequency: Low Output Formats: File
Integration Exports	EPM Architect	Outputs hierarchies, nodes, and properties from Data Relationship Management which can be shared with and deployed to multiple EPM applications, such as Oracle Hyperion Financial Management and Oracle Hyperion Planning. Usage Frequency: Only when integrating with Oracle Hyperion EPM Architect Output Formats: File
Version Exports	Version	Outputs a record for each node and its properties for an entire version. Hierarchy relationships and local properties are not available for this export type. Usage Frequency: Medium Output Formats: File, Database table
Property Exports	Property List	Outputs a record for each item in the list of values for a property definition. This is a special use case export type. Usage Frequency: Low Output Formats: File
	Property Lookup	Outputs a record for each key-value pair in the lookup table for a property definition. This is a special use case export type. Usage Frequency: Low Output Formats: File

Exports	Export Type	Description
Log Exports	Transaction Log	Outputs a record for each item in the transaction log that met the filter criteria. This export can be used for audit purposes. Usage Frequency: Medium Output Formats: File, Database table
	Merge Log	Outputs a record for each item in the merge log that met the filter criteria. This export is applicable only if the Merge feature is used. Usage Frequency: Low Output Formats: File

Using Substitution Parameters

Substitution parameters can be added to an export profile to pass in user-defined values to the export process at runtime. The following export types offer the use of substitution parameters:

- Hierarchy
- Generation
- Compare
- Hierarchy XML
- Version
- EPM Architect

The following export options allow the use of substitution parameters:

- Hierarchy group selection
- Export query filter criteria
- Dynamic export columns
- Header/footer

Substitution parameter formatting is <%ParamKey%>, where <% and %> denote a substitution parameter and ParamKey is the name of the parameter to be used for substitution. For example, <%TopNode%>.

Export Specific Columns

Some export types allow you to designate specific export columns.

Table 14-1 Export Specific Columns

Export Column	Description	Export Types
[Action Type Code]	Text label for the action being compared in the current export record. On the Style tab, enter an optional custom label for any of these options Additions, Deletions, Merges, Moves, Renames or Property Updates .	Compare
[Calculated Level{0}] [Calculated Level{1}]	Level of the current node from the selected top node for the export, starting at {0, 1}.	Hierarchy, Hierarchy XML
[Calculated Parent]	Parent of the current node, or nearest ancestor included in the export if an exclusion query filter is used. Default is None.	Hierarchy, Hierarchy XML
[Dynamic Column]	Adds a specific value to a column that is not present in the export data. You can specify a combination of constant values and substitution variables in the dynamic column. On the Columns tab, select Export Specific from the Category list and then select [Dynamic Column] and add it to the Selected list. Enter a name for the dynamic column. On the Column Options tab, enter the value for the export column.	Hierarchy, Generation, Compare, Version
[Merged Node]	Name of the merge node selected for a node that was deleted. This value is blank if the node was not merged or deleted or if merge functionality is not enabled for the application.	Compare
[Node Sequence #]	Outputs a sequence number for a node's position in its ancestor list, with leaf nodes starting at zero (0).	Generation
[Property Label]	Outputs the property display label for property comparison records.	Compare
[Property Value]	Outputs the To Version property value for property comparison records.	Compare



Table 14-1 (Cont.) Export Specific Columns

Export Column	Description	Export Types
[Record ID]	<p>Preserves the export order when exporting to a database table. This column provides an auto-generated numeric value, starting with 1, that increases sequentially for each row in the export.</p> <p>On the Columns tab, select Export Specific from the Category list and then select [Record Id] and add it to the Selected list.</p>	Hierarchy, Hierarchy XML, Compare, Version

Hierarchy Export

Hierarchy exports output nodes and properties from hierarchies. This information can be output to a file or a database table. This export type can be used to create a parent-child format or a basic list of hierarchy nodes.

To create a hierarchy export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Hierarchy** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

Note:

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
6. From **Hierarchy Selection**, select an option:
 - **Top Nodes**—Click **Add**, select a hierarchy, and select a node and then click **OK**. Repeat until all nodes are added.

 **Note:**

Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.

- **Hierarchy Group**—Select a hierarchy group property and a hierarchy group.


 **Note:**

You can use substitution variables for the hierarchy group property and the hierarchy group.

- **Hierarchy Group Property**—Select a hierarchy group property or select **[Get Value from Parameter]** and then select the parameter to use for the property.
 - **Hierarchy Group**—Select a hierarchy group or select **[Get Value from Parameter]** and then select the parameter to use for the group.
7. On the **Style** tab, select from these options:
- **Node Selection** — Determines which nodes (All Nodes, Limb Nodes Only, Leaf Nodes Only) are included in the export
 - **Recurse from Top Node** — Select to include output for all nodes below the top node. If cleared, only the top node is included. This option is useful when exporting a single record for each hierarchy selected.
 - **Include Inactive Nodes** — Select to include inactive nodes in the export.
 - **Include Implicitly Shared Nodes** — Select to include descendant nodes below shared limb nodes that are explicitly shared. Implicit shared nodes are exported only if not filtered out by another option.
 - If a query filter is used in the export, the administrator must explicitly include the shared nodes. For example, the filter (#Children=0) must be changed to ((#Children=0 OR (Shared Node Flag = True)).
 - For the implicitly shared node, Parent properties reflect the values of the actual node (the primary node). For example, if your export includes the Parent Node property for an implicitly shared node, it will show the parent as the primary node.
 - **Remove Duplicates Based on Key** — Select to use the primary key (defined on the Column Options tab) to suppress duplicate records from being exported.
 - **Tabbed Output** — Select to indent the nodes for each level of a hierarchy using tab characters to produce a structured output format.
8. In the **Batch Validations** section, select the type of validations to run.


 **Note:**

Batch validations are performed only for the hierarchies and nodes included in the export. Validations are run before the export. If errors occur, the export stops.

- **None** — No validations run.
 - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export run.
 - **Selected** — Click  to select validations to run.
9. In the **Repeat Parameters** section, select from these options:
- **Repeat Number Property** — References a local, node-level, integer property that repeats a node within an export structure. For example, given an A-B-C parent-child hierarchy, if a Repeat Num Property has a value of 2 for node B and a value of 1 for nodes A and C, then the export looks like A,B,B,C (node B is repeated twice).

 **Note:**

When using Repeat Number Property, the value must be set to at least 1 for nodes to be included in the export results. It is recommended to set the Repeat Number Property default value to 1.

- **Bottom Level Property** and **Bottom Level Value** — Determines the maximum depth of the export structure. This value can be defined as a hierarchy-level, integer property (Bottom Level Property) or as an absolute value (Bottom Level Value.)
If a bottom level is defined, the repeat options are enabled to automatically repeat nodes down to the defined level in the export structure. For example, for nodes A-B-C in a parent-child hierarchy, if the Bottom Level Value is defined as 6, node B has a Repeat Number Property value of 2, and the Repeat Bottom Leaf Node option is selected, then the export structure is A,B,B,C,C,C (the export structure is six levels deep, the leaf node C is repeated until level 6 is reached, and node B is repeated twice).
10. On the **Filter** tab, select from these options:
- From **Validation**, select a validation to use to filter the export.
 - From **Query**, select a query to apply to the export, or, to define a query, click .
 - **Use text file to include/exclude descendant nodes** — Select if you have created a text file with nodes to be included or excluded from the export. The file format is a list of node names with one per line. Select a connection, enter a file name, and select whether to include or exclude the nodes from the export.
11. On the **Columns** tab, select the properties to display as columns in the export results and move them from **Available** to **Selected**.
- For more information, see [Export Specific Columns](#)

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

12. On the **Column Options** tab, select from these options:

- **Pivot**—If the property contains a comma-delimited list of values, a separate record is exported for each value with all other export columns set to the same values. The Pivot option may be selected on only one column.
- **Skip Defaults**—To place a blank (empty string) or NULL (in output to database table) in the export if the field value equals the default value of the property
- **Primary Key**—To define a field (or fields) as the primary key to be used when determining whether duplicates exist. (This is used by the Remove Duplicates option on the Style tab).
- **Dynamic Value**—If you added a Dynamic Column, define the value for the column here. The value can be null or a combination of constant values and substitution variables.

13. On the **Target** tab, select the output type for the results:

- **Client File** — Download results via the browser.
- **Server File** — Use an external connection and save results to a network file system or an FTP directory.
- **Database Table** — Save results to a database table.

14. Perform an action:

- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export based on the labels of the selected properties.
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.


For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options. Encoding applies to the output file and to the node include/exclude file if specified.

- **Replace options** — Select up to three characters from the lists to be replaced. All instances in the export are replaced as specified. For example, all commas can be replaced by tabs.
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter options** — To select a field delimiter character and a record delimiter character.
- If you selected to export to a **Database Table**, you must set your connection and select the database table. You must have already created a valid external connection to a database for it to show up in the drop-down list. You can also select from these options:
 - Clear table options
 - * **Do Not Clear Table** — Appends to existing data in the specified table.
 - * **Clear Based on Version IDs** — Deletes data in the table where the specified key field equals the current version ID of the data being exported. If this option is selected, a Key field must be specified in the Database Column Options section. When the export is run, all data in the table is first deleted where the Key field equals the current version ID of the data being exported. For the other Clear Table options, the Key field is ignored.
 - * **Clear All Rows** — Deletes data from the table before writing the export output.
 - **Use Transactions** — The export is performed within a database transaction and allows the entire operation to be rolled back if an export error occurs.
 - **Use Fast Appends** — Performs a bulk insert operation into the target database table for improved export performance. Only available with a database-specific connection.
 - **Use Fast Deletes** — Performs a bulk delete operation on the target database table for improved performance. Only available with a database-specific connection.
 - **Database Column Options** — Each column being exported must be mapped to a field in the external database table. Under Field Name, select the name of a field. If you selected Clear Based on Version ID(s), specify a Key field.

 **Tip:**

The Length option can be left blank so that string lengths up to the database column declaration can be sent to the database.

 **Note:**

The column must be the same data type as the database field to which it is mapped. For example, if a string property is mapped to a numeric database field, an error occurs.

- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:


- **Column Headings** — Includes column headings
- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF(""ABC"", ""XYZ"", )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character



15. Do any of the following:


- To save the export, click .


You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User**—Personal objects that are available only to an individual user to view and run.

- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are available only to Application Administrator role users to view and run.

- To validate the export, click .
- To run the export, click .

If you selected to run validations on the **Style** tab, clicking  performs the validations. Then, if no errors occur, the export runs. If validation errors occur, then

the export stops and an error is displayed. Click  for a list of validation errors.

Customization Tags

Customization tags can also be inserted to further enhance the readability of the export results. The following table describes the customization tags that are available for use. Optional parameters are enclosed in [brackets].

Table 14-2 Customization Tags

Customization Tag	Description
<%Date[(format)]%>	Returns current date based on the user's regional settings, where format = MM/DD/YYYY
<%Time[(format)]%>	Returns current time based on the user's regional settings, where format = HH:MM:SS AMPM
<%Timestamp%>	Returns combination of date and time.
<%CurrentUser%>	Returns current Oracle Data Relationship Management user.
<%CurrentVersion%>	Returns current Data Relationship Management version for this export.
<%PreviousVersion%>	Returns previous Data Relationship Management version for this export.
<%ExportName%>	Returns name of export.
<%FileName%>	<ul style="list-style-type: none"> • Returns output file name if specified • Returns default file name if not specified
<%RecordCount%>	Returns count of records exported (only available in footer)
<%AppVersion%>	Returns the Data Relationship Management product number (for example, 11.1.2)
<%PropValue()%>	Returned value depends on the specified export type as described following:
<%PropValue([Previous Current(Default)], PropName)%>	For Compare/Difference Exports, returns value of first node in CompareCriteria for Previous/Current version



Table 14-2 (Cont.) Customization Tags

Customization Tag	Description
<%PropValue([HierName], PropName)%>	<ul style="list-style-type: none"> For Hierarchy Exports, returns value from first node in list If HierName is supplied, return value from the first node that matches HierName
<%PropValue(PropName)%>	For Version Exports, returns only version properties
<%RunTimeParam(ParamName)%>	Used with these export types: Version, Hierarchy, Generation, Compare, Difference, Property Lookup, Transaction Log, and Merge Log - where ParamName defines the name of the parameter (run time and/or default parameters). The value of the substitution variable is placed in the export header or footer when the tag is processed.

Hierarchy XML Export

Hierarchy XML exports output nodes, their relationships, and properties from hierarchies in an XML format. This information can be output to a file.

To create a hierarchy XML export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Hierarchy XML** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

Note:

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.
You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
6. From **Hierarchy Selection**, select an option:
 - **Top Nodes**—Click **Add**, select a hierarchy, and select a node and then click **OK**. Repeat until all nodes are added.

 **Note:**

Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.

- **Hierarchy Group**—Select a hierarchy group property and a hierarchy group

 **Note:**

You can use substitution variables for the hierarchy group property and the hierarchy group.

- **Hierarchy Group Property**—Select a hierarchy group property or select [**Get Value from Parameter**] and then select the parameter to use for the property.
 - **Hierarchy Group**—Select a hierarchy group or select [**Get Value from Parameter**] and then select the parameter to use for the group.
7. On the **Style** tab, select from these options:
 - **Node Selection** — Determines which nodes (All Nodes, Limb Nodes Only, Leaf Nodes Only) are included in the export
 - **Recurse from Top Node** — Select to include output for all nodes below the top node. If cleared, only the top node is included in the export. This option is useful when exporting a single record for each hierarchy selected.
 - **Include Inactive Nodes** — Select to include inactive nodes in the export.
 8. In the **Batch Validations** section, select the type of validations to run.

 **Note:**

Batch validations are performed only for the hierarchies and nodes included in the export. Validations are run before the export, and if errors occur, the export stops.

- **None** — No validations run.
 - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export run.
 - **Selected** — Click to select validations to run.
9. On the **Filter** tab, select from these options:
 - From **Validation**, select a validation to apply to the export.
 - From **Query**, select a query to apply to the export or, to define a query, click .
 - **Use text file to include/exclude descendant nodes** — Select if you have created a text file with nodes to be included or excluded from the export. The file format is a list of node names with one per line. Select a connection, enter a file name, and select whether to include or exclude the nodes from the export.


10. On the **Columns** tab, select the properties to display as columns in the export results and move them from **Available** to **Selected**.



For more information, see [Export Specific Columns](#)



 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

11. On the **Target** tab, select the output type for the results:
 - **Client File** — Download results via the browser.
 - **Server File** — Use an external connection and save results to a network file system or an FTP directory.
 - **Database Table** — Save results to a database table.
12. Perform an action:
 - If you selected to export to a **Client File**, go to the next step.
 - If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file.
13. Do any of the following:

- To save the export, click .
You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.



- To validate the export, click .
- To run the export, click .

If you selected to run validations on the **Style** tab, clicking  performs the validations and then, if no errors occur, runs the export. If validation errors occur, then the export stops and an error is displayed. Click  for a list of validation errors.

Generation Export

Generation exports create a generation- or level-based format for nodes in hierarchies. The export outputs a record for each node, its ancestor nodes, and their properties. You can output this information to a file. You can use this export type to create a format where hierarchy relationships are represented as columns.

To create a generation export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Generation** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

Note:

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.
You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
6. From **Hierarchy Selection**, select an option:
 - **Top Nodes**—Click **Add**, select a hierarchy, and select a node and then click **OK**. Repeat until all nodes are added.

Note:

Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.

- **Hierarchy Group**—Select a hierarchy group property and a hierarchy group

Note:

You can use substitution variables for the hierarchy group property and the hierarchy group.

- **Hierarchy Group Property**—Select a hierarchy group property or select **[Get Value from Parameter]** and then select the parameter to use for the property.

- **Hierarchy Group**—Select a hierarchy group or select **[Get Value from Parameter]** and then select the parameter to use for the group.
7. On the **Style** tab, select from these options:
- **Node Selection** — Determines which nodes (All Nodes, Limb Nodes Only, Leaf Nodes Only) are included
 - **Recurse from Top Node** — Select to include output for all nodes below the top node. If this option is not selected, then only the selected top node is included. This option is useful when exporting a single record for each hierarchy selected.
 - **Include Inactive Nodes** — Select to include inactive nodes.
 - **Include Implicitly Shared Nodes** — Select to include descendant nodes below shared limb nodes that are explicitly shared. Implicit shared nodes are only exported if not filtered out by another option such as top node selection or a query filter. The structure of implicitly shared nodes below a shared parent node is reflected in the ancestor section of the hierarchy export.
 - If a query filter is used in the export, the administrator must explicitly include the shared nodes. For example, the filter (#Children=0) must be changed to ((#Children=0 OR (Shared Node Flag = True)).
 - For the implicitly shared node, the Parent properties reflect the values of the actual node (the primary node). For example, if your export includes the Parent Node property for an implicitly shared node, it will show the parent as the primary node.
 - **Root to Node** — Select to set the export order (if not selected, the order is Node to Root.) For example, if node A is a parent to node B and this option is selected, then the output is A,B. Otherwise, with this box unchecked, it is B,A.
8. In the **Batch Validations** section, select the type of validations to run.

 **Note:**

Batch validations are performed only for the hierarchies and nodes included in the export. Validations are run before the export. If errors occur, the export stops.

- **None** — No validations run.
 - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export run.
 - **Selected** — Click to select validations to run.
9. In the **Repeat Parameters** section, select from these options:
- **Repeat Number Property** — References a local, node-level, integer property that repeats a node within an export structure. For example, given an A-B-C parent-child hierarchy, if a Repeat Num Property has a value of 2 for node B and a value of 1 for nodes A and C, then the export looks like A,B,B,C (node B is repeated twice).

 **Note:**

When using Repeat Number Property, the value must be set to at least 1 for nodes to be included in the export results. It is recommended to set the Repeat Number Property default value to 1.

- **Bottom Level Property** and **Bottom Level Value** — Determines the maximum depth of the export structure. This value can be defined as a hierarchy-level, integer property (Bottom Level Property), or as an absolute value (Bottom Level Value.)

If a bottom level is defined, repeat options are enabled to automatically repeat nodes down to the defined level in the export structure. For example, for nodes A-B-C in a parent-child hierarchy, if the Bottom Level Value is defined as 6, node B has a Repeat Number Property value of 2, and the Repeat Bottom Leaf Node option is selected, then the export structure is A,B,B,C,C,C (the export structure is six levels deep, the leaf node C is repeated until level 6 is reached, and node B is repeated twice).

10. In the **Ancestor Options** section, select from these options:

- **Max Ancestor Levels** — Defines the number of ancestor levels in the export structure.
- **Use Inverted Levels** — The Oracle Essbase level methodology is used where the leaves are at 0 and count up. For a node that has multiple branches under it, the node is the largest of the possible values.
- **Blank Repeated Levels** — Prevents repeated nodes from appearing in the output. If cleared, a node may be repeated in the output to achieve the specified level.
- **Group Repeated Columns** — Repeated columns are grouped by property in the ancestor columns.

11. On the **Filter** tab, select from these options:

- From **Validation**, select a validation to apply to the export.
- From **Query**, select a query to apply to the export, or, to define a query, click

 **Note:**

The inclusion filter determines the set of nodes to include in the results. The exclusion filter restricts the nodes displayed as ancestor columns.

- **Use text file to include or exclude descendant nodes** — Select if you have created a text file with nodes to be included or excluded from the export. The file format is a list of node names with one per line. Select a connection, enter a file name, and select whether to include or exclude the nodes from the export.

12. On the **Columns** tab, select an option from the **Column Set** list:

- **Pre Columns** — Properties to export at the beginning of each export record
- **Bottom Node Columns** — Properties for the specified bottom node
- **Ancestor Columns** — Node properties for the ancestor list of the specified bottom node

- **Post Columns** — Properties to export at the end of each export record
13. Select the properties to display as columns in the export results and move them from **Available** to **Selected**.

For more information, see [Export Specific Columns](#)

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.


14. On the **Column Options** tab, select options:
- **Pivot** — If the property contains a comma-delimited list of values, a separate record is exported for each value with all other export columns set to the same values. The Pivot option may be selected on only one column.
 - **Skip Defaults** — To place a blank (empty string) or NULL (in output to database table) in the export if the field value equals the default value of the property
 - **Primary Key** — To define a field (or fields) as the primary key to be used when determining whether duplicates exist. This is also used by the Remove Duplicates option on the Style tab.
 - **Fill Value**—Used with the **Max Ancestor Levels** setting on the **Style** tab to fill the values for ancestor columns beyond the bottom level of the ancestors up to the number of ancestor levels defined in the export profile.
 - **Dynamic Value**—If you added a Dynamic Column, define the value for the column here. The value can be null or a combination of constant values and substitution variables.
15. On the **Target** tab, select the output type for the results:
- **Client File** — Download results via the browser.
 - **Server File** — Use an external connection and save results to a network file system or an FTP directory.
16. Perform an action:
- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character
- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:


- **Column Headings** — Includes column headings
- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

17. Do any of the following:


- To save the export, click .


You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:


- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

- To validate the export, click .

You can copy validation errors to the clipboard.

- To run the export, click .

If you selected to run validations on the **Style** tab, clicking  performs the validations. Then, if no errors occur, the export runs. If validation errors occur,

then the export stops, and an error is displayed. Click  for a list of validation errors.

Compare Export



Compare exports compare two hierarchies and output a record for each node that meets the comparison criteria. You can output this information to a file or a database table. This export can be used to provide incremental changes to target systems.

 **Tip:**

To preserve the export order when exporting to a database table, you can add a record ID column to your output. This column provides an autogenerated numeric value, starting with 1, that increases sequentially for each row in the export.

To create a compare export:

1. On the Home page, select **Export**.

2. Click .
3. Select **Compare** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select versions, hierarchies, and top nodes to compare.

 **Note:**

You must select a hierarchy and top nodes for both the current and previous versions.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.

6. On the **Style** tab, select options:
 - **Node Selection** — Determines node types (All Nodes, Limb Nodes Only, Leaf Nodes Only) included in the export
 - **Join Property** — By default, compares map nodes from each hierarchy that have the same name. You can also join nodes based on other properties. For example, if a property named SortCode has a unique key in the specified hierarchies, then this property is used to identify differences between nodes having the same SortCode in each hierarchy.
 - **Additions** — Includes all nodes in the target (to) hierarchy that are not in the source (from) hierarchy.
 - **Deletions** — Includes all nodes in the source (from) hierarchy that are not in the target (to) hierarchy.
 - **Merges** — Includes merge information in the export (if the merge feature is enabled.)
 - **Moves** — Includes all nodes that have different parent nodes in the compared hierarchies.
 - **Renames** — Includes all nodes that have different names but the same internal NodeID in the compared hierarchies.
 - **Property Updates** — Includes all nodes that have different values in the specified list of properties in the compared hierarchies. Select **Edit** to choose properties.
7. In the **Batch Validations** section, select the type of validations to run.

 **Note:**

Batch validations are performed only for the hierarchies and nodes included in the export. Validations are run before the export. If errors occur, the export stops.

- **None** — No validations run.
 - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export run.
 - **Selected** — Click to select validations to run.
8. On the **Filter** tab, select from these options:
- From **Validation**, select a validation to apply to the export.
 - From **Query**, select a query to apply to the export, or, to define a query, click .

 **Note:**

The inclusion filter determines the set of nodes from each hierarchy to be compared. The exclusion filter excludes nodes from export results.

9. On the **Columns** tab, select the properties to display as columns in the export results and move them from **Available** to **Selected**.

For more information, see [Export Specific Columns](#)

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

10. On the **Column Options** tab, select from these options:
- **Pivot** — If the property contains a comma-delimited list of values, a separate record is exported for each value with all other export columns set to the same values. The Pivot option may be selected on only one column.
 - **Skip Defaults** — To place a blank (empty string) or NULL (in output to database table) in the export if the field value equals the default value of the property
 - **Primary Key** — To define a field (or fields) as the primary key to be used when determining if duplicates exist
 - **Dynamic Value**—If you added a Dynamic Column, define the value for the column here. The value can be null or a combination of constant values and substitution variables.

11. On the **Target** tab, select the output type for the results:

- **Client File** — Download results via the browser.
- **Server File** — Use an external connection and save results to a network file system or an FTP directory.
- **Database Table** — Save results to a database table.

12. Perform an action:


- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF(""ABC"", ""XYZ"", )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character
- If you selected to export to a **Database Table**, you must set your connection and select the database table. You must have already created a valid external connection to a database for it to show up in the drop-down list. You can also select from these options:
 - **Clear Table** options
 - * **Do Not Clear Table** — Appends to existing data in the specified table.
 - * **Clear Based on Version ID(s)** — deletes data in the table where the specified key field equals the current version ID of the data being exported. If this option is selected, a Key field must be specified in the Database Column Options section. When the export is run, all data in the table is first deleted

where the Key field equals the current version ID of the data being exported. For the other Clear Table options, the Key field is ignored.

- * **Clear All Rows** — Deletes data from the table before writing the export output.
- **Use Transactions** — The export is performed within a database transaction and allows the entire operation to be rolled back if an export error occurs.
- **Use Fast Appends** — Performs a bulk insert operation into the target database table for improved export performance. Only available with a database-specific connection.
- **Use Fast Deletes** — Performs a bulk delete operation on the target database table for improved performance. Only available with a database-specific connection.
- **Database Column Options** — Each column being exported must be mapped to a field in the external database table. Under Field Name, select the name of a field. If you selected Clear Based on Version ID(s), specify a Key field.

 **Tip:**

The Length option can be left blank so that string lengths up to the database column declaration can be sent to the database.

 **Note:**

The column must be the same data type as the database field to which it is mapped. For example, if a string property is mapped to a numeric database field, an error occurs.

- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:

- **Column Headings** — Includes column headings
- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.


For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.


- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

13. Do any of the following:


- To save the export, click .



You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User** — Personal objects that are available only to an individual user to view and run.
- **Standard** — Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System** — Restricted objects that are available only to Application Administrator role users to view and run.

- To validate the export, click .

You can copy validation errors to the clipboard.


- To run the export, click .


If you selected to run validations on the **Style** tab, clicking  performs the validations and then, if no errors occur, runs the export. If validation errors occur, then the export stops, and an error is displayed. Click  for a list of validation errors.

Difference Export

Difference exports compare two hierarchies and output a record for each difference found. You can output this information to a file. You can use this export to report incremental changes to users.

To create a difference export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Difference** as the export type.

4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.


- c. Click **OK**.
5. On the **Source** tab, select versions, hierarchies, and top nodes for which you want to export differences.

 **Note:**

You must select a hierarchy and top nodes for both the current and previous versions.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.

6. On the **Style** tab, select from these options:
 - **Node Selection** — Determines which nodes (All Nodes, Limb Nodes Only, Leaf Nodes Only) are included in the export
 - **Join Property** — By default, compares map nodes from each hierarchy that have the same name. You can also join nodes based on other properties. For example, if a property named SortCode has a unique key in the specified hierarchies, then this property is used to identify differences between nodes having the same SortCode in each hierarchy.
 - **Include Summary** — Includes summary of differences in the export result.
 - **Additions** — Includes all nodes in the target (to) hierarchy that are not in the source (from) hierarchy.
 - **Deletions** — Includes all nodes in the source (from) hierarchy that are not in the target (to) hierarchy.
 - **Merges** — Includes merge information in the export (if the merge feature is enabled.)
 - **Moves** — Includes all nodes that have different parent nodes in the compared hierarchies.
 - **Renames** — Includes all nodes that have different names but the same internal NodeID in the compared hierarchies.
 - **Property Updates** — Includes all nodes that have different values in the specified list of properties in the compared hierarchies. Click **Edit** to select properties.

7. On the **Filter** tab, select an option:
 - From **Validation**, select a validation to apply to the export.
 - From **Query**, select a query to apply to the export, or, to define a query, click .

 **Note:**

The inclusion filter determines the set of nodes from each hierarchy to be compared.

8. On the **Columns** tab, select the properties to display as columns in the export results and move them from **Available** to **Selected**.

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

9. On the **Column Options** tab, select options:
 - **Pivot** — If the property contains a comma-delimited list of values, a separate record is exported for each value with all other export columns set to the same values. The Pivot option may be selected on only one column.
 - **Skip Defaults** — To place a blank (empty string) or NULL (in output to database table) in the export if the field value equals the default value of the property
 - **Primary Key** — To define a field (or fields) as the primary key to be used when determining if duplicates exist
10. On the **Target** tab, select the output type for the results:
 - **Client File** — Download results via the browser.
 - **Server File** — Use an external connection and save results to a network file system or an FTP directory.
 - **Database Table** — Save results to a database table.
11. Perform an action:
 - If you selected to export to a **Client File**, select options:
 - **Column Headings** — Includes column headings in the export.
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF ("ABC", "XYZ", )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. All instances in the export are replaced as specified. For example, all commas can be replaced by tabs.
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click to open a text editor.



Note:

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character.
- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:
 - **Column Headings** — Includes column headings
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF ("ABC", "XYZ", )
```

changes to

```
"=IF ("ABC", "XYZ", )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

12. Do any of the following:

- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User** — Personal objects that are available only to an individual user to view and run.
- **Standard** — Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System** — Restricted objects that are available only to Application Administrator role users to view and run.

- To run the export, click .

If you selected Client File, you have a download option where you can download the file locally.

EPM Architect Export

Oracle Hyperion EPM Architect exports output hierarchies, nodes, and properties from Oracle Data Relationship Management in an import format for Performance Management Architect. The output can be shared with and deployed to multiple EPM applications, such as Oracle Hyperion Financial Management and Oracle Hyperion Planning. Performance Management Architect exports may be run from within Data Relationship Management or from an Performance Management Architect import profile.



 **Note:**

The Performance Management Architect export cannot create new Performance Management Architect dimensions that use properties that require dimension associations. The dimensions must first be created in Performance Management Architect along with their dimension associations, and then you can import from Data Relationship Management into those newly created dimensions in Merge or Replace mode.

For more information on running these exports from Performance Management Architect import profiles, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

To create an EPM Architect export:

1. On the Home page, select **Export**.

2. Click .
3. Select **EPM Architect** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.



 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, do the following:
 - a. Select a version or version variable.
 - b. Select a target application type:
 - Consolidation
 - Planning
 - Essbase (ASO)
 - Essbase (BSO)
 - Shared Library
 - c. Select a dimension property. This is a property with data type Hierarchy Group, used to determine the dimensions and hierarchies being exported.
 - d. Select a dimension type property. This is a hierarchy-level lookup property used to determine the dimension type for each dimension being exported.
6. On the **Style** tab, select an option:
 - **Include Inactive Nodes** — Select to include inactive nodes.
 - **Include Implicitly Shared Nodes** — Select to include descendant nodes below shared limb nodes that are explicitly shared. Implicit shared nodes are exported only if not filtered out by another option such as top node selection or a query filter. The structure of implicitly shared nodes below a shared parent node is reflected in the ancestor section of the hierarchy export.
 - If a query filter is used in the export, the administrator must explicitly include the shared nodes. For example, the filter (#Children=0) must be changed to ((#Children=0 OR (Shared Node Flag = True)).
 - For the implicitly shared node, the Parent properties reflect the values of the actual node (the primary node). For example, if your export includes the Parent Node property for an implicitly shared node, it will show the parent as the primary node.
7. In the **Batch Validations** section, select the type of validations to run.


 **Note:**

Batch validations are performed only for the hierarchies and nodes included in the export. Validations are run before the export. If errors occur, the export stops.


- **None** — No validations run.
 - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export run.
 - **Selected** — Click  to select validations to run.
8. **Optional:** On the **Filter** tab, select a query to apply to the export or, to define a query, click .


 **Note:**



The inclusion filter determines the set of nodes from each hierarchy to be compared.

9. On the **Columns** tab, select the **Parent Key Property** and **Member Key Property** (the Data Relationship Management property for the node when determining duplicates).
10. Do any of the following:
- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

 - **User** — Personal objects that are only available to an individual user to view and run.
 - **Standard** — Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
 - **System** — Restricted objects that are only available to Application Administrator role users to view and run.
 - To validate the export, click .

You can copy validation errors to the clipboard.
 - To run the export, click .

If you selected to run validations on the **Style** tab, clicking  performs the validations. Then, if no errors occur, the export runs. If validation errors occur, then the export stops, and an error is displayed. Click  for a list of validation errors.

Version Export

Version exports output a record for each node and its properties for an entire version. This information can be output to a file or a database table.



 **Tip:**

To preserve the export order when exporting to a database table, you can add a record ID column to your output. This column provides an auto-generated numeric value, starting with 1, that increases sequentially for each row in the export.

 **Note:**

Hierarchy relationships and local properties are not available for this export type.

To create a version export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Version** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
 6. On the **Style** tab, select from these options:
 - **Node Selection** — Determines which types of nodes (All Nodes, Limb Nodes Only, Leaf Nodes Only) are included in the export
 - **Include Implicitly Shared Nodes** — Select to include descendant nodes below shared limb nodes that are explicitly shared. Implicit shared nodes are

exported only if not filtered out by another option such as top node selection or a query filter. The structure of implicitly shared nodes below a shared parent node is reflected in the ancestor section of the hierarchy export.

- If a query filter is used in the export, the administrator must explicitly include the shared nodes. For example, the filter (#Children=0) must be changed to ((#Children=0 OR (Shared Node Flag = True)).
- For the implicitly shared node, the Parent properties reflect the values of the actual node (the primary node). For example, if your export includes the Parent Node property for an implicitly shared node, it will show the parent as the primary node.

7. In the **Batch Validations** section, select the type of validations to run.

 **Note:**

Batch validations are performed for the entire version specified in the export. Validations are run before the export. If errors occur, the export stops.

- **None** — No validations run.
- **Assigned** — All assigned validations run.
- **Selected** — Click to select validations to run.

 **Note:**

Validations are performed for the entire version.

8. On the **Filter** tab, select options:
 - From **Validation**, select a validation to apply to the export.
 - From **Query**, select a query to apply to the export or, to define a query, click .

 **Note:**

The inclusion filter determines the set of nodes from each hierarchy that will be compared.

9. On the **Columns** tab, select the properties to display as columns in the export results and move them from **Available** to **Selected**.

For more information, see [Export Specific Columns](#)

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

10. On the **Target** tab, select the output type for the results:

- **Client File** — Download results via the browser.
- **Server File** — Use an external connection and save results to a network file system or an FTP directory.
- **Database Table** — Save results to a database table.

11. Perform an action:


- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to


```
"=IF(""ABC"", ""XYZ"", )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace options** — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter options** — To select a field delimiter character and a record delimiter character

- If you selected to export to a **Database Table**, you must set your connection and select the database table. You must have already created a valid external connection to a database for it to show up in the drop-down list. You can also select from these options:
 - **Clear Table** options
 - * **Do Not Clear Table** — Appends to existing data in the specified table.
 - * **Clear Based on Version ID(s)** — deletes data in the table where the specified key field equals the current version ID of the data being exported. If this option is selected, a Key field must be specified in the Database Column Options section. When the export is run, all data in the table is first deleted where the Key field equals the current version ID of the data being exported. For the other Clear Table options, the Key field is ignored.
 - * **Clear All Rows** — Deletes data from the table before writing the export output.
 - **Use Transactions** — The export is performed within a database transaction and allows the entire operation to be rolled back if an export error occurs.
 - **Use Fast Appends** — Performs a bulk insert operation into the target database table for improved export performance. Only available with a database-specific connection.
 - **Use Fast Deletes** — Performs a bulk delete operation on the target database table for improved performance. Only available with a database-specific connection.
 - **Database Column Options** — Each column being exported must be mapped to a field in the external database table. Under Field Name, select the name of a field. If you selected Clear Based on Version ID(s), specify a Key field.
-  **Tip:**

The Length option can be left blank so that string lengths up to the database column declaration can be sent to the database.
-  **Note:**


The column must be the same data type as the database field to which it is mapped. For example, if a string property is mapped to a numeric database field, an error occurs.
- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:
 - **Column Headings** — Includes column headings
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ",)
```

changes to

```
"=IF("ABC", "XYZ",)"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace options** — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter options** — To select a field delimiter character and a record delimiter character

12. Do any of the following:


- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:


- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

- To validate the export, click .

You can copy validation errors to the clipboard.

- To run the export, click .

If you selected to run validations on the **Style** tab, clicking  performs the validations. If no errors occur, the export runs. If validation errors occur, then

the export stops and an error is displayed. Click  for a list of validation errors.


Property List Export

Property List exports output a record for each item in the list of values for a property definition. You can output this information to a file.

Note:

This is a special use case export type.

To create a property list export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Property List** as the export type.
4. On the **Source** tab, select a version or version variable.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.

Note:

Even though the results are not filtered by version, for this release you must select a version.

5. From **Available**, select a list property upon which to base the export.
6. On the **Target** tab, select the output type for the results:
 - **Client File** — Download results via the browser.
 - **Server File** — Use an external connection and save results to a network file system or an FTP directory.
7. Perform an action:
 - If you selected to export to a **Client File**, select options:

- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to


```
"=IF ("ABC", "XYZ", )"
```

- **Column Headings** — Includes column headings
 - If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.


For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```
 - **Column Headings** — Includes column headings
8. Do any of the following:
- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
 - To run the export, click .

If you selected Client File, you have a download option where you can download the file locally.

Property Lookup Export


Property Lookup exports output a record for each key-value pair in the lookup table for a property definition. You can output this information to a file.




Note:

This is a special use case export type.

To create a property lookup export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Property Lookup** as the export type.

4. **Optional:** To create substitution parameters for the export:

- a. Click .
- b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.

5. On the **Source** tab, select a version or version variable.

You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.

 **Note:**

Even though the results are not filtered by version, for this release you must select a version.

6. From **Available**, select a lookup property upon which to base the export.

7. **Optional:** Select the **Style** tab, and select **Differences Only** to filter out lookup pairs where the Lookup Key and Result Value are the same.

8. On the **Columns** tab, select the columns for the export.

Only two columns are available because the Lookup Key and the Result Value are the only relevant columns for this export type.

 **Tip:**

Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

9. On the **Target** tab, select the output type for the results:

- **Client File** — Download results via the browser.
- **Server File** — Use an external connection and save results to a network file system or an FTP directory.

10. Perform an action:


- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ",)
```

changes to

```
"=IF("ABC", "XYZ",)"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace options** — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter options** — To select a field delimiter character and a record delimiter character
- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:
 - **Column Headings** — Includes column headings
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.


For example,

```
=IF("ABC", "XYZ",)
```

changes to

```
"=IF("ABC", "XYZ",)"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace options** — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs


- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

11. Do any of the following:

- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.



- To run the export, click .

If you selected Client File, you have a download option where you can download the file locally.

Transaction Log Export

Transaction Log exports output a record for each item in the transaction log meeting the filter criteria. You can output this information to a file or a database table. This export can be used for audit purposes.

To create a transaction log export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Transaction Log** as the export type.
4. **Optional:** To create substitution parameters for the export:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.
You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
6. Click to select hierarchies and nodes for the export.
7. On the **Filter** tab, select options:
 - **Levels** — Click to specify levels to filter:
 - **Core** — Includes actions that indicate a change to version data (not metadata objects).
 - **Logged** — Includes actions that indicate system activity occurred but did not result in additions, changes, or updates to data.
 - **Loss** — Includes transaction history records that store data that has been lost due to a core action, such as deleting a node.
 - **Result** — Includes actions that are associated with, and are a result of, a Core action.
 - **Admin** — Includes actions that affect metadata objects (not version related data).
 - **Actions** — Click to specify transaction log action types to filter.
 - **Properties** — Click to specify a list of properties to filter.
 - **Users** — Click to specify users to filter.
 - **Include Child Nodes** — Select to include child nodes.
 - **Include Shared Nodes** — Select to include shared nodes.
 - **From Date** — Select and enter a start date.
 - **To Date** — Select and enter an end date.
 - **From Transaction** — Select and enter a start transaction ID.
 - **To Transaction** — Select and enter an end transaction ID.
 - **Filter to Current Session** — Filter transactions to those performed during the current session.
8. On the **Columns** tab, select the columns for the export.

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

9. On the **Target** tab, from the **Device** list, select the location to which you want to export the results: Client File Database Table, or Server File.

10. Perform an action:


- If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF(""ABC"", ""XYZ"", )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character
- If you selected to export to a **Database Table**, you must set your connection and select the database table. You must have already created a valid external connection to a database for it to show up in the drop-down list. You can also select from these options:
 - **Clear Table** options
 - * **Do Not Clear Table** — Appends to existing data in the specified table.

- * **Clear Based on Version ID(s)** — deletes data in the table where the specified key field equals the current version ID of the data being exported. If this option is selected, a Key field must be specified in the Database Column Options section. When the export is run, all data in the table is first deleted where the Key field equals the current version ID of the data being exported. For the other Clear Table options, the Key field is ignored.
- * **Clear All Rows** — Deletes data from the table before writing the export output.
- **Use Transactions** — The export is performed within a database transaction and allows the entire operation to be rolled back if an export error occurs.
- **Use Fast Appends** — Performs a bulk insert operation into the target database table for improved export performance. Only available with a database-specific connection.
- **Use Fast Deletes** — Performs a bulk delete operation on the target database table for improved performance. Only available with a database-specific connection.
- **Database Column Options** — Each column being exported must be mapped to a field in the external database table. Under Field Name, select the name of a field. If you selected Clear Based on Version ID(s), specify a Key field.

 **Note:**

The column must be the same data type as the database field to which it is mapped. For example, if a string property is mapped to a numeric database field, an error occurs.

- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:

- **Column Headings** — Includes column headings
- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.


For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.


- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

11. Do any of the following:

- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

- To run the export, click .

If you selected Client File, you have a download option where you can download the file locally.


Merge Log Export


Merge Log exports output a record for each item in the merge log meeting the filter criteria. You can output this information to a file.

 **Note:**

This export is applicable only if the Merge feature is used.


To create a merge log export:

1. On the Home page, select **Export**.
2. Click .
3. Select **Merge Log** as the export type.
4. **Optional:** To create substitution parameters for the export:

- a. Click .
- b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
5. On the **Source** tab, select a version or version variable.
You can select Normal, Baseline, or As-Of versions. The Version list defaults to the Normal versions. To select a Baseline or As-Of version, click **Normal** under the Version list.
6. On the **Filter** tab, select from these options:
 - **Begin/End Dates** — Set the beginning and ending date for the export
 - **Deleted Node** — Enter the name of a deleted node.
 - **Merge Node** — Click  to specify a merge node
 - **User** — Select a user
 - **Status** — Select a status
7. On the **Columns** tab, select the columns for the export.

 **Tip:**

Use the Category list to select a property category to ease navigation. Use the up and down arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.


8. On the **Target** tab, from the **Device** list, select the location to which you want to export the results: Client File or Server File.
9. Perform an action:
 - If you selected to export to a **Client File**, choose options:
 - **Column Headings** — Includes column headings in the export
 - **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF ("ABC", "XYZ", )"
```

- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).

- **Delimiter** options — To select a field delimiter character and a record delimiter character
- If you selected to export to a **Server File**, you must set your connection and enter a file name. You must have already set up a valid external connection to a UNC or FTP file. You can also select from these options:


- **Column Headings** — Includes column headings
- **Quoted Strings** — Puts quotation marks around each column value. Any existing quotation marks within the column value are doubled.

For example,

```
=IF("ABC", "XYZ", )
```

changes to

```
"=IF( ""ABC"" , ""XYZ"" , )"
```


- **Fixed Width** — Allows you to specify field widths, left or right justification for the column, and number of pad characters. You can enter column information in the Fixed Width Option section.
- **Character Encoding** — Select one of the encoding options.
- **Replace** options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- **Header/Footer** — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

 **Note:**

You can insert customization tags to further enhance the readability of the export results. For more information, see [Customization Tags](#).


- **Delimiter** options — To select a field delimiter character and a record delimiter character

10. Do any of the following:

- To save the export, click .

You are prompted to define a name, description, and an object access group for the export. Select a custom group or one of the following:

- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

- To run the export, click .

If you selected Client File, you have a download option where you can download the file locally.

 **Note:**

If you selected to run batch validations for exports in the book, these validations are run first.

Creating Export Books

Saved exports can be grouped and run together in Export Books. This enables a set of commonly used exports to be executed with one action rather than individually. Books can also be used to combine the output of multiple exports into a single output file.

 **Note:**

Book exports cannot reference multiple versions. All exports for a book must reference the same versions.



A book requires both a Current Version and Previous Version to be specified.

- The Current Version is used for all exports in the book where a Version or To Version parameter is required.

- The Previous Version is used for all exports in the book where a From Version parameter is required.

The Current Version and Previous Version parameters for the book override any versions configured for the exports in the book. This includes all version types (normal, baseline, as-of, variable).

To create an export book:

1. On the Home page, select **Export**.
2. Click .
3. **Optional:** To create substitution parameters for the book:
 - a. Click .
 - b. Click **Add** and add Key-Value pairs.

 **Note:**

Key-Value pairs cannot use "<%" or "%>" sequentially or an equal sign.

- c. Click **OK**.
4. On the **Source** tab, select a version or version variable from which to select exports for the book.
 5. From **Object Access**, select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.

 **Note:**

Depending on the access level you select, the list of available exports may be restricted. User books can select user and standard exports. Standard books can only reference standard exports, and System books can only reference standard or system exports.

6. Select exports to include in the book and move them from **Available** to **Selected**.

 **Note:**

You can use the Export Type list to filter the list of available exports.


7. **Optional:** If the output is to be combined, select the **Target** tab and do the following:
 - a. From **Device**, specify if the combined file will be a server file or client file.
 - b. Select **Include Combined Export Output File**.

- c. From **Character Encoding**, select the encoding to apply to the combined file and the pre and post files if used.
- d. If you are combining into a server file, set up an external connection.
- e. **Optional:** Set up external connections for Pre Files and Post Files.
Pre Files and Post Files can be used to specify data that is prepended and appended to the combined output file.

 **Note:**


If a book contains both exports configured to output to a file and exports configured to output to a database table, the Combine option affects only the exports that are writing to files.

8. Do any of the following:

- To save the export, click .


You are prompted to define a name, description, and an object access group for the book. Select a custom group or one of the following:

- **User**—Personal objects that are only available to an individual user to view and run.
- **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
- **System**—Restricted objects that are only available to Application Administrator role users to view and run.

- To run the book, click .

If you selected Client File, you have a download option where you can download the file locally.

 **Note:**

For any exports in the book that are running validations, clicking  performs the validations and then, if no errors occur, runs the book. If validation errors occur, then the book process stops and an error is displayed. Open each export that is running validations and click




to find the errors.


Running, Copying, Moving, and Deleting Exports and Export Books

You can run, copy, move, and delete exports and export books. You must open an export or export book before you can run it.


To run an export or book:

1. On the Home page, select **Export**.
2. Do one of the following:
 - On the **Export** tab, double-click an export.
 - On the **Books** tab, double-click an export book.
3. Click the tabs to modify the export or export book parameters, and then click .

 **Note:**

For exports, if you selected to run validations on the **Style** tab, clicking  performs the validations and then, if no errors occur, runs the export. If validation errors occur, then the export stops and an error is displayed. Click



for a list of validation errors. For books, if a validation error occurs, then open each export that is running validations and click  to find the errors.

To copy an export or export book:

1. On the Home page, click **Export** and then select an export or export book to copy.
2. From **Exports** select **Copy**.
3. Enter a new name and description for the export or export book.
4. From **Object Access Group**, select a custom group or one of the following:
 - **User**—Personal objects that are only available to an individual user to view and run.
 - **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
 - **System**—Restricted objects that are only available to Application Administrator role users to view and run.
5. Click **OK**.

To move an export or export book:

1. On the Home page, select **Export**, and then select an export to move.
2. From **Exports**, select **Move**.
3. From **Object Access Group**, select a different group and click **OK**.

To delete an export or export book:

1. On the Home page, select **Export** and then select an export or export book to delete.
2. From **Exports**, select **Delete**.

15

Job Management

Job management allows for greater control and visibility over long running operations (jobs) and the results they produce. Running an operation as a job enables a user to perform a different task while the process is running and have access to the status and any results of the operation when the user returns. The following types of operations in Oracle Data Relationship Management are processed as jobs:

- Action Script: Load File, Load Log Criteria, Process Actions
- Annul All Below
- Blend
- Book
- Compare: Structure, Property, Rename
- Domain Assign Membership
- Export
- Import
- Property Profile
- Put Nodes
- Query: Local, Global
- Validate: Version Selected, Version Assigned, Version Selected Global, Version Assigned Global, Hierarchy Selected, Hierarchy Assigned, Node Selected, Node Assigned, Specific Node Selected, Specific Node Assigned
- Version: Backup, Close, Copy, Create AsOf, Delete, Load, Restore, Save

Some job types can be run in the background to enable users to cancel a job or navigate away from the page where the job was executed to perform another task. The following job types are run in the background:

- Action Script
- Blend
- Book
- Compare (Structure, Property, Rename)
- Export
- Import
- Query

Running Jobs

When you run a job, the job is added to a queue of currently running jobs. If you navigate away from the running job tab, close the running job tab, or log off of Data Relationship Management, the job continues to run. If the running job tab is open when the job completes, the results (for those jobs that return results) are displayed on the tab.

Cancelling Jobs

For jobs running on a tab, you can click the Cancel button (red square) to cancel the job. You can also cancel a job from the job history window. Note that some jobs cannot be cancelled. The following job types cannot be cancelled:

- Run Action Script
- Blend to Saved Version (if the blend target is "Copy To New Version", then the job can be cancelled, otherwise the job cannot be cancelled.
- Domain Assign Membership
- Put Nodes
- Version Delete
- Version Save

Purging Job Results

Job results can be individually purged from the job history. Application administrators can also use the JobResultsRetentionAge system preference to define how long job results are stored before they are automatically purged by the system .

Auditing Transactions, Jobs, and Requests

Oracle Data Relationship Management records a history of different activities performed within an application for audit purposes. The transaction history logs all operations performed including changes to version data, application metadata, and user security. The job history tracks the completion of long running processes such as imports, blenders, and exports. The request history maintains a record of change requests submitted from external applications.

The audit wizard is used to define the parameters and criteria for the query. The wizard content is organized across Source, Filters, and Columns tabs to guide input. Although the wizard tabs are ordered to gather input in a sequence, you can navigate between any tabs.

You can query and view transactions, requests, and job history using the Audit task. For more information, see:





- [Auditing Transaction History](#)
- [Data Actions in Transaction History](#)
- [Auditing Requests](#)
- [Viewing Job History](#)

Auditing Transaction History

All Oracle Data Relationship Management operations are logged in the Transaction History. Each operation is recorded with a time stamp, the user name, the type of action performed, and other relevant information.

To query and view transaction history:

1. From the Home page, select **Audit**.
2. From **Audit Type** select **Transactions**.
3. On the **Source** tab, do the following:
 - From **Version, Hierarchies, Nodes, and Request ID** select the source for the audit.
 - Select the type of transactions to return: **System Transactions, Data Transaction, or both**.
4. On the **Filters** tab, select from these options:
 - **Levels** — Click to specify levels to filter:
 - **Core** — Includes actions that indicate a change to version data (not metadata objects).
 - **Logged** — Includes actions that indicate system activity occurred but did not result in additions, changes, or updates to data.
 - **Loss** — Includes transaction history records that store data that has been lost due to a core action, such as deleting a node.
 - **Result** — Includes actions that are associated with, and are a result of, a Core action.

- **Admin** — Includes actions that affect metadata objects (not version related data).
 - **Actions** — Click  to specify transaction log action types to filter.
 - **Properties** — Click  to specify a list of properties to filter.
 - **Users** — Click  to specify users to filter.
 - **Include Child Nodes** — Select to include child nodes for nodes returned by the query.
 - **Include Shared Nodes** — Select to include shared nodes for nodes returned by the query.
 - **From Date** — Select and enter a start date.
 - **To Date** — Select and enter an end date.
 - **Filter to Current Session** — Select to filter transactions to those performed during the current session.
 - **From Transaction** — Select and enter a start transaction ID.
 - **To Transaction** — Select and enter an end transaction ID.
 - **Max Records** — Enter a value to limit the number of transactions displayed.
5. On the **Columns** tab, select the fields to be displayed as columns in the query results.
 6. Click .

The transaction history query results are displayed below the wizard.
 7. After running the query, you can:
 - View the details of a transaction by clicking the arrow to the left of the row. The transaction details are displayed across Action, Object, History, and Request tabs.

 **Note:**

The Request tab is available only if the transaction is associated with a workflow request.

- Click **Download**, and then select the format for the file : TSV, CSV, PDF, RTF, or XLS.

Data Actions in Transaction History

The transaction history provides a record of all the data changes to the core system such as version, hierarchy, and node data, as well as logging many basic functions such as user logins. The data transactions are categorized by action types. Following is a list of the action types that are logged:

Core Data Changes

- Add Domain To Version

- Add Hierarchy
- Add Node
- Add Orphan Node
- Annul All Below Node
- Annul Node
- Approve Request
- Assign Hierarchy Validations
- Assign Node to Domain
- Assign Version Validations
- Change Top Node
- Clear All Below
- Delete Hierarchy
- Delete Node
- Delete Orphan
- Inactivate Node
- Insert Node
- Lock Node Property
- Move Node
- Reactivate Node
- Remove Domain From Version
- Remove Duplicate Property Values
- Remove Hierarchy Property Value
- Remove Node
- Remove Node Property Value
- Remove Version Property Value
- Unlock Node Property
- Update Controlled Properties
- Update Hierarchy Property Value
- Update Node Property Value
- Update Version Property Value

Logged Actions

- Action Script Run
- Add Category
- Add Domain
- Add Glyph
- Add Hierarchy Group

- Add Node Access Group
- Add Node Type
- Add Object Access Group
- Add Property Definition
- Add Property Lookup Value
- Add User
- Add Validation
- Add Version
- Add Version Variable
- Add Workflow Model
- Add Workflow Task
- Apply Node Type Props
- Apply Node Type Validations
- Assign User Categories
- Assign User Node Access Groups
- Backup To File
- Batch Validation Run
- Blender Object Access Group Changed
- Blender Run
- Book Object Access Group Changed
- Bulk Copy Process Run
- Change User Password
- Close Version
- Compare Object Access Group Changed
- Compare Run
- Connection Object Access Group Changed
- Copy Version
- Delete Blender
- Delete Book
- Delete Category
- Delete Compare Criteria
- Delete Criteria List
- Delete Domain
- Delete Export
- Delete Glyph
- Delete Hierarchy Group
- Delete Import

- Delete Node Access Group
- Delete Node Type
- Delete Object Access Group
- Delete Property Definition
- Delete Property Lookup Value
- Delete Request
- Delete User
- Delete Validation
- Delete Version
- Delete Version Variable
- Delete Workflow Model
- Delete Workflow Task
- Detach Version
- Export Object Access Group Changed
- Export Run
- Global Query
- Import Object Access Group Changed
- Import Run
- Log In
- Log Out
- Migration Export
- Migration Import
- Open Version
- Order Children
- Profiler Run
- Property Compare Run
- Query
- Query Object Access Group Changed
- Rebuild Hierarchy Node
- Rename Compare Criteria
- Rename Compare Run
- Rename Criteria List
- Restore from File
- Restore from File Warning
- Save Blender
- Save Book
- Save Compare Criteria

- Save Export
- Save Import
- Save Request
- Save Version
- Set User Password
- Set Validation Params
- Update Access Group Controlling Hiers
- Update Access Group Members
- Update Category
- Update Category List for Prop Def
- Update Category Properties
- Update Category Users
- Update Compare Criteria
- Update Criteria List
- Update Domain
- Update Glyph
- Update Hier Validations
- Update Hierarchy Group
- Update Node Access Group
- Update Node Type
- Update Object Access Group
- Update Property Definition
- Update Property Definition Parameters
- Update Property List Values
- Update Property Lookup Value
- Update System Preference
- Update User
- Update User Lockout
- Update Validation
- Update Version Validations
- Update Version Variable
- Update Workflow Model
- Update Workflow Task
- Validations Sorted

Loss Actions

- Node Property Data Loss

- Remove From Orphan

Auditing Requests

Change and remediation requests processed by governance workflows in Oracle Data Relationship Management can be audited by governance users and Data Managers using the Worklist. For more information, see [Navigating the Worklist](#).


External requests, which are processed using the request interface in the Web Service API, are recorded in Request History. Change or remediation requests available from the Worklist page are not recorded in Request History. Users must have the Workflow User role to access the Request History. Each request is recorded with timestamps, the user ID of the request owner, the actions being performed, the validation and approval status, and other request level information. When you first access the Request History screen, a default query displays all unapproved requests that you own.

The top section of the Request History screen allows you to define query criteria. The bottom section displays the list of change requests that meet the current query criteria. Query criteria are grouped into four categories:

- Summary
 - Request ID — Cannot be changed; automatically generated when the request is created.
 - Owned By user ID — Cannot be changed; defined when request is created.
 - Committed By user ID — Cannot be changed; defined when request is approved.
 - Target version for the request.
 - Perform exact match on string values option.
 - Maximum number of records to return setting.
- Workflow — Used to define applicable data.
 - Custom ID
 - Custom Label
 - Custom Status
 - Comments
- History
 - Created By, Updated By, and Approved By fields for managing user-related information for the workflow application.
 - Create Date, Update Date, and Approval Dates for when a request is created, updated, and approved.

To query and view request history:

1. From the Home page, select **Audit**.
2. From **Audit Type** select **Requests**.
3. On the **Source** tab, enter a request ID, a workflow ID, or select a version, hierarchies, and nodes to use as the source for the query.
4. On the **Filter** tab, define the filtering criteria for the request history query including date ranges and created by, updated by, approved by users.

5. On the **Columns** tab, select the fields to be displayed as columns in the query results.
6. Click .
The request history query results are displayed below the wizard.
7. After running the query, you can:
 - To view the details of a request, click the plus-sign to the left of the row. The request details are displayed across Action, Object, and History tabs.
 - Click **Download**, and then select the format for the file : TSV, CSV, PDF, RTF, or XLS.



Viewing Job History

The list of running and completed jobs can be viewed and queried in the job history. Job results are retained by the application server and may be retrieved later in the same user session or in a later session. The following job types return results which can be viewed in job history:

- Action Script
- Blend
- Compare
- Export or Book - results are stored when the output device is a client file only.
- Import
- Query

The job history list can be filtered to show a subset of jobs. If no filtering is used, all jobs are returned. You can also select and order the columns for the job history list.

To view job history:

1. From the Home page, select **Audit**.
2. From **Audit Type**, select **Jobs**.
3. On the **Filter** tab, make changes to these fields to filter the job history list:
 - **Type**—Click  and select a job type.
 - **Status**—Click  and select a job status
 - **Object Name**—Enter the name of an object. You can use an asterisk for wildcard searches.
 - **Version**—Select a version to filter the results to jobs from a single version.
 - **From Date**—Select the check box and enter a starting date and time for the job.
 - **To Date**—Select the check box and enter an ending date and time for the job.
 - **Max Records**—Enter the maximum number of records to return.
4. On the **Columns** tab, select columns from the **Available** list and move them to the **Selected** list. Use the up and down arrows to order the columns.



 **Note:**

The Type, Description, and Action columns are always included in the job history list. These columns cannot be deselected or ordered. Type and Description are the first two columns and Action is the last column.

The columns for the job history list are:

- Action—For jobs that return a result, Delete action is available. For jobs currently in progress, Cancel action is available.
- Description—Description of the job
- Elapsed Time—Amount of time that the job took to run
- Finish—Date and time the job finished
- Job Name—Name of the object that was run
- Message—Information message for jobs that completed with errors
- Results Purged—Date and time that the results were purged
- Results Size—Size in kilobytes of the results attachment
- Results Viewed—Date and time that the results were viewed
- Start—Date and time the job was started
- Status—Status of the job such as Done or Done with Errors
- Type—Type of job being run, such as Version Load, Export, and Action Script Load File
- Version From—Name of the From version
- Version To—Name of the To version

5. Perform an action:

- To refresh the job history list, click .
- To view job results for a successful job, click the **Done** link in the **Status** column.
- To purge job results, in the **Action** column, click .

 **Note:**

When job results are purged from the job history, the job results attachment is deleted but the job entry itself remains in the history.

Using the Data Relationship Management Batch Client

The Oracle Data Relationship Management Batch Client is a command-line utility that allows access to various Data Relationship Management operations in batch mode including:

- Action Script
- Import
- Blender
- Export (individual export and book export)
- Opening and closing versions
- Submitting governance requests

The Batch Client utility (`drm-batch-client.exe`) is installed by default to: `C:\Oracle\Middleware\EPMSysstem11R1\products\DataRelationshipManagement\client\batch-client\`.

By passing command-line parameters appropriate to the operation, the Batch Client is able to log in to the Data Relationship Management engine using supplied credentials, perform the operation, and receive any corresponding results from the server, while logging its activities as it works. If the operation is successful, an error level value of 0 is returned. If the operation fails, a non-zero error level value is returned. For more information, see [Error Level Values](#).

The Batch Client accepts options from the command line, a configuration file, or a combination of the two.



Note:

If an option is supplied in both a configuration file and the command line with different values, the value on the command line takes precedence over the value in the configuration file.

Securing the Batch Client

The Batch Client is used to perform batch processes in Oracle Data Relationship Management. It requires a username and password to connect to the Data Relationship Management server.

The Batch Client can use the Windows Data Protection API (DPAPI) to allow a more secure method to storing credentials for the Batch Client. The DPAPI protects the information and places it in a location that is only accessible to the user that creates it. Data Relationship Management provides a command line utility that can set the credentials used by the Batch Client.

When the Batch Client runs and no credentials are passed on the command line or in the configuration file, then it uses the credentials stored using the Windows DPAPI.

Setting up Batch Client Credentials

The Credential Utility (`drm-batch-client-credentials.exe`) is used to set up the username and password credentials for the batch. The Credential Utility associates a Data Relationship Management user to the current Windows account in order to execute the Batch Connect tool securely.

The following command line parameters are valid:

- **Add** — Stores Data Relationship Management credentials for the current Windows account.
- **Clear** — Clears the Data Relationship Management credentials for the current Windows account from the secure store.
- **Info** — Displays the Data Relationship Management username associated with the current Windows account.
- **?** — Displays the help file.

If no arguments are passed, the Credential Utility launches an interactive prompt.

Running the Batch Client

To run the Oracle Data Relationship Management Batch Client:

1. Open a command prompt.
2. Navigate to the directory where you downloaded the installation program; for example
`C:\Oracle\Middleware\EPMSysstem11R1\products\DataRelationshipManagement\client\batch-client\.`
3. Enter `drm-batch-client.exe`, followed by the parameters as described in this chapter.

You can run the Batch Client from a command prompt or schedule it to run using an external program. The Batch Client performs these actions:

1. Opens the log file. If the log file already exists, the Batch Client appends to it. If the log file does not exist, the Batch Client creates it.
2. Processes and validates options from the command line or configuration file.
3. Logs into Data Relationship Management using supplied credentials.
4. Runs the appropriate operations.
5. Writes output file.
6. Logs out of the server, closes the log file, sets the error level, and exits.

Example 17-1 Dependencies

You must have the required Microsoft .NET Framework installed on the machine running the Batch Client. See the *Oracle Data Relationship Management Installation Guide* for the required .NET version. Connectivity/visibility to the Engine application is also required.

The Batch Client requires a TCP/IP connection to Data Relationship Management. You must have permissions to use the connection and to read files from directories indicated for input files and to write to directories indicated for log files and export files. For more information on Data Relationship Management requirements, see the *Oracle Data Relationship Management Installation Guide*.

**Note:**

The Batch Client is a separate utility program and is not accessible from within the Data Relationship Management Web client interface.

Example 17-2 Saving Versions in the Batch Client

When the Batch Client runs, it logs into an existing Data Relationship Management application server. When new versions are created, they remain detached (not saved to the database) on the application server until they are saved.

When running imports or blends in the Batch Client, you have the option to save the new version (Autosave=True). You can save the version or leave it detached.

You can specify date and time As-Of versions only. As-Of versions using date offsets or transaction IDs are not valid.

Command Line and Configuration File Syntax

The Batch Client can be used with command-line parameters or with a configuration file.

Command-line Syntax and Example

The Batch Client can be used with the command-line parameters in [Command Line Parameters and Configuration File Keys](#). Bracketed values ([]) are optional extensions provided for clarity. Both shorter and longer versions of the value are supported. Commands and parameters are not case-sensitive.

**Caution:**

For all parameters, version names containing a space must be in quotes, for example, "April – Production".

Below is an example of the command-line syntax.

```
C:\Oracle\Middleware\EPMSysstem11R1\products\DataRelationshipManagement\client
\batch-client\drm-batch-client.exe /url=net.tcp://localhost:5210/Oracle/Drm/
Engine /u=mylogin /pw=mypassword /log="C:\Oracle\Data Relationship
Management\rdsc.log" /op=Export
/xtype=Export /xname="WeeklyExport" /cver="COA Current" /pver="COA Previous"
/outfile="C:\Oracle\Data Relationship Management\export_data.txt"
```

Configuration File Syntax and Example

Configuration files are formatted like INI files, with sections, keys, and values. There is a General section and an additional section for the specific operation. Section names are enclosed in brackets ([]) and can be any string value but must match the value specified by the operation key. Lines within each section are formatted in key=value pairs. The Batch Client can be used with the configuration file keys in [Command Line Parameters and Configuration File Keys](#)

Below is an example Batch Client configuration file.

```
[General]

Operation=Export

UserName=mylogin

Password=mypassword

URL=net.tcp://localhost:5210/Oracle/Drm/Engine

LogFileNames="C:\Oracle\Data Relationship Management\rdsc.log"

[Export]

ExportType=Export

ExportName=WeeklyExport

OutFile= "C:\Oracle\Data Relationship Management\export_data.txt"

CurrentVersionAbbrev="COA Current"

PrevVersionAbbrev="COA Previous"
```

A configuration file can include multiple operations. Each operation is contained in a separate section with a unique label. A special section labeled [Operations] is used to specify the number of operation sections in the file and the label for each. The operations are performed in the order listed.

Below is an example of a Batch Client configuration file with multiple operations. Note that when using multiple operations, the operation key, for example Operation=Import, must be included in each section.



Note:

Sections, keys, and values are not case sensitive.

```
[General]
```

```
Operation=Multiple
```

```
UserName=mylogin
```

```
Password=mypassword
```

```
URL=net.tcp://localhost:5210/Oracle/Drm/Engine
```

```
LogFileName="C:\Oracle\Data Relationship Management\rdsc.log"
```

```
[Operations]
```

```
OperationCount=3
```

```
Operation1=Weekly_Import
```

```
Operation2=Update_CostCenters
```

```
Operation3=Merge_Data
```

```
[Weekly_Import]
```

```
Operation=Import
```

```
InFile="COA_data.txt"
```

```
ImportAbbrev="RDSC Import"
```

```
ImportLogFileName="C:\Oracle\Data Relationship Management\Batch
```

```
Client\import_log.txt"
```

```
VersionAbbrev="COA Current"
```

```
VersionDescription="COA for this month"
```

```
[Update_CostCenters]
```

```
Operation=Action Script
```

```
InFile=CostCenter_changes.txt
```

```
StripQuotedStrings=True
```

```
Delimiter=comma
```

```
[Merge_Data]
```

```
Operation=Blend
```

```
Blendname="COA Blender"
```

```
SourceVersionAbbrev="COA Current"
```

```
TargetVersionAbbrev="COA Previous"
```

```
NewVersionAbbrev="COA Blended"
```

```
NewVersionDescription="Current and Previous COA"
```

Command Line Parameters and Configuration File Keys

The tables in this section describe the command line parameters and configuration file keys that can be used in the Batch Client.

Table 17-1 General Commands

Key	Parameter	Description	Required
Operation	/op=	Specifies the operation to perform: <ul style="list-style-type: none"> • ActionScript • BackupVersionToFile • Blend • CloseVersion • DeleteVersion • Export • Import • Multiple • OpenVersion • RefreshVersion • Request • RestoreVersionFromFile 	Y
UserName	/u=	Specifies the Oracle Data Relationship Management user name for login.	Y
Password	/pw=	Specifies the Data Relationship Management password for login.	Y
URL	/url=	Specifies the URL (including the port number) for the Engine application in the Data Relationship Management server processes. This URL is configured and is available in the Data Relationship Management configuration console. For example, net.tcp://localhost:5210/Oracle/Drm/Engine	Y

Table 17-1 (Cont.) General Commands

Key	Parameter	Description	Required
LogFileName	/log=	<p>Specifies the path and file name for the log file. If the file exists, the log is appended to the end of the existing file. It is recommended that you put the value in double quotes.</p> <p>Caution: To run multiple Batch Client processes concurrently, you must specify a unique log name for each process.</p> <p>Note: The specified log file may not be created if the /log parameter is placed after a problematic parameter such as a parameter with a missing or invalid value. To ensure that the specified log file is created, the /log parameter should be placed as the first parameter.</p> <p>If not specified, the following name is used: <code>DRM_Batch_ClientMMD DYYYY.log</code>, where <i>MM</i> = month, <i>DD</i> = day, <i>YYYY</i> = year.</p>	N
N/A	/cfgfile=	<p>Specifies the path and file name of the configuration file describing desired options. It is recommended that you put the value in double quotes.</p>	Y, if using a configuration file
N/A	/h[elp] /?	<p>Displays a list of command-line parameters.</p>	N
TimezoneOffset	/timezoneid	<p>Local minutes offset from GMT. For example, U.S. CST offset minutes from GMT is -360. Used for rendering of dates/times in output.</p> <p>The default is the culture used by the server.</p>	N

Table 17-1 (Cont.) General Commands

Key	Parameter	Description	Required
CultureName	/culturename	Local culture name, in the format: <i>languagecode-country/regioncode</i> where <i>languagecode</i> is a lowercase two-letter code derived from ISO 639-1 and <i>country/regioncode</i> is an uppercase two-letter code derived from ISO 3166. For example, U.S. English is en-US. Affects both language and formatting of input data. Language specifically applies to informational and error messages logged by the Batch Client. Culture controls formatting of Date/Time/Float for output in operations such as Exports. The parameter has fallback values that default based on the Data Relationship Management application's culture setting and the Data Relationship Management host server's language and culture settings on the Service user.	N
ObjectAccess	/objectaccess	Object access method for accessing import, export, and blender objects. Valid values are Standard, System, or User or custom group. The default is Standard.	N
InstanceID	/instanceid=	Specifies the InstanceID of the Application to be used if multiple Applications were configured to use the <i>drm-batch-client-credentials.exe</i> functionality.	Y, if multiple Applications were configured to use the <i>drm-batch-client-credentials</i> functionality.

Table 17-2 Action Script

Key	Parameter	Description	Required
InFile	/infile=	Specifies the path and file name for the input file. It is recommended that you put the value in double quotes.	Y

Table 17-2 (Cont.) Action Script

Key	Parameter	Description	Required
Delimiter	/delim=	Specifies the delimiter. Specify one of the following values: <ul style="list-style-type: none"> • Tab • Comma • #<i>nnn</i> where # <i>nnn</i> represents any valid 3-digit ASCII character value (0–255). The default is Tab.	N
ColumnOrder	/cols=	Specifies the column order. If supplied, must contain all column identifiers. The default is A12345.	N
StripQuotedStrings	/stripq=	Specifies whether to remove quoted strings. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 The default is True.	N
PropertyRefByLabel	/propertyrefbylabel	To look up property references by label (instead of name), specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 Default is True.	N
Encoding	/encoding	Character encoding of input file as either an encoding name or encoding code page number. This parameter is optional. The default is UTF8.	N

Table 17-3 Backup Version To File

Key	Parameter	Description	Required
VersionAbbrev	/vabbrev=	Specifies the names of versions to backup; delimited by /delim if there are multiple versions.	Y
Connection	/conn=	Specifies the name of the external connection.	Y

Table 17-3 (Cont.) Backup Version To File

Key	Parameter	Description	Required
Filename	/filename=	Specifies the name of the backup file to create using the external connection (/conn).	Y
Delimiter	/delim=	Printable delimiter character. Uses a single character such as colon (:), comma (,), dash (-), and so on.	N, unless vabbrev is a list of versions

Table 17-4 Blend

Key	Parameter	Description	Required
BlendName	/bl=	Specifies the name of blender to run.	Y
SourceVersionAbbrev	/sver=	Specifies the name for the source version to blend.	Y
TargetVersionAbbrev	/tver=	Specifies the name for the target version to blend.	Y
NewVersionAbbrev	/nvera=	Specifies the name for the new version to create.	N
NewVersionDescription	/nverd=	Specifies a description for the new version. You must enclose the value in double quotes.	N
AutoSave	/save=	Specifies whether to save the blended version or leave it detached. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 The default is False. For information on saving versions, see Saving Versions in the Batch Client .	N
BlendLogFileName	/blendlog=	Specifies the name of the blend log file (not the same as the Batch Client log file). The default value is stored in the blend profile.	Y

Table 17-4 (Cont.) Blend

Key	Parameter	Description	Required
RunTimeParameters	N/A	<p>Specifies a comma-delimited list of runtime parameters. An example of the format is: RuntimeParameters=City=Dallas,State=Texas,ZipCode=75234.</p> <p>If the value contains a delimiter such as a comma, special characters, or double quotes then enclose the value in double quotes.</p> <p>Note: [CR/LF], [CR], [LF] are not allowed as part of the value.</p>	N

Table 17-5 Close Version

Key	Parameter	Description	Required
VersionAbbrev	/abbrev=	Specifies the name for the version to close.	Y

Table 17-6 Delete Version

Key	Parameter	Description	Required
VersionAbbrev	/vabbrev=	Specifies the name for the version to delete.	Y

Table 17-7 Export

Key	Parameter	Description	Required
ExportType	/xtype	<p>Specifies the type of export: individual export or book. Specify one of the following values:</p> <ul style="list-style-type: none"> E[xport] B[ook] 	Y
ExportName	/xname=	Specifies the name of the saved export to run.	Y if running a single report

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
OutFile	/outfile=	For exports configured to output to a client file, this parameter specifies the path and file name for the output file. If the output file exists, it is overwritten. It is recommended that you put the value in double quotes. If this parameter is specified and the export is configured to send output to an external server connection, this parameter overrides the external server connection and the export is written to the file specified here.	Y
CurrentVersionAbbrev	/cver=	Specifies the name for the current version. If not specified, the system uses the value of the DefaultCurrentVersion system preference.	Y, if no version variable is saved with the export
PrevVersionAbbrev	/pver=	Specifies the name for the previous version. Used for Hierarchy Compare and Hierarchy Difference exports. If not specified, the system uses the value of the DefaultPreviousVersion system preference.	Y, for Compare and Difference exports if no version variable is saved with them
CurrentVersionType	/cvertype=	Specifies the version type for the current version. Valid values are: <ul style="list-style-type: none"> • A[s Of] • N[ormal] The default is N[ormal].	N
PrevVersionType	/pvertype=	Specifies the version type for the previous version. Valid values are: <ul style="list-style-type: none"> • A[s Of] • N[ormal] The default is N[ormal].	N

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
CurrentAsOfDate	/casofdate=	Specifies the As Of Date/ Time to use when creating an As Of Version for the current version for the export. Note: Date/time formats are determined by the regional settings on the client computer.	Y, when version type is AsOf
PrevAsOfDate	/pasofdate=	Specifies the As Of Date/ Time to use when creating an As Of Version for the previous version for the export. Note: Date/time formats are determined by the regional settings on the client computer.	Y, when version type is AsOf
BookName	/bk=	Specifies the name of the saved book to run. It is recommended that you put the value in double quotes.	Y if exporting a book
CombineExportOutput	/bkcmb=	Specifies whether to combine the export output into the file named in / outfile or OutFile. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 If this parameter is False, each export is saved in the file name stored in its export profile. Exports for which the file name in the export profile is invalid or for which Output is set to a value other than File are combined into the file named in /outfileor OutFile. The default value is stored in book.	N
PreFileName	/prefile=	Specifies the path and file name to prepend to a combined export book. The default value is stored in book.	N

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
PostFileName	/pstfile=	Specifies the path and file name to append to a combined export book. The default value is stored in book.	N
FromHierarchy	/fhier=	Specifies the From (Previous) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.	N
ToHierarchy	/thier=	Specifies the To (Current) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.	N
FromTopNodeAbbrev	/fabbrv=	Specifies the name for the top node of the From (Previous) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.	N
ToTopNodeAbbrev	/tabbrv=	Specifies the name for the top node of the To (Current) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports	N
Hierarchy n	/hier=	Specifies the name for the hierarchy to export where n is a number (1 through HierarchyCount) identifying the current hierarchy/top node pair (for example, TopNodeAbbrev1). Must match Hierarchy n . Used for Hierarchy Export and other similar exports, but not used for Hierarchy Compare and Hierarchy Difference exports.	N

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
TopNodeAbbrev <i>n</i>	/nabbrev=	Specifies the name of the top node to export where <i>n</i> is a number (1 through HierarchyCount) identifying the current hierarchy/top node pair (for example, TopNodeAbbrev1). Must match Hierarchy <i>n</i> . Used for Hierarchy Export and other similar exports, but not used for Hierarchy Compare and Hierarchy Difference exports.	N
Encoding	/encoding	Character encoding of file as either an encoding name or encoding code page number. This parameter is optional. If an encoding is specified, it overrides the encoding saved with the export or book being run. The specified encoding also applies to the batch client's available local prefile and postfile options. Note: If a book is run, the encoding does not override the encoding of the exports within the book, only the encoding setting of the book itself. The default is UTF8.	N
HierarchyCount	/hier	Specifies the number of hierarchy/top node pairs defined.	N

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
HierarchyGroupPropFromParam	N/A	Specifies whether to get the hierarchy group property name from a parameter or a specified hierarchy group property. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 If True, specify a parameter name for HierarchyGroupProperty; if False, specify a hierarchy group property name for HierarchyGroupProperty. The default value is False.	N
HierarchyGroupProperty	N/A	Specifies a parameter name or a hierarchy group property name. If HierarchyGroupPropFromParam is True, a parameter name is expected; if False, a hierarchy group property name (including its namespace) is expected.	N
HierarchyGroupFromParam	N/A	Specifies whether to get the hierarchy group value from a parameter or a specified hierarchy group. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 If True, specify a parameter name for HierarchyGroup; if False, specify a hierarchy group name for HierarchyGroup. The default value is False.	N
HierarchyGroup	N/A	Specifies a parameter name or a hierarchy group name. If HierarchyGroupFromParam is True, a parameter name is expected; if False, a hierarchy group name is expected.	N

Table 17-7 (Cont.) Export

Key	Parameter	Description	Required
RunTimeParameters	N/A	Specifies a comma-delimited list of runtime parameters. An example of the format is: RuntimeParameters=City=Dallas,State=Texas,ZipCode=75234. If the value contains a delimiter such as a comma, special characters, or double quotes then enclose the value in double quotes. Note: [CR/LF], [CR], [LF] are not allowed as part of the value.	N
FromDate	/fdate=	Specifies the From date/time for transaction log exports Note: Date/time formats are determined by the regional settings on the client computer.	N
ToDate	/tdate=	Specifies the To date/time for transaction log exports Note: Date/time formats are determined by the regional settings on the client computer.	N

Table 17-8 Import

Key	Parameter	Description	Required
InFile	/infile=	For imports configured to import from a client file, this parameter specifies the path and file name for the input file. It is recommended that you put the value in double quotes. Caution: Do not use this parameter for imports configured to use an input file from an external server connection, otherwise an error occurs.	Y, if importing from a client file
ImportAbbrev	/iabbrev=	Specifies the name for the import to run.	Y

Table 17-8 (Cont.) Import

Key	Parameter	Description	Required
ImportLogFileName	/implog=	Specifies the path and file name of the import log file (not the same as the Batch Client log file.) The default file name is <code>DRM_Batch_ClientMMD DYYYY.log</code> , where <i>MM</i> = month, <i>DD</i> = day, <i>YYYY</i> = year.	Y
VersionAbbrev	/vabbrev=	Specifies the version name to be created by import. The default value is stored in the import profile.	N
VersionDescription	/vdescr=	Specifies the description of the version to be created by import. The default value is stored in the import profile.	N
AutoSave	/save=	Specifies whether to save the imported version or leave it detached. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 The default is False. For information on saving versions, see Saving Versions in the Batch Client .	N
RunTimeParameters	N/A	Specifies a comma-delimited list of runtime parameters. An example of the format is: <code>RuntimeParameters=City=Dallas,State=Texas,ZipCode=75234.</code> If the value contains a delimiter such as a comma, special characters, or double quotes then enclose the value in double quotes. Note: [CR/LF], [CR], [LF] are not allowed as part of the value.	N

Table 17-9 Multiple

Key	Parameter	Description	Required
OperationCount	N/A	Specifies the number of operation sections in the file. Specify an integer greater than 0.	Y
OperationN	N/A	Specifies the label for an operation section where <i>N</i> is a number between 1 and the OperationCount value.	Y

Table 17-10 Open Version

Key	Parameter	Description	Required
VersionAbbrev	/vabbrev=	Specifies the name of the version to open.	Y

Table 17-11 Refresh Version

Key	Parameter	Description	Required
VersionAbbrev	/vabbrev=	Specifies the name of the version to refresh (close and re-open).	Y

Table 17-12 Request

Key	Parameter	Description	Required
WorkflowModel	/WorkflowModel=	Specifies the name of a workflow model	Y
RequestTitle	/RequestTitle=	Specifies a request title	Y
Version	/vabbrev=	Specifies the name of a version	Y, if no version variable is specified on the workflow model
WorkflowTask	/WorkflowTask=	Specifies the name of a workflow task. Must be a task assigned to the Submit stage of the request.	Y
Hierarchy	/hier=	Specifies the name of a hierarchy.	Y, if not defaulted using hierarchy group
Connection	/conn=	Specifies the name of the external connection. Must refer to a server file connection accessible via the batch client user's object access groups.	N

Table 17-12 (Cont.) Request

Key	Parameter	Description	Required
FileName	/filename=	Specifies the name of the file to locate using the external connection. If Server File Connection Name is specified, then this parameter refers to a file available to the server connection	Y if Connection is specified
InFile	/infile=	Specifies the path and file name for the input file. It is recommended that you put the value in double quotes.	Y, if Connection is not specified
Delimiter	/delim=	Printable delimiter character. Uses a single character such as colon (:), comma (,), dash (-), and so on.	N
StripQuotedStrings	/StripQuotedStrings=	Specifies whether to remove quoted strings from data values found in the file. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 The default is True.	N
AllowBlankOverrides	/AllowBlankOverrides=	Specifies whether to allow blank overrides. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 	N
BlankValueIndicator	/BlankValueIndicator=	Specify text value to indicate to indicate blank values in the file.	N
DeleteOnSubmitFailure	/DeleteOnSubmitFailure=	Specifies whether to delete the request if the submit is not successful. Specify one of the following values: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 Default value is False.	N
Encoding	/encoding	Character encoding of input file as either an encoding name or encoding code page number. This parameter is optional. The default is UTF8.	N

Table 17-13 Restore Version From File

Key	Parameter	Description	Required
Connection	/conn=	Specifies the name of the external connection.	Y
Filename	/filename=	Specifies the name of the file to locate using the external connection (/conn).	Y
SourceVersionAbbrev	/sver=	Specifies the names of versions in the source file to restore; delimited by /delim if there are multiple versions.	Y
TargetVersionAbbrev	/tver=	Specifies the names of new versions, correlating to the /sver parameter; delimited by /delim if there are multiple versions.	Y
Delimiter	/delim=	Printable delimiter character. Uses a single character such as colon (:), comma (,), dash (-), and so on.	N, unless sver or tver are specified as list of versions
AutoSave	/save=	Save versions after restoring: <ul style="list-style-type: none"> • Y[es], T[rue], or 1 • N[o], F[alse], or 0 	N

Outputs and Result Codes

All Batch Client operations generate a log file. In addition, exports generate an exported data file and an export log file. The Batch Client logs warnings in the log file for any script items that fail. All errors are also recorded in the log file, including the complete error message. The log file is saved using the path/filename indicated with the /log startup parameter. If not specified, the log defaults to the name `DRM_Batch_Client_mmdyyyyy.log`, and is written to the same folder where the Batch Client is launched.

In addition, the Batch Client returns an error level (exit code or result code) indicating its success or failure. Error levels can be tested (via `ERRORLEVEL`, `%ERRORLEVEL%`, and so on) by a scheduling program or batch file and appropriate action can automatically be taken.

Table 17-14 Error Level Values

Error Level Value	Description
0	Successful/normal termination
1	Unknown error
2	Batch operation completed but generated warnings
3	Batch operation did not complete due to error

Table 17-14 (Cont.) Error Level Values

Error Level Value	Description
100	Error communicating to or returned from the Oracle Data Relationship Management Server
200	Error occurred during initialization of the Batch Client
210	An invalid parameter value was passed
220	Invalid URL
230	Invalid user name or password

 **Note:**

If other error level values are returned, the error is from the Windows operating system. See Windows documentation for more information.

Using Data Relationship Management Analytics

The Oracle Data Relationship Management Analytics module provides multiple dashboards and reports to allow data managers to measure performance, monitor changes, and mitigate issues. Data Relationship Management Analytics provides:

- Dashboards to track master data and governance request metrics
 - [Change Management](#)—Analyze change activities based on version, hierarchy, node type, property, and workflow model, including governance vs. interactive changes
 - [Growth Analysis](#)—Analyze how versions and hierarchies have changed over time by reviewing total number of nodes, number of orphan nodes, and number of shared nodes
 - [Request Monitoring](#)—Monitor in-flight change requests, identify bottlenecks and requests that are overdue or near due, and drill-back into Oracle Data Relationship Governance requests.
 - [Model Analysis](#)—Analyze workflow model design, identify participant behavior trends, view resource workload and time committed, and drill-back into Data Relationship Governance requests.
- Reports to monitor user and group membership, assignments and authorization, user activity, and metadata object usage
 - [User Role Assignment](#) — Provides a list of users by role or roles by user with counts by licensed user types
 - [Access Group Membership Report](#) — Provides a list of users by interactive and workflow user groups
 - [Object Access Group Authorization Report](#) —Provides mapping of users and user groups to specific Oracle Data Relationship Management objects
 - [Hierarchy Access Group Assignment Report](#) — Provides data grants of users and groups to nodes in a hierarchy
 - [Workflow Access Group Assignment Report](#) — Provides data grants of users and groups to workflow model stages
 - [User Login Activity Report](#) — Provides trend reports for user login activity over time
 - [Metadata Object Usage Report](#) — Provides frequency distribution and aging information for Data Relationship Management objects: queries, compares, imports, exports, blenders, and books

Execution plans are used to prepare data for analysis and are run with a built-in job scheduler.

 **Note:**

After running jobs to generate data, any dashboards or reports that are currently open do not automatically refresh until some user action, such as changing the filter criteria, is performed.

Drill-through from dashboards to Data Relationship Governance requests is provided. Information from dashboards and reports can be exported in .XLS format for further analysis.

 **Note:**

Data Relationship Management Analytics detects the language used by the Web browser to determine the language in which to display the module. Culture is not supported as in Data Relationship Management, therefore numbers may display differently in Data Relationship Management Analytics.

Version Lineage

The version lineage feature of Oracle Data Relationship Management captures the source version each time a copy of the version is made. This allows you to trace the history of any version back to its original source and provide interesting ways to analyze changes based on version lineage.


Version lineage is set in Data Relationship Management. To set up lineage for a version, edit the Lineage Version property field to assign the version from which lineage should be built. If you have copied a version, the copied version populates the Lineage Version field. If you delete the lineage version that a version points to, then the version no longer has a lineage version assigned. For more information, see [Editing Version Properties](#).

Setting up version lineage allows Oracle Data Relationship Management Analytics to aggregate changes across lineages and across multiple versions.

Using Filters

All dashboards and reports use filters. Filters are specific to a user and are not data dependent, therefore a combination of filters can return no data. You can save, clear, or restore the filter selections for dashboards and reports.

To save, clear, or restore filters:

1. Click .
2. Select an option:
 - **Save Filter**—Saves the current filter selections
 - **Clear Filter**—Clears the current selections from the filter and resets the selection to the default

- **Restore Filter**—Resets the filter selections to the default or to the last saved filter

 **Note:**

The Growth Analysis dashboard does not have a filter bar but your version selections can be saved or restored.

Change Management Dashboard

The Change Management dashboard provides aggregated views of node and property changes that have occurred in the Oracle Data Relationship Management system over time. Metrics in this dashboard are based on committed requests and all interactive requests. Information displayed includes number of added, inserted, inactivated, moved, removed, and deleted nodes as well as property updates. Access to the Change Management dashboard is based on a user's read access to versions. Users will see all hierarchies in a version as long as they have access to at least one hierarchy in the version. Data Managers can see all versions.

The Change Management dashboard provides a summary pane on the left side of the dashboard containing:

- A breakdown of changes by change type displayed in a pie chart. Selecting a specific change type in the pie chart filters the list and chart views for that change type.
- A breakdown of property updates by category displayed in a list. Only categories to which you have access are displayed. Property updates and removes are combined for this number. Version and hierarchy property changes are not tracked. Selecting a specific property category filters the list and chart views for that category.

 **Note:**

Because properties may exist in multiple categories, the number of changes per category may exceed the total number of property changes displayed.

- The total number of node and property changes with a timestamp for when the data was last refreshed. This information is read only.


The right side of the dashboard displays change information based on selected filter options. Information can be displayed in either List View or Graph View. From the List View, you can click the bars or names to drill into the specific change. The Graph View displays the change distribution and trend with duration (as set by the timeframe filter) on the x axis and change count on the y axis.

 **Note:**


Historical data before release 11.1.2.4.320 does not have a node type set and will display as "Unknown".

Viewing Change Metrics


To view change metrics:


1. From Oracle Data Relationship Management Analytics, select **Change**.
2. Click the filter bar and set filter options. Click  to select versions, hierarchies, node types, and workflow models.
 - **Version**—Select one or more versions. You can see only the versions to which you have access
 - **Version Lineage**—If selected then only data from versions selected in the filter are displayed.
 - If selected and the head version of a given lineage change is also selected in the filter, then data from all versions in the lineage change are displayed.
 - If selected and the head version is not selected in the filter, then no data from any version in the lineage is displayed regardless of which other versions are selected.
 - **Hierarchy**—You can select hierarchies only when a single version is selected.
 - **Node Type**—Select a node type.
 - **Change Method**—Select from these methods: All, Interactive, and Workflow.
 - **Workflow Model**—You can select workflow models only when the Workflow change method is selected.
 - **Timeframe**—Select a predefined timeframe or select **Custom**, and then select a **Start Date** and **End Date**.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. Set how you want to view metrics in the Change Management dashboard:

- Click  **List View**—Select from these options:
 - View By
 - * Hierarchy
 - * Node Type
 - * Workflow Model—Available only when Change Method is set to Workflow
 - * Property—Available only when you select Property Changes in the summary pane

- Sort By
 - * Count
 - * Name
- Click  **Graph** to view the change distribution and trend chart. Duration (as set by the timeframe filter) is displayed on the x axis and number of changes within the timeframe on the y axis.
- Set how you want the chart to display:
 - If the selected change method is All, then the chart displays three trend lines:
 - * Workflow trend line
 - * Interactive trend line
 - * Cumulative trend line for both workflow and interactive changes
 - If Workflow or Interactive is selected for the change method, then only a single trend line for that method is displayed.
- Drill into a change method for additional information.

Drilling is available only when the duration is set to weeks or months. Click on a data point to drill into a specific month/week and change method. If duration is set to days, then drilling is not available.

Growth Analysis Dashboard

The Growth Analysis dashboard allows you to analyze how versions and hierarchies have changed over time by displaying total number of nodes, number of orphans and shared nodes, and the increase in nodes over the last 30 days. This dashboard displays only the versions to which you have Read access. Users will see all hierarchies in a version as long as they have access to at least one hierarchy in the version. Only node access groups are considered, not workflow node access group access.


The Growth Analysis dashboard provides a list of versions on the left with corresponding graphs of node metrics. Bar charts are used for lineaged versions and line charts with points for each available snapshot are used for non-lineaged versions. You can hover over a bar or point in either chart for detailed information. The right side of the dashboard displays node counts for total nodes, orphans, shared nodes, and the increase or decrease in the node count. For lineaged versions, this count is the increase or decrease from the previous version. For non-lineaged versions, this count is the increase or decrease in the last 30 days. The group of versions can be sorted by whether a version is lineaged or not, version name, or total node count.

Click the name of a version to drill into node metrics by hierarchy for the selected version. You can filter the single version view by hierarchy group and sort by name or node count. Select a different version from the drop-down list in the top left corner of the dashboard or, to return to the full list of versions, click the **# Versions** link.

Viewing Growth Metrics


To view growth metrics:

1. From Oracle Data Relationship Management Analytics, select **Growth**.


2. If you have more than five versions, you are prompted to select versions. Click **Select Versions**, click , select versions, click **OK**, and then click **Apply**.

 **Note:**

If you have already saved a filter then the versions selected previously are displayed.

3. Choose from these options for displaying metrics:
 - To add or remove versions, click **Select Versions**, click , select versions, click **OK**, and then click **Apply**.
 - To drill into node metrics by hierarchy, click a version name.
 - To sort the list of versions, from **Sort by**, select Lineage, Name or Count.

 **Tip:**

Click  to save the current filter settings or reset the filter to the default settings.

Request Monitoring Dashboard

The Request Monitoring dashboard displays key performance indicators as they relate to open Oracle Data Relationship Governance requests which do not have a status of Draft, Withdraw, Rejected, or Committed. It provides a convenient way to identify outliers and problem areas and allows for drill back into Data Relationship Governance requests. Data Governance users have access to all requests to which they have commit access. Data Managers have access to all requests in the system.

The left side of the Request Monitoring dashboard provides a breakdown showing the number of requests that are:

- Open
- Near due—Requests due within 3 days
- Urgent
- Escalated
- Overdue
- Claimed
- Unclaimed

Corresponding details display in the center of the dashboard. Request details can be displayed in Graph View, Scatter Graph View, or List View.

Viewing Request Metrics

To view request metrics:


1. From Oracle Data Relationship Management Analytics, select **Request**.
2. Click the filter bar and set filter options:

 **Note:**



Filter options are derived from the in-flight requests in the system.






- **Model**—Select individual models or select **All**.
Only models to which you are related are displayed.
 - **Access Group**—Select individual access groups or select **All**.
 - **Version**—Select individual versions or select **All**.
 - **Submitter**—Select individual submitters or select **All**.
 - **Start Date**—Enter a start date. This is the request submit date.
 - **End Date**—Enter an end date.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. Set how you want to view metrics in the Request Monitoring dashboard:

-  **Graph View**—You can view requests by model or user group. Click Model or User Group, then click on a model or user group to drill into it. Click on a model to display the breakdown of counts by stage. All stages of the model are displayed even if there are no requests in that stage. Click on a user group name to drill into the user chart which displays the breakdown of counts by user name.
-  **Scatter Graph View**—You can view requests by model or user group. Click Model or User Group, then click on a model or user group to drill into it. The x-axis of the chart displays the relative due date for each request. A number greater than 0 indicates that the request is overdue. A number less than 0 indicates the number of days in which the request will be due. The y-axis is the actual age of the request since request submission. Click on a request on the chart to drill into additional request metadata and details and to be able to drill back into the Oracle Data Relationship Governance request.

-  **List View**—You can view requests in a list with the ability to sort, reorder, and select columns to display.
 - To view additional request metadata and be able to drill back into the Data Relationship Governance request, click on a request ID and then click .
 - To detach the table of requests to make it larger and easier to use, click  or click **View** and then **Detach**.
 - To filter columns using query by example, click  or click **View** and then **Query by Example**.
 - To set up columns:
 - a. Click **View** and then **Columns**.
 - b. Select from these options to set up columns:
 - * **Show All**—To display all columns in the view.
 - * Select or deselect columns for the view.
 - * **Manage Columns**—Move columns from the **Hidden Columns** list to the **Visible Columns** list, and then click **OK**.
 - To reorder columns:
 - a. Click **View** and then **Reorder Columns**.
 - b. Select one or more columns and then use the up and down arrows to move the columns to the new location.
 - c. Click **OK**.
 - To export the list view results in .XLS format, click  and save the file.

Model Analysis Dashboard

The Model Analysis dashboard provides analysis of Oracle Data Relationship Governance workflow model design by providing historical performance of requests that are completed. Completed requests are requests that have been committed or rejected. Requests that are in process are not included in the analysis. The dashboard allows you to identify participant behavioral trends, request activity trends, resource workload, and time committed. Data Governance users have access to all requests to which they have commit access. Data Managers have access to all requests in the system. The dashboard provides the ability to drill back into Data Relationship Governance requests and to export to an .XLS format for further analysis.


The Model Analysis dashboard provides a summary pane on the left side of the dashboard containing:

- The number of requests that fit the filter criteria
- A breakdown of requests by due status. Requests in the pie chart are organized into three groups:
 - Requests with no due date


- Requests that completed on time
- Requests that were overdue when completed
- Cycle time —Average time it took for the requests to complete
- Participants—Average number of users who participated in the requests

Viewing Model Metrics




To view model metrics:

1. From Oracle Data Relationship Management Analytics, select **Model**.
2. Click the filter bar and set filter options:
 - **Model**—Click , select models, and then click **OK**.
 - **Model Status**— Select a status:
 - Active and Hidden
 - Hidden Only
 - Active Only
 - **Timeframe**—Select a predefined timeframe or select **Custom**, and then select a **Start Date**. **End Date** is optional.
3. Click **Apply**.

Tip:

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. Set how you want to view metrics in the Model Analysis dashboard:



- **Model Performance View**  —You can view performance in the List View  or the Service Level Agreement Plot view . In the List View, models matching the filter selections are displayed with the following information: number of requests, average cycle time for the requests to complete, average number of request items, and average number of participants. The color of the bars in the List View correspond to the colors in the pie chart. Click the name of a model to drill into specifics for the model. The List View can be sorted by Count, Avg Cycle Time, or Label. Cycle time is the request completion time minus the request submit time. When sorted by Avg Cycle Time, the model with the largest average cycle time is listed first.

In the List View, click on a model name to drill into the model. In the Service Level Agreement plot, click on a point in the graph to drill into the model. The following tabs are available in the Model Performance View:



- **Model Overview** tab —Provides overview information for the model and allows you to drill back into requests in Oracle Data Relationship Management.

The following information is provided on the Model Overview tab:


- * Clickable segmented color bar to drill into requests based on due status—After clicking the bar, click  to export results to an Excel spreadsheet or click  next to a request to drill back into the request in Data Relationship Management.

After view content on this screen, click Back in the top right corner of the screen to return to the Model Overview screen. Do not use the browser Back button.

- * Request Duration—Request duration setting value for the filtered model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
- * Claim Duration—Current claim duration setting value for the filtered model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
- * Unclaimed Duration—Percentage of time spent in unclaimed status across all requests for the selected model
- * Cycle Time—Average time it took all requests for the selected model to complete
- * Participants—Average number of unique participants across all requests for the selected model
- * Conditional Stages—Number of stages in the model that are conditional. For example, in a 4 stage model that has 3 conditional stages, "3 of 4" is displayed.
- * Separation of Duties Enabled—Number of stages for which separation of duties is enabled
- * SLA Violations—Number of requests that exceeded the request duration or due date at the request level
- * Slippages—Number of requests that had at least one claim duration violation
- * Pushbacks—Number of requests that had at least one push back
- * Escalations—Number of requests that had at least one escalation
- * Auto-skipped—Number of requests with at least one conditional Enrich or Approve stage that was skipped
- * Auto-approved—Number of requests with at least one Enrich or Approve stage that was promoted (approved by the system)
- * Auto-committed—Number of requests with a skipped or promoted commit stage

 **Note:**

Auto-Committed counts are not included in the Auto-Skipped or Auto-Approved counts.


- **Process Efficiency** tab  —Provides a breakdown analysis for each stage in a workflow model. Identifies the bottlenecks, resource requirements, and participant workload.


The following metrics are provided for the filtered model:


- * Average cycle time
- * Percentage of claimed vs. unclaimed duration
- * Request duration
- * Request claim duration
- * Total # of requests
- * Total resources


The following metrics are provided for each stage of the filtered model:



- * Average cycle time
- * Percentage of cycle time the stage was idle.
- * Number of requests that were delayed (where the claim duration had elapsed)
- * Number of person hours for the resources committed
- * Average number of participants

- **Throughput** tab  —Shows a graphical representation of the submitted requests for the selected model.

- **Participant** tab  —Shows participant metrics for a single model. The views can be filtered to include a minimum number of requests or items. Both views can be sorted by the percentage committed, request items, or participant name. You can click on the segmented bar or the color coded value to drill into the request list for rejected or committed requests.

- **Request Throughput View**  —Shows a graphical representation of the submitted requests for the filtered models. The default view shows metrics for all submitted, committed, and rejected requests for the selected timeframe. You can filter the display to show only Committed, Rejected, or Submitted requests.


- **Participant Performance View**  —Shows metrics by two views: Submitted By and Committed By. You can determine which participants are processing the most or

fewest number of requests. Click  to view metrics based on users who submitted requests. Click  to view metrics based on users who completed requests. The views can be filtered to include a minimum number of requests or items. Both views can be sorted by the percentage committed, request items, or participant name.


User Role Assignment Report


This report provides a list view and a graph view of the numbers of users assigned to user roles.



To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **User Role Assignment Report**.
2. Click the filter bar and set filter options:
 - **Users**—Select **All** or click , select users, and then click **OK**.
 - **Roles**— Select **All** or select roles.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.


4. Set how you want to view information in the report:
 -  **Graph View**—Number of users assigned to each user role and counts for these user types:
 - **Governance Users**—Users with the Governance User role
 - **Data Stewards**—Users with any of these roles:
 - * Interactive User
 - * Data Creator
 - * Data Manager
 - * Application Administrator
 - * Access Manager
 - **Analytics Users**—Users with any of these roles:
 - * Analytics User
 - * Governance Manager
 - **Anonymous Users**—Users with the Anonymous User role

-  **List View**—View the list by user or role. Click  to export the report to .XLS format.


Access Group Membership Report

This report provides a list view of users by interactive and workflow user groups.

To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **Access Group Membership Report**.
2. Click the filter bar and set filter options:
 - **Users**—Select **All** or click , select users, and then click **OK**.
 - **Access Group**— Select **All** or select access groups.
 - **Group Type**—Select Interactive, Workflow, or Both.
 - **Scope**—Select Local, Global, or Both
3. Click **Apply**.

Tip:

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. **Optional:** Click  to export the report to .XLS format.

Object Access Group Authorization Report


This report provides a list view mapping of users and user groups to specific Oracle Data Relationship Management objects.

Note:


A user with the Access Manager role has access to this report and can see all Object Access Groups regardless of which groups that user can see in Data Relationship Management.

To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **Object Access Group Authorization Report**.
2. Click the filter bar and set filter options:
 - **Object Access Group**—Select **All** or select groups.

- **Users**—Select **All** or click , select users, and then click **OK**.
 - **Access Group**— Select **All** or select access groups.
 - **Object Type**—Select **All** or select objects. Default is None.
3. Click **Apply**.

 **Tip:**


Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. **Optional:** Click  to export the report to .XLS format.


Hierarchy Access Group Assignment Report


This report provides a list view displaying data grants of users and groups to nodes in a hierarchy.

To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **Hierarchy Access Group Assignment Report**.
2. Click the filter bar and set filter options:
 - **Version**—Select a version.
 - **Hierarchy**— Select **All** or select hierarchies.
 - **User**—Select **All** or click , select users, and then click **OK**.
 - **Access Group**—Select **All** or select access groups.
 - **Access Group Type**—Select Interactive, Workflow, or Both.
 - **Access Level**—Select All or select access levels.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. Set how you want to view the list: by User or Hierarchy.
5. **Optional:** Click  to export the report to .XLS format.



Note:

For global node access groups, this report shows only the controlling hierarchy where the access is assigned. The user may have access to additional hierarchies to which they have been granted global access.

Workflow Access Group Assignment Report



This report provides a list view displaying data grants of users and groups to workflow model stages.



Caution:


Models using the same label will not be distinguishable from each other in the Workflow Access Group Assignment Report.


To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **Workflow Access Group Assignment Report**.
2. Click the filter bar and set filter options:
 - **Model**—Select **All** or click , select models, and then click **OK**.
 - **Stage**—Select **All** or select stages.
 - **Users**—Select **All** or click , select users, and then click **OK**.
3. Click **Apply**.



Tip:

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.


4. Set how you want to view the list: by Model or User.
5. **Optional:** Click  to export the report to .XLS format.

User Login Activity Report


This report provides a list view displaying trend reports for user login activity over time.

To run the report:


1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **User Login Activity Report**.

2. Click the filter bar and set filter options:
 - **Timeframe**—Select a predefined timeframe or select **Custom**, and then select a **Start Date** and **End Date**.
 - **Users**—Select **All** or click , select users, and then click **OK**.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. Set how you want to view information in the report:

-  **Graph View**—Scale for Unique Logins is displayed on the left and the scale for Total Logins is displayed on the right.

 **Note:**


For login results to display in this view, there must be at least two logins in the timeframe selected. For fewer than two logins, view the results in the List View.

-  **List View**—Click  to export the report to .XLS format.


Metadata Object Usage Report


This report provides frequency distribution and aging information fro Oracle Data Relationship Management objects: queries, compares, imports, exports, blenders, and books.

To run the report:

1. From Oracle Data Relationship Management Analytics, select **Reports**, and then select **Metadata Object Usage Report**.
2. Click the filter bar and set filter options:
 - **Object Type**—Select an object: Exports, Books, Compares, Queries, Imports or Blenders.
 - **Timeframe**—Select a predefined timeframe or select **Custom**, and then select a **Start Date** and **End Date**.
 - **Users**—Select **All** or click , select users, and then click **OK**.
3. Click **Apply**.

 **Tip:**

Click  to save the current filter settings, reset to default saved filter settings, or clear the current filter selections.

4. **Optional:** Click  to export the report to .XLS format.

A

Dashboard Metrics

Change Management Dashboard

This dashboard provides aggregated views of node and property changes that have occurred in the Oracle Data Relationship Management system over time. Metrics in this dashboard are based on committed requests and all interactive changes. This dashboard includes change actions such as adds, updates, moves, and deletes across node and property changes to lend change perspectives by hierarchy, node type, property category, and so on. Users can understand change trends by change method to ratify governance uptake. Users can drill into each change contextually to inspect transaction details and export these details to a flat file for further analysis offline. This dashboard allows users to drill down to request level and transaction level detail.

This dashboard can be filtered by:

- Version—A secured filter that displays only the versions for which the logged in user has read access. Data Managers can view all versions.
- Hierarchy—A dependent filter that highlights hierarchies within the selected versions
- Node Type—An independent filter that lists all configured node types
- Change Method—A filter that allows breakdown of changes based upon workflow, interactive or both methods
- Workflow Model—A filter that displays all workflow models, both active and inactive
- Timeframe— A filter to select pre-defined date ranges or a custom date range to reflect when changes were performed interactively in a version or committed to a version via governance workflow

The metrics displayed in the dashboard are dependent on the filter selections.

Table A-1 Metrics for Change Management Dashboard

Metric	Description
List View	

Table A-1 (Cont.) Metrics for Change Management Dashboard

Metric	Description
Node Changes	<p>Total number of node changes broken down in a pie chart by change type. This metric tracks nodes that have been:</p> <ul style="list-style-type: none"> • Added • Inserted • Moved • Inactivated • Removed • Deleted <p>The colors in the pie chart correspond to the colors in the segmented bars in the List View. Each change type in the pie chart is clickable to filter the List View to show node changes for the selected type. Each change type in the segmented bar is clickable to drill into the transactions associated with the selected type.</p>
Property Changes	<p>Total number of property changes broken down by property category. Property updates and removes are combined in the total number. This metric is secured to show only properties to which the logged in user has access.</p> <p>Note: Because properties may exist in multiple categories, the number of property changes per category may exceed the total number of property changes displayed.</p> <p>Each property category is clickable to filter the List View to show property changes for the selected category. The gray bar in the List View is clickable to drill into the transactions associated with the selected category.</p>
Total Changes	Total number of node and property changes.
Graph View	
Workflow	The blue color in the segmented bar indicates the percentage of requests that were submitted via the workflow method. The blue points on the graph indicate the number of requests that were submitted via the workflow method.
Interactive	The orange color in the segmented bar indicates the percentage of requests that were submitted via the interactive method. The orange points on the graph indicate the number of requests that were submitted via the interactive method.

Growth Analysis Dashboard

This dashboard provides analysis of how versions and hierarchies have changed over time by displaying the total number of nodes, the number of orphan and shared nodes, and the total increase or decrease in nodes from a previous version (for lineaged versions) and the total increase or decrease in the last 30 days for non-lineaged

versions. This dashboard allows users to drill from the version level to the hierarchy level. This dashboard does not use filters.

Table A-2 Metrics for Growth Analysis Dashboard

Metric	Description
Total Nodes (version)	Total number of unique nodes in the version
Total Nodes (hierarchy)	Total number of nodes in the hierarchy
Orphans	Number of orphan nodes in the version
Shared	Number of shared nodes in the version or hierarchy
Increase/Decrease	Non-lineaged versions: increase or decrease in the version or hierarchy in the last 30 days Lineaged versions: increase or decrease in the version or hierarchy from the previous version

Request Monitoring Dashboard

The Request Monitoring dashboard displays key performance indicators as they relate to open Oracle Data Relationship Governance requests allowing you to identify bottlenecks and requests that are overdue or near due, and provides the ability to drill back into Data Relationship Governance requests to make changes to a request. This dashboard allows users to drill down to request level and transaction level detail. Users can further drill back into Data Relationship Governance to perform actions (for example, a user can unclaim a request that is claimed and overdue so others within the assigned access group can claim and work on it.

This dashboard can be filtered by:

- Model—A secured filter that displays only those models to which the user has Commit access. Data Managers can view all workflow models with in-flight requests.
- Access Group—A filter that displays node access groups for which there are active requests
- Version—A filter that displays only versions with active, in-flight requests
- Submitter—A filter that displays all Submit stage users who have active in-flight requests
- Start Date—Filters requests based on submit date
- End Date—Filters requests based on submit date



Note:

Either the start date, end date, or both may be provided.

The metrics displayed in the dashboard are dependent on the filter selections.

Table A-3 Metrics for Request Monitoring Dashboard

Metric	Description
Summary Metrics	
Open	Number of open requests
Near Due	Number of requests due within 3 days
Urgent	Number of requests marked urgent by a Submit stage user
Escalated	Number of requests that have been either automatically escalated by the system or manually escalated by a participant.
Overdue	Number of requests that are overdue. Derived based on a combination of submit date and the request duration (set at the model level). Participants may also set a Due Date tag in a request. When set, this date overrides workflow model defaults. If the request duration is not set, then the request is only included in this count if a Due Date has been set.
Claimed	Number of claimed requests for the selected metric - Open, Near Due, Urgent, Escalated, or Overdue.
Unclaimed	Number of unclaimed requests for the selected metric - Open, Near Due, Urgent, Escalated, or Overdue.
Request Count by Stage	
Requests	Number of requests in this stage
Claimed	Number of claimed requests in this stage
Unclaimed	Number of unclaimed requests in this stage
Stage Type	Number of requests in each stage type, for example Submit, Approve, Commit.
Request Count by User	
Requests	Number of requests for this user
Claimed	Number of claimed requests for this user
Unclaimed	Number of unclaimed requests for this user
Stage Type	Number of requests in each stage type, for example Submit, Approve, Commit, for this user
Request Count by Model	
Requests	Number of requests for this model
Claimed	Number of claimed requests for this model
Unclaimed	Number of unclaimed requests for this model
Stage Type	Number of requests in each stage type, for example Submit, Approve, Commit, for this model
Request Count by Access Group	

Table A-3 (Cont.) Metrics for Request Monitoring Dashboard

Metric	Description
Requests	Number of requests for this access group
Claimed	Number of claimed requests for this access group
Unclaimed	Number of unclaimed requests for this access group
Stage Type	Number of requests in each stage type, for example Submit, Approve, Commit, for this access group

Model Analysis Dashboard

This dashboard provides analysis of Oracle Data Relationship Governance workflow model design by displaying historical performance of requests that are completed (committed or rejected), including participant behavior trends, resource workload, and the ability to drill-back into Data Relationship Governance requests. It reports on performance of completed requests processed by each workflow model to understand model performance based on service level agreements, level of automation achieved, cycle time, resources committed, request workload, throughput, and participant engagement.

This dashboard can be filtered by:

- **Model**—A secured filter that displays only the models to which the logged in user has commit access. Data Managers can view all workflow models.
- **Model Status**—This filter allows end users to show active or hidden workflow models or both
- **Timeframe**—A filter to select pre-defined date ranges or a custom date range over which to evaluate completed requests associated with each workflow model

The metrics displayed in the dashboard are dependent on the filter selections.

Table A-4 Metrics for Model Analysis Dashboard

Metric	Description
Summary Metrics - Completed Requests	
Metrics in this section are for all models meeting the filter criteria.	
Count	Total number of completed requests, comprised of the number of on time requests, overdue requests, and requests with no due date
On Time	Percentage of requests that completed within the due date timeframe
Overdue	Percentage of requests that exceeded the due date
No Due Date	Percentage of requests that had no due date

Table A-4 (Cont.) Metrics for Model Analysis Dashboard

Metric	Description
Cycle Time	Average time it took all requests to complete. Cycle time of an individual request is computed as request completion time minus request submit time. Request completion time is when the request is either committed or rejected and reaches a terminal status. The average cycle time is an aggregate of time divided by the number of requests.
Participants	Average number of unique participants across all requests. For each request, a count of the number of unique participants is maintained. Submitters, enrichers, and approvers are counted as participants. Each participant is counted only one time, even if they touch a request multiple times regardless of the stage. Whenever a user claims a request, they are assumed to be a participant.
Model Performance - All Models - List View	
Metrics in this section are specific to individual workflow models. The model name above the segmented color bar is clickable to view metrics for that specific model.	
Requests	Total number of requests for the model. A segmented color bar indicates the number of requests for each category.
Cycle Time	Average time it took for the requests in the model to complete. Computed as request completion time minus request submit time. Request completion time is when the request is either committed or rejected and reaches a terminal status.
Request Items	Average number of request items per request in the model
Participants	Average number of unique participants for the requests in the model
Overview tab of Model Inspector	
Metrics in this section are specific to the model that was selected in the List View	
Requests	Number of requests for the model that meet the filter criteria
Request Duration	Request duration setting value for the model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
Claim Duration	Current claim duration setting value for the filtered model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
Unclaimed Duration	Percentage of time spent in unclaimed status across all requests for the selected model
Cycle Time	Average time it took all requests for the selected model to complete

Table A-4 (Cont.) Metrics for Model Analysis Dashboard

Metric	Description
Participants	Average number of unique participants across all requests for the selected model
Conditional Stages	Number of stages in the model that are conditional. For example, in a 4 stage model that has 3 conditional stages, "3 of 4" is displayed.
Separation of Duties Enabled	Number of stages for which separation of duties is enabled
SLA Violations	Number of requests that exceeded the request duration or due date at the request level
Slippages	Number of requests that had at least one claim duration violation
Pushbacks	Number of requests that had at least one push back
Escalations	Number of requests that had at least one escalation
Auto-skipped	Number of requests with at least one conditional Enrich or Approve stage that was skipped
Auto-approved	Number of requests with at least one Enrich or Approve stage that was promoted (approved by the system)
Auto-committed	Number of requests with a skipped or promoted commit stage Note: Auto-Committed counts are not included in the Auto-Skipped or Auto-Approved counts.
Process Efficiency tab of Model Inspector - Model Overview Section	
Metrics in this section are specific to the model that was selected in the List View	
Average model cycle time	The average cycle time for all requests in the model
Unclaimed Duration	Percentage of time unclaimed for all requests in the model
Claimed Duration	Percentage of time claimed for all requests in the model
Request duration	Request duration setting value for the model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
Claim duration	Current claim duration setting value for the model. The value displayed is dynamic and will change if changed within the model. Displayed for reference when viewing additional metrics.
Total requests	Total number of requests that meet the filter criteria

Table A-4 (Cont.) Metrics for Model Analysis Dashboard

Metric	Description
Total resources	Total number of resources (in full-time equivalent hours) for the model. This number is based on the total time the request spent in claimed status normalized using a 2000 hour per person year metric.
Process Efficiency tab of Model Inspector - Details by Stage Section	
Metrics in this section are specific to the model that was selected in the List View	
Avg Stage Cycle Time	Average amount of time that the requests spent in the specified stage
Cycle Time	Average time it took all requests to complete in the specified stage
Delayed	Number of requests that were delayed in this stage. Delayed requests are defined as requests with the claim duration elapsed.
Resources Committed	Number of person hours across all requests for the specified stage
Participants	Average number of unique participants across all requests for the specified stage
Throughput tab of Model Inspector	
Metrics in this section are specific to the model that was selected in the List View	
Submitted	Number of requests submitted for the model. Submitted requests include both committed and rejected requests. To get this count, the timeframe filter is applied on the submit date.
Committed	Number of requests committed for the model. To get this count, the timeframe filter is applied on the commit date.
Rejected	Number of requests rejected for the model. To get this count, the timeframe filter is applied on the reject date.
Participant Performance tab of Model Inspector - Submitted By View	
Metrics in this section are specific to the model that was selected in the List View and are cumulative for all participants.	
Requests Submitted	Number of requests submitted by model participants
% Committed	Percentage of the submitted requests that were committed
Rejected	Number of requests that were rejected
Committed	Number of requests that were committed
Request Items	Cumulative number of request items for all requests submitted
Participant Performance tab of Model Inspector - Committed By View	
Metrics in this section are specific to the model that was selected in the List View and are specific to individual participants.	
Requests Completed	Number of requests completed by the user

Table A-4 (Cont.) Metrics for Model Analysis Dashboard

Metric	Description
% Committed	Percentage of the completed requests that were committed
Rejected	Number of requests that were rejected
Committed	Number of requests that were committed
Request Items	Cumulative number of request items for all requests completed by the user
Model Performance - All Models - Service Level Agreement Plot Metrics in this section are for all selected models.	
On Time	The line on the SLA plot indicating requests that are on time. The x-axis plots the average cycle time of all the requests relative to the request duration. The request duration for each model is used to create the reference line and cycle time of each model is plotted with that reference. In general, any dots to the right side of the reference line indicate the model is not performing as expected.
Request Throughput - All Models Metrics in this section are for all selected models.	
Submitted	Number of requests submitted for all selected models. Submitted requests include both committed and rejected requests.
Committed	Number of requests committed for all selected models
Rejected	Number of requests rejected for all selected models
Participant Performance - All Models - Submitted By View Metrics in this section are for all selected models and are cumulative for all participants. Note: Participant performance is presented based on either Submit or Commit user context.	
Requests Submitted	Number of requests submitted by model participants
% Committed	Percentage of the submitted requests that were committed
Rejected	Number of requests that were rejected
Committed	Number of requests that were committed
Request Items	Cumulative number of request items for all requests submitted
Participant Performance - All Models - Committed By View Metrics in this section are for all selected models and are specific to individual participants.	
Requests Completed	Number of requests completed by the user
% Committed	Percentage of the completed requests that were committed
Rejected	Number of requests that were rejected
Committed	Number of requests that were committed

Table A-4 (Cont.) Metrics for Model Analysis Dashboard

Metric	Description
Request Items	Cumulative number of request items for all requests completed by the user