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

User's Guide
Release 11.1.2.3
Updated: June 2015
# Contents

**Documentation Accessibility** ........................................................... 9

**Documentation Feedback** ............................................................. 11

## Chapter 1. Getting Started ........................................................... 13

  - About Data Relationship Management ........................................... 13
  - Accessing Data Relationship Management ....................................... 13
  - Changing Passwords ....................................................................... 14
  - Security ......................................................................................... 14
  - Navigation Menu ........................................................................... 15
  - Naming Metadata Objects .......................................................... 16

## Chapter 2. Managing Versions ......................................................... 17

  - Data Relationship Management Version Life Cycle ............................ 17
  - Version Type and Status .................................................................. 19
  - Creating Versions .......................................................................... 19
  - Working with Version Variables ..................................................... 19
    - Using Version Variables .............................................................. 20
    - Creating Version Variables ......................................................... 20
    - Assigning a Version to a Variable ................................................ 20
    - Deleting a Version Variable ......................................................... 21
  - Copying Versions ........................................................................... 21
  - Saving Versions ............................................................................. 22
  - Creating As-Of Versions .................................................................. 22
  - Editing Version Properties ............................................................ 22
  - Assigning Validations to Versions .................................................... 23
  - Assigning Nodes to Domains .......................................................... 24
  - Assigning Domains to Versions ....................................................... 25
  - Unloading and Deleting Versions ..................................................... 25
  - Version Backup and Restore .......................................................... 26

## Chapter 3. Managing Hierarchies ....................................................... 27

  - Creating Hierarchies ....................................................................... 27
Chapter 4. Working with Nodes ........................................................ 35
  About Nodes ............................................................................. 35
  Using Domains ........................................................................ 36
  Adding and Inserting Nodes ..................................................... 36
    ID Function ............................................................................ 39
  Modeling a Node ....................................................................... 39
  Copying Node Properties .......................................................... 41
  Moving and Ordering Nodes ..................................................... 41
    Relaxed Move Option .............................................................. 42
    Sorting Nodes ....................................................................... 42
  Node Deletion Methods ............................................................ 43
    Merging Nodes ....................................................................... 44
    Deleting Nodes ..................................................................... 44
    Removing Nodes ................................................................... 44
    Annulling Nodes .................................................................... 45
    Inactivating Nodes ................................................................. 45
    Reactivating Nodes ................................................................. 46
  Assigning Validations ............................................................... 46
  Orphan Nodes .......................................................................... 47
    Viewing Orphan Nodes .......................................................... 47
    Deleting Orphan Nodes .......................................................... 47
  Navigating in Tree View ........................................................... 48
  Finding Nodes .......................................................................... 48
  Node Security .......................................................................... 50
  Using Shared Nodes ................................................................... 50
    Shared Nodes and Domains ...................................................... 51
    Shared Node Properties .......................................................... 51
    Enabling Shared Nodes .......................................................... 52
Claiming a Request .................................................  72
Approving a Request ................................................  72
Enriching a Request ................................................  72
Pushing Back a Request ..........................................  73
Escalating a Request ..............................................  73
Committing a Request ..............................................  74
Rejecting a Request .................................................  74
Navigating the Worklist .............................................  74
  Worklist Views ....................................................  75
  Viewing Requests by Age .........................................  75
Searching for Requests ............................................  75
Previewing a Request .............................................  76
Opening a Request ...............................................  76
Deleting a Request ...............................................  76
Process Control and Automation ................................  77
  Workflow Path ....................................................  77
  Workflow Tags ....................................................  77
Request and Claim Duration .....................................  78
  Skipping Stages and Automatic Escalation ....................  78
  Reapproval ......................................................  79
Alerts and Notifications ...........................................  79
  Web Client Alerts ................................................  79
  Email Notifications .............................................  80

Chapter 8. Using Queries ...........................................  81
  Query Types .......................................................  81
  Creating Queries ..................................................  82
    Operator Descriptions ......................................  84
  Working with Query Results ..................................  85
    Navigating in List View ....................................  85
    Navigating in Tree View ....................................  86
    Downloading Query Results ................................  86
  Managing Queries ..............................................  86
    Opening Queries .............................................  87
    Copying Queries .............................................  87
    Moving Queries .............................................  87
    Deleting Queries ...........................................  88

Chapter 9. Comparing Hierarchies ................................  89
  Compare Types ...................................................  89
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.
Send feedback on this documentation to: epmdoc_ww@oracle.com

Follow EPM Information Development on these social media sites:
LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051
Twitter - http://twitter.com/hyperionepminfo
Facebook - http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642
Google+ - https://plus.google.com/106915048672979407731/#106915048672979407731/posts
YouTube - http://www.youtube.com/user/OracleEPMWebcasts
Getting Started

In This Chapter

About Data Relationship Management ................................................................. 13
Accessing Data Relationship Management............................................................. 13
Security ........................................................................................................ 14
Navigation Menu ............................................................................................ 15
Naming Metadata Objects .................................................................................. 16

About Data Relationship Management

Oracle 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

Accessing Data Relationship Management

To start the 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.
Select an application and click Log On.

## Changing Passwords

To change a password:

1. From the 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 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**

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.

**Naming Metadata Objects**

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

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. You can save versions and load them to other Data Relationship Management applications.

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 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.

For more information, see:
- Using Version Variables
- Creating Version Variables
- Assigning a Version to a Variable
- Deleting a Version Variable

**Using Version Variables**

In all areas of 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 ![Add](image).  
2. Click ![Add](image).  
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 to a variable.
Select Set Variables, and then 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 a Version Variable

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.

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

➢ To create an As-Of version:

1 From Browse, 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 1  Version Properties and Definitions

<table>
<thead>
<tr>
<th>Version Property</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version ID</td>
<td>Unique, internal identifier for the version.</td>
</tr>
<tr>
<td>Version Name</td>
<td>Name of the version.</td>
</tr>
<tr>
<td>Version Description</td>
<td>Description for the version.</td>
</tr>
<tr>
<td>Version Owner</td>
<td>The user name of the version owner.</td>
</tr>
<tr>
<td>Legacy Name</td>
<td>Previous name of the version.</td>
</tr>
<tr>
<td>Domains in Use</td>
<td>Domains assigned to the version.</td>
</tr>
</tbody>
</table>

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.

Unloading and Deleting Versions

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.

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 Data Relationship Management application may be 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 encountered during the restore process.

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.
A 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 2  Hierarchy Properties**

<table>
<thead>
<tr>
<th>Hierarchy Property</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Display Properties</td>
<td>Default for displaying nodes within the hierarchy</td>
</tr>
<tr>
<td>Default Paste Properties</td>
<td>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.</td>
</tr>
<tr>
<td>Default Match By</td>
<td>Default property by which to synchronize nodes for the current hierarchy.</td>
</tr>
<tr>
<td>Enable Shared Nodes</td>
<td>Determines if shared nodes are allowed in the hierarchy.</td>
</tr>
<tr>
<td>Hierarchy Node Type</td>
<td>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.</td>
</tr>
<tr>
<td>Hierarchy URL</td>
<td>URL hyperlink for directly accessing the hierarchy from an external resource</td>
</tr>
<tr>
<td>Hierarchy Description</td>
<td>Hierarchy description</td>
</tr>
<tr>
<td>Hierarchy Group</td>
<td>Displays the hierarchy group to which the hierarchy is assigned</td>
</tr>
<tr>
<td>Hierarchy ID</td>
<td>Unique, internal identifier for the hierarchy</td>
</tr>
<tr>
<td>Hierarchy Name</td>
<td>Hierarchy name</td>
</tr>
<tr>
<td>Hierarchy Owner</td>
<td>The user name of the hierarchy owner</td>
</tr>
<tr>
<td>Hierarchy Sort Order</td>
<td>Integer value that determines the order of display for hierarchies within a version or hierarchy group</td>
</tr>
<tr>
<td>ID Length Leaf Property</td>
<td>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.</td>
</tr>
<tr>
<td>ID Length Limb Property</td>
<td>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.</td>
</tr>
<tr>
<td>Prefill Leaf Property</td>
<td>Points to the node-level property that is used to pre-fill node name when creating new leaf node</td>
</tr>
<tr>
<td>Hierarchy Property</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Prefill Limb Property</td>
<td>Points to the node-level property that is used to pre-fill node name when creating new limb node</td>
</tr>
<tr>
<td>Standard Hierarchy Sort</td>
<td>Points to a node-level property to determine default sort order within the hierarchy</td>
</tr>
<tr>
<td>Top Node</td>
<td>Name of the top node for the hierarchy</td>
</tr>
<tr>
<td>Top Node ID</td>
<td>Node ID of the top node</td>
</tr>
<tr>
<td>User Display Properties</td>
<td>Determines how hierarchy nodes display for the current user</td>
</tr>
<tr>
<td>User Hierarchy Sort</td>
<td>Points to a node-level property to determine sort order within the hierarchy for the current user</td>
</tr>
<tr>
<td>User Match By</td>
<td>Property by which to synchronize nodes in the selected hierarchy for the current user</td>
</tr>
<tr>
<td>User Paste Properties</td>
<td>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. <strong>Note:</strong> If this property and Default Paste Properties are both set, the user property takes precedence over the default property.</td>
</tr>
</tbody>
</table>

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.**

For more information, see **Enabling and Disabling Node Types**.

**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.

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

Perform an action:
- Expand the top node to view its children.
- Expand limb nodes until you reach the end of the branch.
- View a branch of nodes from the hierarchy. Select a node and choose a command from Options:
  - View From, then Current Node to view the hierarchy from the selected node
  - View From, then Top Node to view the hierarchy from the top node

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.

For more information, see:

- Creating Hierarchy Groups
- Assigning Hierarchies to Hierarchy Groups
- Assigning Controlled Properties

### 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.
About Nodes

Master or reference data records used to describe, qualify, or summarize transactional data in a subscribing system are managed in 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>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf</td>
<td>A node that cannot have children</td>
</tr>
<tr>
<td>Limb</td>
<td>A node that can have children</td>
</tr>
</tbody>
</table>
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
- Insert an existing node not currently in the hierarchy
Caution! Using commas in node names can cause difficulties when working with certain properties that are comma-delimited, such as Ancestor List, Child List and client-created node lists. In addition, using the characters defined in the system preferences for the shared node separator and the hierarchy separator can cause unpredictable behavior and should be avoided. If the query wildcard characters asterisk ( * ) and underscore ( _ ) are used in a node name they cannot be easily searched for unless a derived formula is used to search for them using the Pos function.

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.
Select whether to add the node as a child or sibling of the selected node.

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:
        o 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.
        o 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 a Node

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.
Copy```ing 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 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 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 4  Node Deletion Methods

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destroy</td>
<td>Deletes a node from the version and overrides any required merge logic if enabled</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The Destroy option is only available from the Orphans list.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the node is a domain node whose domain prevents deletion, you cannot delete the node.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the node from all parents in all hierarchies.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the node is a domain node whose domain prevents deletion, you cannot delete the node.</td>
</tr>
<tr>
<td>Delete &amp; Merge</td>
<td>Deletes the node from all parents in all hierarchies and allows you to select a node to use in place of the deleted node.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option is only available if the UseMerge system preference is enabled.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the node is a domain node whose domain prevents deletion, you cannot delete the node.</td>
</tr>
<tr>
<td>Annul</td>
<td>Removes the node from all hierarchies where it has the same parent, and also deletes the node if it is then an orphan</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• If the node does not have the same parent in all hierarchies, the Annul produces the same result as Remove.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the node is a domain node whose domain prevents deletion, you cannot delete the node.</td>
</tr>
<tr>
<td>Annul All Below</td>
<td>Removes all the children of the selected node. If any of the children become orphans, they are also deleted.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the node is a domain node whose domain prevents deletion, you cannot delete the node.</td>
</tr>
<tr>
<td>Remove</td>
<td>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.</td>
</tr>
<tr>
<td>Inactivate</td>
<td>Flags the node as inactive so that it can be filtered from an export.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Inactivate &amp; Merge</td>
<td>Flags the node as inactive and allows you to select a node to use in place of the inactivated node.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option is only available if the UseMerge system preference is enabled.</td>
</tr>
</tbody>
</table>

For more information, see:

- Merging Nodes
- Deleting Nodes
- Removing Nodes
- Annulling Nodes
- Inactivating Nodes
- Reactivating Nodes
Merging Nodes

Subscribing systems supported by 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

1. 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.

1. 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.

### Annulling 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.

For more information, see:

- Viewing Orphan Nodes
- Deleting Orphan Nodes

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 \( \text{ } \) 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 in Tree View

Use the following toolbar buttons to navigate in tree view.

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>←</td>
<td>Navigates to the previously selected node</td>
</tr>
<tr>
<td>↑</td>
<td>Navigates to the parent of the selected node.</td>
</tr>
<tr>
<td>EXP</td>
<td>Expands the tree to display all nodes of the specified level</td>
</tr>
</tbody>
</table>

Finding Nodes

You can search for nodes within any hierarchy in a selected version. Wildcard and exact match searches can be performed. 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 Find By 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.
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:

   - **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

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>
<thead>
<tr>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Node</td>
<td>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.</td>
</tr>
<tr>
<td>Explicit Shared Node</td>
<td>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.</td>
</tr>
<tr>
<td>Implicit Shared Node</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 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

For more information, see:

- Shared Nodes and Domains
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.

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

### 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.

1. 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**

Primary and shared nodes can be located in different branches of a hierarchy and it may be desirable for the primary node to be located in a specific branch. If an explicit shared node is created in the place where the primary node should be, you can convert the shared node to the primary node to switch their locations in the hierarchy 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 🌑.

**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.
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. Data Relationship Management includes default property categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>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. <strong>Note:</strong> Properties cannot be assigned to this category.</td>
</tr>
<tr>
<td>Stats</td>
<td>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.</td>
</tr>
<tr>
<td>Validation</td>
<td>Validations assigned for the node; one property for each validation. This category is available for nodes only.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Property</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of node</td>
</tr>
<tr>
<td>Display String</td>
<td>Value of combined properties used for viewing nodes in a hierarchy. Controlled by the Default Display By and User Display By properties.</td>
</tr>
<tr>
<td>Domain Name</td>
<td>Name of the domain to which the node is a member</td>
</tr>
<tr>
<td>Domain Node Inactive Date</td>
<td>Specified date for which the domain node is declared inactive</td>
</tr>
<tr>
<td>Domain Node Name</td>
<td>Name of the node within its domain. This value does not include the domain qualifier</td>
</tr>
<tr>
<td>Domain Node Description</td>
<td>Most recent description defined for a domain node in any version where it exists</td>
</tr>
<tr>
<td>Leaf</td>
<td>• If True, node is a leaf node and cannot have children</td>
</tr>
<tr>
<td></td>
<td>• If False, node is a limb node and may have children</td>
</tr>
<tr>
<td>Leaf Access</td>
<td>Indicates level of access to node for current user (only applies if node is leaf)</td>
</tr>
<tr>
<td>Limb Access</td>
<td>Indicates level of access to node for current user (only applies if node is limb)</td>
</tr>
<tr>
<td>Name</td>
<td>Unique name of node within a version</td>
</tr>
<tr>
<td>Node Approved</td>
<td>Boolean flag to indicate that node has been in a Finalized version</td>
</tr>
<tr>
<td>Node ID</td>
<td>Unique numeric identifier for the node generated by the system</td>
</tr>
<tr>
<td>Node URL</td>
<td>URL hyperlink for directly accessing the node from an external resource</td>
</tr>
</tbody>
</table>

**Stats Properties**

Statistical properties displayed at the node level are listed in the following table.
### Table 7  Stats Properties

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

#### 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 8  Property Scope Icons

<table>
<thead>
<tr>
<th>Property Scope</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>The property value is constant across all hierarchies within a version.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td>The property value is specific to the hierarchy in which it exists.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td></td>
</tr>
<tr>
<td>Global Inherited</td>
<td>The value of the property is global and inherited from an ancestor node in</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>the controlling hierarchy for the property.</td>
</tr>
<tr>
<td>Local Inherited</td>
<td>The value of the property is local and inherited from an ancestor node in</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>the current hierarchy.</td>
</tr>
</tbody>
</table>

### Table 9  Property Origin Values

<table>
<thead>
<tr>
<th>Property Origin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>The value of the property is populated by a default value (as set by the</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>application administrator.) Default values are used when:</td>
</tr>
<tr>
<td></td>
<td>● The property is non-inheriting and you have not entered a value.</td>
</tr>
<tr>
<td></td>
<td>● The property is inheriting and no ancestor value is found.</td>
</tr>
<tr>
<td>Inherited</td>
<td>The value of the property is being supplied by an ancestor node. (The name</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>of the ancestor node and the hierarchy from which the inherited value</td>
</tr>
<tr>
<td></td>
<td>originates is displayed in the Status column.)</td>
</tr>
<tr>
<td>Overridden</td>
<td>The value of the property was set by a user at this specific node.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td></td>
</tr>
<tr>
<td>Derived</td>
<td>The value of the property is derived (calculated) from other property</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>values.</td>
</tr>
<tr>
<td>Locked</td>
<td>The value of the property has been locked for the current node and cannot</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>be changed in any descendant node that inherits this property's value.</td>
</tr>
<tr>
<td></td>
<td>If this property value is viewed in a descendant node, its status is</td>
</tr>
<tr>
<td></td>
<td>indicated with the Static icon.</td>
</tr>
<tr>
<td>Static</td>
<td>The value of the property has been inherited from an ancestor node, whose</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>value has been locked:</td>
</tr>
<tr>
<td></td>
<td>● As a result, the value cannot be changed in the descendant node</td>
</tr>
<tr>
<td></td>
<td>● The name of the ancestor node and the hierarchy from which the inherited</td>
</tr>
<tr>
<td></td>
<td>value originates is shown in the Status column of the Property window</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Property status</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear All Below</td>
<td>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.</td>
</tr>
<tr>
<td>Locked</td>
<td>The classification of the property has been changed to Locked.</td>
</tr>
<tr>
<td>Modified Value</td>
<td>The value has been modified.</td>
</tr>
<tr>
<td>Remove Duplicates</td>
<td>A request has been made to Remove Duplicates for the selected property. Removes any equivalent value in the node and its children.</td>
</tr>
<tr>
<td>Remove Value</td>
<td>The value has been removed; only applicable to Overridden property values.</td>
</tr>
<tr>
<td>Unlocked</td>
<td>The classification of the property has been changed to Unlocked and the value can be changed in descendant nodes that inherits the property.</td>
</tr>
</tbody>
</table>

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 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 in the node and its children. For an inheriting property, removes any value
equal to the inherited value in the node and its children. For a read/write derived property, removes any value equal to the derived value in the node and its children.

- **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.

**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.

To filter properties by label or value:

1. In the property grid, 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 **Show Filter Row**.
2. Select **Show All Properties**.
Validating Data

In This Chapter

Validating Versions ......................................................................................... 61
Validating Hierarchies ..................................................................................... 62
Validating Nodes ........................................................................................... 62
Working With Validation Results ...................................................................... 63

Versions, hierarchies, nodes and their properties are checked for data integrity using 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

You can work with validation results in the following ways:

- Navigating Version and Hierarchy results in List View
- Navigating Node Results in List View
- Navigating Node Results in Tree View
- Downloading Node Results
- Clearing 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>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Parent" /></td>
<td>Navigates to the parent of the selected node.</td>
</tr>
<tr>
<td><img src="image" alt="Previous" /></td>
<td>Navigates to the previously selected marked node.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td>Navigates to the next marked node.</td>
</tr>
<tr>
<td></td>
<td>Expands the tree to display the marked nodes.</td>
</tr>
<tr>
<td></td>
<td>Clears the marked nodes.</td>
</tr>
</tbody>
</table>

**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.

1. 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.

1. 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.

1. 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**.
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 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 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

For more information, see:
- Workflow Tasks
- Workflow Stages

**Workflow Tasks**

A workflow task represents a change action to be performed along with a set of properties to be displayed, edited, and required during the task. Workflow tasks are defined outside of 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 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.

For more information, see:
- Change Requests
- Remediation Requests
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

A workflow task defines a set of changes to be performed for a hierarchy node in a Data Relationship Management version. Each stage of a workflow model identifies the tasks applicable to that stage for requests using the model. Workflow tasks are performed during requests by adding or editing request items. A request may consist of one or more request items which may be added during initial submission or later enrichment of the request. During Submit or Enrich workflow stages, new request items can be added to a request to perform additional tasks. During Enrich stages, request items added in previous stages can be edited using an update task assigned to the workflow stage. Request items can only be deleted during workflow stages which have a matching workflow task to the request item. The order of request items in a request is important because it controls the order in the changes in the request get applied to the target version for validation and commitment purposes.

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.

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.

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>
<thead>
<tr>
<th>Action Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>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.</td>
</tr>
<tr>
<td>Submit</td>
<td>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.</td>
</tr>
<tr>
<td>Calculate</td>
<td>Calculates request item details based on proposed changes in the request.</td>
</tr>
</tbody>
</table>
| 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:  
  - The request is in the Draft workflow status and has at least one request item  
  - The request is in the Claimed workflow status |
<table>
<thead>
<tr>
<th>Action Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>Copies request items to a new request. To copy a request, you must have access to submit the type of request you have selected.</td>
</tr>
<tr>
<td>Claim</td>
<td>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.</td>
</tr>
<tr>
<td>Unclaim</td>
<td>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.</td>
</tr>
<tr>
<td>Pushback</td>
<td>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.</td>
</tr>
<tr>
<td>Escalate</td>
<td>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.</td>
</tr>
<tr>
<td>De-escalate</td>
<td>Reassigns the request to the next node access group for the current workflow stage.</td>
</tr>
<tr>
<td>Approve</td>
<td>Approves and commits the changes in the request to the target version.</td>
</tr>
<tr>
<td>Reject</td>
<td>Rejects the request. A submitted request may be rejected by any approving user in the workflow path. Rejecting a request assigns the request back to the submitter and terminates the workflow for the request. A rejected request cannot be resubmitted. To resubmit the request items in the rejected request, the user can copy the rejected request to a new draft request, edit the request items in the new request, and then submit the new request.</td>
</tr>
</tbody>
</table>

**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. The actions that can be performed by each participating user will vary depending on the stage at which they are participating and their workflow access to the data referred to by the request items.

For more information, see:

- Creating a Request
- Submitting a Request
- Claiming a Request
- Approving a Request
- Enriching a Request
- Pushing Back a Request
- Escalating a Request
- Committing a Request
- Rejecting a Request

**Creating a Request**

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 **Add Items**.
4. 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.

5 Click OK.

6 In the **Item Details** section of the page, define new property values for the request item.

**Note:** Required values are marked with an asterisk.

7 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.

8 Repeat steps 3-7 to add items to the request.

9 Click , enter a title for the request, and then click OK.

### Submitting a Request

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.

- To submit a request:
  1 Create a request or open a drafted request.
  2 Click .
Claiming a Request

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. Unclaiming a request makes the request available for any other assigned user to claim for the active workflow stage.

To claim a request:
1. Open a submitted request.
2. Click 📖.

Approving a Request

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. Open a claimed request.
2. Perform the tasks assigned.
3. Click 📖.

Enriching a Request

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. Open a claimed request.
2. Update request items.
3. Click 📋.

Pushing Back a Request

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. 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.

Escalating a Request

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 to involve them in resolving the situation. When escalated, the request is assigned to data manager users and normal approvers for the stage cannot claim the request to participate. While escalated, data managers 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 the
data manager permits 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. **Open a claimed request.**
2. Click **.**
3. **Enter a comment to explain the escalation and click OK.**

## Committing a Request

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.

## Rejecting a Request

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. **Open a claimed request.**
2. Click **.**
3. **Enter a comment to explain the rejection and click OK.**

## 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.

For more information, see:
• Worklist Views
• Viewing Requests by Age
• Searching for Requests
• Previewing a Request
• Opening a Request
• Deleting a Request

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 the Search In button, 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 the search button.

Previewing a Request

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 a Request

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 the open button.

Deleting a Request

A draft request that is never submitted may be deleted by the creator while previewing the request on the Worklist page. After a request is submitted, the ability to delete the request is restricted in order to preserve an accurate history of the request for governance purposes. Submitted requests can only be deleted along with the target version to which they are associated.

To delete a draft request:

1. From the Worklist, select a drafted request to preview.
2. In the request preview, click the delete button.
3. Click Delete the Request to confirm the deletion.
Process Control and Automation

Governance workflows include several features:

- Workflow path
- Workflow tags
- Request and Claim Duration
- Skipping Stages and Automatic Escalation
- Reapproval

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

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 \(\times\) next to the workflow tags to remove from the request.

**Request and Claim Duration**

The workflow model for a request may be optionally configured with a request or claim duration interval to control automatic handling of the request by a governance workflow based on an estimated amount of time expected for a particular type of request.

- **Request Duration**—The expected number of days that a request should take to be approved and committed. After the age of a request exceeds the request duration, the request is marked as overdue.
- **Claim Duration**—The expected number of days that a request should be claimed for a workflow stage by a governance user. After the age of a request exceeds the claim duration, the request is automatically unclaimed to make it available for other assigned users to claim.

**Skipping Stages and Automatic Escalation**

When a request is assigned to the next stage in the workflow path, 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. In these situations, governance workflows may permit a workflow stage to be skipped if all required values are already provided and all validations for the stage pass successfully. The ability to skip workflow stages enables the workflow model to adapt the workflow path to only stages necessary to be completed based on the actual items in a request. This adaptation capability enables broader workflow models to handle specific workflow requirements for certain data sets differently using the same model rather than requiring a separate workflow model and separate requests to be used for that purpose. Any approve, enrich, or commit workflow stage can be configured to permit or restrict the stage from being skipped.
A workflow stage cannot be skipped if any request items in a request do not have required values or fail validations configured for the stage, or if the workflow model has been configured to restrict this behavior. In this case, the request is escalated to Data Manager role users.

**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 Re-Approve option for each workflow stage in a workflow model controls this behavior for requests using that model.

**Alerts and Notifications**

Governance users and data managers are made aware of request activities for requests to which they are associated using alerts and notifications. 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.

For more information, see:

- Web Client Alerts
- Email Notifications

**Web Client Alerts**

When a user needs to be notified of request activity, the user receives an alert in the 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.
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.
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.

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. 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.
4. 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.
5. 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.
6. 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.
7. On the Filter, click Add to insert a criteria row.
8. Select a property and operator, enter a value for the criteria row, and press Enter to save the row.
Note: You can include parenthesis to group criteria and to define whether to include inherited values for the property. For operator descriptions see “Operator Descriptions” on page 84.

Tip: Click ✍️ to modify a row. Click ✗ to delete a row. Click the arrows to move a row.

9 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

10 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.

11 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).

12 Select the Columns tab and choose a Category.

13 Select the properties to display as columns in the query results and move them from Available to Selected.

   Note: 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.

14 Optional: Select Include Criteria Properties to automatically include properties referenced in the filter criteria in the query results.

15 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.
## Operator Descriptions

Table 11  Operator Descriptions

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Data Type Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Above</td>
<td>The specified value of the property is used by an ancestor node.</td>
<td></td>
</tr>
<tr>
<td>Below</td>
<td>The specified value of the property is used by a descendant node.</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>The comma-delimited list value of the property contains the specified item in the list.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Equal</td>
<td>The value of the property is equal to the specified value.</td>
<td></td>
</tr>
<tr>
<td>Greater Than</td>
<td>The value of the property is greater than (but not equal to) the specified value.</td>
<td></td>
</tr>
<tr>
<td>Greater Than, Equal</td>
<td>The value of the property is greater than or equal to the specified value.</td>
<td></td>
</tr>
<tr>
<td>Has Characters</td>
<td>The value of the property has one or more of the specified characters.</td>
<td></td>
</tr>
<tr>
<td>Has Not Characters</td>
<td>The value of the property does not have any of the specified characters.</td>
<td></td>
</tr>
<tr>
<td>In</td>
<td>The value of the property is in the specified comma-delimited list of values.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Is Alpha</td>
<td>The value of the property consists of alphabetic characters only.</td>
<td></td>
</tr>
<tr>
<td>Is Alpha Numeric</td>
<td>The value of the property consists of alphabetic or numeric characters only.</td>
<td></td>
</tr>
<tr>
<td>Is Blank</td>
<td>The value of the property is blank (the specified value field is unavailable).</td>
<td></td>
</tr>
<tr>
<td>Is Defined</td>
<td>The value of the property is a defined (overridden) value.</td>
<td></td>
</tr>
<tr>
<td>Is Not Alpha</td>
<td>The value of the property does not consist of alphabetic characters only.</td>
<td></td>
</tr>
<tr>
<td>Is Not Alpha Numeric</td>
<td>The value of the property does not consist of alphabetic or numeric characters only.</td>
<td></td>
</tr>
<tr>
<td>In Not Blank</td>
<td>The value of the property is not blank.</td>
<td></td>
</tr>
<tr>
<td>Is Not Numeric</td>
<td>The value of the property is not numeric.</td>
<td></td>
</tr>
<tr>
<td>Is Numeric</td>
<td>The value of the property is numeric.</td>
<td></td>
</tr>
<tr>
<td>Length Equal</td>
<td>The length of the property value is equal to the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Length Greater Than</td>
<td>The length of the property value is greater than (but not equal to) the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Length Greater Than, Equal</td>
<td>The length of the property value is greater than or equal to the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Length Less Than</td>
<td>The length of the property value is less than (but not equal to) the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Length Less Than, Equal</td>
<td>The length of the property value is less than or equal to the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Length Not Equal</td>
<td>The length of the property value is not equal to the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Less Than</td>
<td>The value of the property is less than (but not equal to) the specified value.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td>Data Type Restrictions</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Less Than, Equal</td>
<td>The value of the property is less than or equal to the specified value.</td>
<td></td>
</tr>
<tr>
<td>Like</td>
<td>The value of the property is like the specified value (including wildcards). The asterisk (*) and percent sign (%) can be used as wildcards.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Not Above</td>
<td>The specified value of the property is not used by an ancestor node.</td>
<td></td>
</tr>
<tr>
<td>Not Below</td>
<td>The specified value of the property is not used by a descendant node.</td>
<td></td>
</tr>
<tr>
<td>Not Contains</td>
<td>The comma-delimited list value of the property does not contain the specified item in the list.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Not Equal</td>
<td>The value of the property is not equal to the specified value.</td>
<td></td>
</tr>
<tr>
<td>Not In</td>
<td>The value of the property is not in the specified comma-delimited list of values.</td>
<td>Strings only</td>
</tr>
<tr>
<td>Not Like</td>
<td>The asterisk (*) and percent sign (%) can be used as wildcards.</td>
<td>Strings only</td>
</tr>
</tbody>
</table>

**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.

For more information, see:

- Navigating in List View
- Navigating in Tree View
- Downloading Query Results

**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>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Parent Icon]</td>
<td>Navigates to the parent of the selected node.</td>
</tr>
<tr>
<td>![Previous Icon]</td>
<td>Navigates to the previously selected marked node.</td>
</tr>
<tr>
<td>![Expand Icon]</td>
<td>Expands the tree to display all nodes of the specified level.</td>
</tr>
<tr>
<td>![Next Marked Icon]</td>
<td>Navigates to the next marked node.</td>
</tr>
<tr>
<td>![Expand Marked Icon]</td>
<td>Expands the tree to display the marked nodes.</td>
</tr>
<tr>
<td>![Clear Marked Icon]</td>
<td>Clears the marked nodes.</td>
</tr>
</tbody>
</table>

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.

1. Go to 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. For more information see:

- Opening Queries
- Copying Queries
Moving Queries

Deleting 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.
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**

Data Relationship Management offers four types of compares:

- **Visual Compare**
- **Structure Compare**
- **Property Compare**
- **Rename Compare**

**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. **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.
4. **For each version, click ⬆️, and from the Select Node dialog box, select a hierarchy and a top node, and click OK.**
5. **Select the Style tab and choose the compare type:**
   - Visual
   - Structure
   - Property
   - Rename
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.

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 for the criteria row, and press Enter to save the row.

  **Note:** You can include parentheses to group criteria and to define whether to include inherited values for the property. For operator descriptions see “Operator Descriptions” on page 84.

  **Tip:** Click 🆙 to modify a row. Click ✗ to delete a row. Click the arrows to move a row.

From Limb/Leaf Filter, select nodes to include.

Select the Columns tab and choose a Category.

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.
Optional: Select Include Compare/Filter Properties to automatically include properties referenced in the filter criteria for Property compares.

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.

For more information, see:

- Navigating in List View
- Navigating in Tree View
- Downloading Query Results

**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>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>navigate to parent</td>
<td>Navigates to the parent of the selected node.</td>
</tr>
<tr>
<td>navigate to previous marked node</td>
<td>Navigates to the previously selected marked node.</td>
</tr>
<tr>
<td>expand tree to level</td>
<td>Expands the tree to display all nodes of the specified level.</td>
</tr>
<tr>
<td>navigate to next marked node</td>
<td>Navigates to the next marked node.</td>
</tr>
<tr>
<td>expand marked nodes</td>
<td>Expands the tree to display the marked nodes.</td>
</tr>
<tr>
<td>clear marked nodes</td>
<td>Clears the marked nodes.</td>
</tr>
</tbody>
</table>

### Moving Nodes Between Versions

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

1. **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. For more information see:

- Opening Compares
- Copying Compares
- Moving Compares
- Deleting 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.
Working with Action Scripts

In This Chapter

- Action Script Parameters ................................................................. 97
- Loading Action Scripts From Flat Files ............................................. 101
- Loading Action Scripts From a Transaction Log .............................. 103
- Loading Action Scripts From a Node Model ..................................... 104

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.

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>
<thead>
<tr>
<th>Table 12  Action Script Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action Name</strong></td>
</tr>
<tr>
<td>Activate</td>
</tr>
</tbody>
</table>

Activates the specified node in the hierarchy
<table>
<thead>
<tr>
<th>Action Name</th>
<th>Param1</th>
<th>Param2</th>
<th>Param3</th>
<th>Param4</th>
<th>Param5</th>
<th>Param6</th>
<th>Param7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Parent node</td>
<td>Leaf property</td>
<td>Description</td>
<td>Domain</td>
</tr>
<tr>
<td>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)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AddDomainToVersion</strong></td>
<td>Version</td>
<td>Domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adds a domain to a version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AddHier</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Description</td>
<td>Top Node</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adds a hierarchy to a version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AddInsert</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Parent node</td>
<td>Leaf property</td>
<td>Description</td>
<td>Domain</td>
</tr>
<tr>
<td>Adds the specified node (or domain node) if it does not already exist in the specified version, otherwise the node is inserted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AddOrphan</strong></td>
<td>Version</td>
<td>Node</td>
<td>Leaf Property</td>
<td>Description</td>
<td>Required if adding a domain node</td>
<td>Domain</td>
<td></td>
</tr>
<tr>
<td>Adds an orphan node (or domain node) to a version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Annul</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annuls the specified node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AnnulAllBelow</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annuls all nodes following the specified node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AssignHierValidations</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assigns validations at the hierarchy level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AssignNodeToDomain</strong></td>
<td>Version</td>
<td>Node</td>
<td>Domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assigns node to domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AssignVersionValidations</strong></td>
<td>Version</td>
<td>Validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assigns validations at the version level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ChangeHierTopNode</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>New Top Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changes the top node for a specified hierarchy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Name</td>
<td>Param1</td>
<td>Param2</td>
<td>Param3</td>
<td>Param4</td>
<td>Param5</td>
<td>Param6</td>
<td>Param7</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>ChangeProp</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Property label or name Controlled by the Property References By Label option.</td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ClearPropsBelow</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Property label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>(Merge node)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHier</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteOrphan</strong></td>
<td>Version</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Destroy</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Inactivate</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>(Merge node)</td>
<td>(Domain Node Inactive Date)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Parent node</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LockProp</strong></td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Property label</td>
<td>Is Locked (0=false, 1=true)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Name</td>
<td>Param1</td>
<td>Param2</td>
<td>Param3</td>
<td>Param4</td>
<td>Param5</td>
<td>Param6</td>
<td>Param7</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Move</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Destination parent node</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move</td>
<td>Version</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NodeExists</td>
<td>Version</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NodeExistsInHier</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NodeHasAsParent</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Parent node</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rem</td>
<td>Remark</td>
<td>Remark</td>
<td>Remark</td>
<td>Remark</td>
<td>Remark</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoveDomainFromVersion</td>
<td>Version</td>
<td>Domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoveDupProps</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Property label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Name</td>
<td>Param1</td>
<td>Param2</td>
<td>Param3</td>
<td>Param4</td>
<td>Param5</td>
<td>Param6</td>
<td>Param7</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>RemoveHierPropValue</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Property</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Removes (clears) the value of a hierarchy property</td>
<td></td>
<td></td>
<td></td>
<td>label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoveProp</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Node</td>
<td>Property</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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)</td>
<td></td>
<td></td>
<td></td>
<td>label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoveVersionPropValue</td>
<td>Version</td>
<td>Property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Removes (clears) the value of a version property</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetControlledProperties</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Property</td>
<td>List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sets the controlling hierarchy for one or more properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateHierPropValue</td>
<td>Version</td>
<td>Hierarchy</td>
<td>Property</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updates the value of a hierarchy property</td>
<td></td>
<td></td>
<td></td>
<td>label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateVersionPropValue</td>
<td>Version</td>
<td>Property</td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updates the value of a version property</td>
<td></td>
<td></td>
<td></td>
<td>label</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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:

- Tab-separated Action Script (TSV)
- Comma-separated file (CSV)
- Tab-separated file (TSV)
- Portable Document Format (PDF)
- Rich Text Document (RTF)
- Spreadsheet (XLS)
To download action script results:

1. From Scripts, 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.
Using Data Relationship Management, you can import data extracted from external systems, from relational database tables, 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.
### Table 13  Import Sections

<table>
<thead>
<tr>
<th>Import Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>Property values of the new version&lt;br&gt;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.&lt;br&gt;<em>Required column: Version Name</em></td>
</tr>
<tr>
<td><strong>Hierarchy</strong></td>
<td>Hierarchies to be created by the import process, their top nodes, and property values for each hierarchy&lt;br&gt;This section is required to relate nodes to each other using hierarchy relationships.&lt;br&gt;<em>Required columns: Hierarchy Name, Top Node</em></td>
</tr>
<tr>
<td><strong>Node</strong></td>
<td>Nodes to be created by the import process and global node property values for each node&lt;br_NODES that are not related to hierarchies will be imported into the new version as orphans.&lt;br&gt;<em>Required column: Node Name</em></td>
</tr>
<tr>
<td><strong>Relationship</strong></td>
<td>Parent-child relationships between nodes in hierarchies and global or local node property values for the child node&lt;br&gt;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.&lt;br&gt;<em>Required columns: Parent Node Name, Node Name</em></td>
</tr>
<tr>
<td><strong>Hierarchy Node</strong></td>
<td>Local node property values specific to a specified hierarchy&lt;br&gt;The Hierarchy import section is required when this section is being used.&lt;br&gt;<em>Required columns: Hierarchy Name, Node Name</em></td>
</tr>
</tbody>
</table>

**Import File Example**

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

### 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. 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.
   - 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 Table, 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 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.
     e. Go to step 6.

4. Select File Format options:
   - **Character Encoding** — Select one of the encoding options.
   - **Strip Quoted Strings** — Select 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.

5. Select from the following Section Headings 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 the Style tab and choose Relationship Sorting options:
- **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, 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.

  **Note:** All hierarchies are sorted by default unless specified here.

- **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.

7 In the Domain Options section, 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.

8 Select Duplicate Handling options:

- **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).

9. **On the Columns tab, from the Section list, select an import section.**

10. **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.

11. **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.

12 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

13 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 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.

14 On the Filters tab, make selections to skip blank values and default values for certain property types.

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

15 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.

16 **On the Target tab,** enter a name and, optionally, a description for the new version that will be created by the import.

17 **Select a value for the maximum number of errors that can occur during processing before the import stops.**

18 **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.
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:
   a. User—Personal objects that are only available to an individual user to view and run.
   b. Standard—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard blenders.
   c. 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:
   a. Download as CSV
   b. Download as PDF
   c. Download as RTF
   d. 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.
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

Data Relationship Management provides several export types to output data in different formats. The following table lists the export types for Data Relationship Management.

<table>
<thead>
<tr>
<th>Export Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Exports</td>
<td>Outputs nodes and properties from hierarchies. This export can be used to create a parent-child format or a basic list of hierarchy nodes.</td>
</tr>
<tr>
<td><strong>Hierarchy</strong></td>
<td><strong>Usage Frequency:</strong> High</td>
</tr>
<tr>
<td></td>
<td><strong>Output Formats:</strong> File, Database table</td>
</tr>
<tr>
<td><strong>Hierarchy XML</strong></td>
<td>Outputs nodes, their relationships, and properties from hierarchies in an XML format.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage Frequency:</strong> High</td>
</tr>
<tr>
<td></td>
<td><strong>Output Formats:</strong> File</td>
</tr>
<tr>
<td><strong>Generation</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage Frequency:</strong> Medium</td>
</tr>
<tr>
<td></td>
<td><strong>Output Formats:</strong> File</td>
</tr>
<tr>
<td>Export Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Comparison Exports</strong></td>
<td></td>
</tr>
<tr>
<td>Compare</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Medium</td>
<td><strong>Output Formats</strong>: File, Database table</td>
</tr>
<tr>
<td>Difference</td>
<td>Compares two hierarchies and outputs a record for each difference found. This export can be used for reporting incremental changes to users.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Low</td>
<td><strong>Output Formats</strong>: File</td>
</tr>
<tr>
<td><strong>Integration Exports</strong></td>
<td></td>
</tr>
<tr>
<td>EPM Architect</td>
<td>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 Planning.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Only when integrating with Oracle Hyperion EPM Architect</td>
<td><strong>Output Formats</strong>: File</td>
</tr>
<tr>
<td><strong>Version Exports</strong></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Medium</td>
<td><strong>Output Formats</strong>: File, Database table</td>
</tr>
<tr>
<td><strong>Property Exports</strong></td>
<td></td>
</tr>
<tr>
<td>Property List</td>
<td>Outputs a record for each item in the list of values for a property definition. This is a special use case export type.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Low</td>
<td><strong>Output Formats</strong>: File</td>
</tr>
<tr>
<td>Property Lookup</td>
<td>Outputs a record for each key-value pair in the lookup table for a property definition. This is a special use case export type.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Low</td>
<td><strong>Output Formats</strong>: File</td>
</tr>
<tr>
<td><strong>Log Exports</strong></td>
<td></td>
</tr>
<tr>
<td>Transaction Log</td>
<td>Outputs a record for each item in the transaction log that met the filter criteria. This export can be used for audit purposes.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Medium</td>
<td><strong>Output Formats</strong>: File, Database table</td>
</tr>
<tr>
<td>Merge Log</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Usage Frequency</strong>: Low</td>
<td><strong>Output Formats</strong>: File</td>
</tr>
</tbody>
</table>

**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. **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.

5. **Click Add and select nodes.**
   
   Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.

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
   
   - **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.

7. **In the Batch Validations section, select the type of validations to run.**

   - **None** — No validations run.
   
   - **Assigned** — All validations assigned to the version, hierarchies, and nodes for the export 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.
Selected — Click to select validations to run.

8 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. Setting the Repeat Number Property default value to 1 is recommended.

- **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).

9 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.

10 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.

  **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.

11 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 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).

12 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.

13 **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 strings in the export.
  - **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.

**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 strings
  - **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

14 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 ![validate](image). Validation errors can be copied to the clipboard.
- To run the export, click ![run](image).

If you selected to run validations on the **Style** tab, clicking ![run](image) 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 ![validate](image) 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>
<thead>
<tr>
<th>Customization Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;%Date[(format)]%&gt;</code></td>
<td>Returns current date based on the user's regional settings, where format = MM/DD/YYYY</td>
</tr>
<tr>
<td><code>&lt;%Time[(format)]%&gt;</code></td>
<td>Returns current time based on the user's regional settings, where format = HH:MM:SS AMPM</td>
</tr>
<tr>
<td><code>&lt;%Timestamp%&gt;</code></td>
<td>Returns combination of date and time.</td>
</tr>
<tr>
<td><code>&lt;%CurrentUser%&gt;</code></td>
<td>Returns current Data Relationship Management user.</td>
</tr>
<tr>
<td><code>&lt;%CurrentVersion%&gt;</code></td>
<td>Returns current Data Relationship Management version for this export.</td>
</tr>
<tr>
<td><code>&lt;%PreviousVersion%&gt;</code></td>
<td>Returns previous Data Relationship Management version for this export.</td>
</tr>
<tr>
<td><code>&lt;%ExportName%&gt;</code></td>
<td>Returns name of export.</td>
</tr>
<tr>
<td><code>&lt;%FileName%&gt;</code></td>
<td>Returns output file name if specified</td>
</tr>
<tr>
<td></td>
<td>Returns default file name if not specified</td>
</tr>
<tr>
<td><code>&lt;%RecordCount%&gt;</code></td>
<td>Returns count of records exported (only available in footer)</td>
</tr>
<tr>
<td><code>&lt;%AppVersion%&gt;</code></td>
<td>Returns the Data Relationship Management product number (for example, 11.1.2)</td>
</tr>
<tr>
<td><code>&lt;%PropValue()%&gt;</code></td>
<td>Returned value depends on the specified export type as described following:</td>
</tr>
<tr>
<td>Customization Tag</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>`&lt;%PropValue([Previous</td>
<td>Current(Default)], PropName) %&gt;`</td>
</tr>
</tbody>
</table>
| `<%PropValue([HierName], PropName) %>`                | • 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                        |

**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 ![Export Icon](image).  
3. **Select Hierarchy XML 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.
5. **Click Add and select nodes.**
   Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.
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
   - **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.
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, 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.
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 ...
- 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.

9 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.

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, 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.

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 .
  Validation errors can be copied 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.

**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. 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.
5. Click Add and select nodes.

   Use the arrows to reposition the nodes, which determine the order in which hierarchy nodes are exported.
6. 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.

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. 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. Setting the Repeat Number Property default value to 1 is recommended.

   - **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).

9. 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.

10. 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.

11 **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

12 **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.

13 **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 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.

14 **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.

15 **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 strings
- **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 strings
- **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

16 **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. 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.

5. 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.

6 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 **Select** to select validations to run.

7 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 **Define**.

**Note:** The inclusion filter determines the set of nodes from each hierarchy to be compared. The exclusion filter excludes nodes from export results.

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 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 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**, choose options:
  - **Column Headings** — Includes column headings in the export
  - **Quoted Strings** — Puts quotation marks around strings
  - **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 strings
  - **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 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.

1. To create a difference export:
   1. On the Home page, select Export.
   2. Click ➤.
   3. Select Difference as the export type.
   4. 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.

5. 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.

6. **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.

7. **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.

8. **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 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

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.
   - **Database Table** — Save results to a database table.

10. **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 strings in the export.
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 strings
- 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 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 ![run](image). If you selected Client File, you have a download option where you can download the file locally.

### EPM Architect Export

Performance Management Architect exports output hierarchies, nodes, and properties from 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.

For more information on running these exports from Oracle Hyperion EPM 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 ![run](image).**
3. **Select EPM Architect as the export type.**
4. **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.
5 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.

6 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.

7 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.

8 On the Columns tab, select the Parent Key Property and Member Key Property (the Data Relationship Management property for the node when determining duplicates).

9 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. 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.
  5. 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.
6 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.

7 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.

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 **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.

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 strings
  - **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 strings
- **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.

- **CharacterEncoding** — 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 ![button](image) 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 ![button](image).
     
     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 ![button](image).
     
     You can copy validation errors to the clipboard.

   - To run the export, click ![button](image).
     
     If you selected to run validations on the **Style** tab, clicking ![button](image) performs the validations. If no errors occur, the export runs. If validation errors occur, then the export stops and an error is displayed. Click ![button](image) 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.
   - **Database Table** — Save results to a database table.
7. Perform an action:
   
   - If you selected to export to a **Client File**, select options:
     
     - **Quoted Strings** — Puts quotation marks around strings
     - **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 strings
     - **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. 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 lookup property upon which to base the export.
6. Optional: Select the Style tab, and select Differences Only to filter out lookup pairs where the Lookup Key and Result Value are the same.
7. 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.

8. 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.
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 strings
- **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 strings
  - **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.

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. 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: You can also select the Run Export for All Versions option.
5. Click ➤ to select hierarchies and nodes for the export.
6. 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.

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 Database Table, 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 strings
  - 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 strings
  - **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 ![text editor] 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.

### 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. 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.
5. 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
6. 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.

7 On the Target tab, from the Device list, select the location to which you want to export the results: Client
File or Server File.

8 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 strings
  - **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 \( \text{...} \) 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 strings
  - **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 \( \text{...} \) 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

9 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.

To create an export book:

1. On the Home page, select Export.
2. Click 
3. On the Source tab, select a version or version variable from which to select exports for the book.
4. 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.

5 **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.

6 **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.

7 **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:

       o **User**—Personal objects that are only available to an individual user to view and run.
       o **Standard**—Public objects that are available to all users to view and run. Only Data Manager users can create and modify standard exports.
       o **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.
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.
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.
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 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
- 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:

- Blend
- Book
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.
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
- Auditing Request History

**Auditing Transaction History**

All 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 Filter 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:

- To view the details of a transaction, click the plus-sign to the left of the row. The transaction 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.

### 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

- Add 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
Auditing Requests

Change and remediation requests processed by governance workflows in 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
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

In This Chapter

- Securing the Batch Client ................................................................. 174
- Running the Batch Client ................................................................. 174
- Command Line Syntax ...................................................................... 175
- Configuration File Syntax ................................................................. 182
- Outputs and Result Codes ................................................................. 188

The 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

The batch client utility (drm-batch-client.exe) is installed by default to: C:\Oracle\Middleware\EPMSystem11R1\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, submit a request to perform the given 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 information on error level values, see Table 36 on page 188.

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 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 Application Interface (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.

For more information, see Setting up Batch Client Credentials.

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 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\EPMSystem11R1\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.

For more information, see:

- Dependencies
- Saving Versions in the Batch Client

**Dependencies**

You must have the Microsoft .NET Framework installed on the machine running the Batch Client. Connectivity/visibility to the Process Manager 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 client interface.

**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 Syntax**

The Batch Client can be used with the command-line parameters described below. 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”.

The following table describes the command-line parameters for all operations.

**Table 15  All Operations**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/op=</td>
<td>Specifies an operation to perform:</td>
</tr>
<tr>
<td></td>
<td>• ActionScript</td>
</tr>
<tr>
<td></td>
<td>• BackupVersionToFile</td>
</tr>
<tr>
<td></td>
<td>• Blend</td>
</tr>
<tr>
<td></td>
<td>• CloseVersion</td>
</tr>
<tr>
<td></td>
<td>• Export</td>
</tr>
<tr>
<td></td>
<td>• Import</td>
</tr>
<tr>
<td></td>
<td>• Multi</td>
</tr>
<tr>
<td></td>
<td>• OpenVersion</td>
</tr>
<tr>
<td></td>
<td>• RefreshVersion</td>
</tr>
<tr>
<td></td>
<td>• RestoreVersionFromFile</td>
</tr>
<tr>
<td></td>
<td>• DeleteVersion</td>
</tr>
<tr>
<td></td>
<td>This parameter is required</td>
</tr>
<tr>
<td>/u=</td>
<td>Specifies the Data Relationship Management user name for login.</td>
</tr>
<tr>
<td>/pw=</td>
<td>Specifies the Data Relationship Management password for login.</td>
</tr>
<tr>
<td>/url=</td>
<td>Specifies the URL (including the port number) for the Process Manager application in the Data Relationship Management server processes. This URL is configured and is available in the Data Relationship Management configuration console. For example, &quot;net.tcp://localhost:5210/Oracle/Drm/ProcessManager&quot;</td>
</tr>
<tr>
<td></td>
<td>This parameter is required.</td>
</tr>
<tr>
<td>/log=</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>This parameter is optional. If not specified, the following name is used: DRM_Batch_ClientMMDDYYYY.log, where MM = month, DD = day, YYYY = year</td>
</tr>
<tr>
<td>/cfgfile=</td>
<td>Specifies the path and file name of the configuration file describing desired options. It is recommended that you put the value in double quotes.</td>
</tr>
<tr>
<td>/h[elp]</td>
<td>Displays a list of command-line parameters.</td>
</tr>
<tr>
<td>/timezoneid</td>
<td>Used for rendering of dates/times in output.</td>
</tr>
<tr>
<td></td>
<td>This parameter is optional. The default is the server's time zone.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/culturename</td>
<td>Local culture name, in the format: languagecode-country/regioncode where languagecode is a lowercase two-letter code derived from ISO 639-1 and country/regioncode is an uppercase two-letter code derived from ISO 3166. For example, U.S. English is en-US. This parameter is optional. The default is the server's culture.</td>
</tr>
<tr>
<td>/objectaccess</td>
<td>Object access method for accessing import, export, and blender objects. Valid values are Standard, System, or User or custom group. This parameter is optional. The default is Standard.</td>
</tr>
</tbody>
</table>

The following tables describe the command-line parameters that can be used with each operation.

### Table 16  Action Script Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/infile=</td>
<td>Specifies the path and file name for the input file. It is recommended that you put the value in double quotes. This parameter is required.</td>
</tr>
<tr>
<td>/delim=</td>
<td>Specifies the delimiter. Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- Tab</td>
</tr>
<tr>
<td></td>
<td>- Comma</td>
</tr>
<tr>
<td></td>
<td>- #nnn</td>
</tr>
<tr>
<td></td>
<td>where #nnn represents any valid 3-digit ASCII character value (0-255).</td>
</tr>
<tr>
<td></td>
<td>Tab is the default value.</td>
</tr>
<tr>
<td>/cols=</td>
<td>Specifies the column order. If supplied, must contain all column identifiers. A12345 is the default value.</td>
</tr>
<tr>
<td>/stripq=</td>
<td>Specifies whether to remove quoted strings. Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- Y[es]</td>
</tr>
<tr>
<td></td>
<td>- N[o]</td>
</tr>
<tr>
<td></td>
<td>- T[true]</td>
</tr>
<tr>
<td></td>
<td>- F[alse]</td>
</tr>
<tr>
<td></td>
<td>- 1</td>
</tr>
<tr>
<td></td>
<td>- 0</td>
</tr>
<tr>
<td></td>
<td>The default value is True.</td>
</tr>
<tr>
<td>/propertyrefbylabel</td>
<td>Look up property references by label (instead of name). Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- Y[es]</td>
</tr>
<tr>
<td></td>
<td>- N[o]</td>
</tr>
<tr>
<td></td>
<td>- T[true]</td>
</tr>
<tr>
<td></td>
<td>- F[alse]</td>
</tr>
<tr>
<td></td>
<td>- 1</td>
</tr>
<tr>
<td></td>
<td>- 0</td>
</tr>
</tbody>
</table>
Table 17  Blend Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/bl=</td>
<td>Specifies the name of blender to run. This parameter is required.</td>
</tr>
<tr>
<td>/sver=</td>
<td>Specifies the name for the source version to blend. This parameter is required.</td>
</tr>
<tr>
<td>/tver=</td>
<td>Specifies the name for the target version to blend. This parameter is required.</td>
</tr>
<tr>
<td>/nvera=</td>
<td>Specifies the name for the new version to create.</td>
</tr>
<tr>
<td>/nverd=</td>
<td>Specifies a description for the new version. You must enclose the value in double quotes.</td>
</tr>
</tbody>
</table>
| /save=    | Specifies whether to save the blended version or leave it detached. Specify one of the following values:  
  - Y[es], T[true], or 1  
  - N[o], F[alse], or 0  
  The default value is False. For information on saving versions, see “Saving Versions in the Batch Client” on page 175. |
| /blendlog=| Specifies the name of the blend log file (not the same as the Batch Client log file). |

Table 18  Close Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/abbrev=</td>
<td>Specifies the name for the version to close. This parameter is required.</td>
</tr>
</tbody>
</table>

Table 19  Export Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| /xtype    | Specifies the type of export: individual export or book. Specify one of the following values:  
  - E[xport]  
  - B[ook]  
  This parameter is required. |
| /fdate=   | Specifies the From date/time for transaction log exports (Optional)  
  **Note:** Date/time formats are determined by the regional settings on the client computer. |
| /tdate=   | Specifies the To date/time for transaction log exports (Optional)  
  **Note:** Date/time formats are determined by the regional settings on the client computer. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/outfile=</td>
<td>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 with /outfile.</td>
</tr>
<tr>
<td>/xname=</td>
<td>Specifies the name of the saved export to run. This parameter is required if running a single export.</td>
</tr>
<tr>
<td>/cver=</td>
<td>Specifies the name for the current version. This parameter is optional. If not specified, the system uses the value of the DefaultCurrentVersion system preference.</td>
</tr>
<tr>
<td>/pver=</td>
<td>Specifies the name for the previous version. This parameter is optional. If not specified, the system uses the value of the DefaultPreviousVersion system preference.</td>
</tr>
</tbody>
</table>
| /cvertype= | Specifies the version type for the current version. Valid values are:  
  - A[s Of]  
  - N[ormal]  
  N[ormal] is the default. |
| /pvertype= | Specifies the version type for the previous version. Valid values are:  
  - A[s Of]  
  - N[ormal]  
  N[ormal] is the default. |
| /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. |
| /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. |
| /bk= | Specifies the name of the saved book to run. This parameter is required if exporting a book. It is recommended that you put the value in double quotes. |
| /bkcmb= | Specifies whether to combine the export output into the file named in /outfile. Specify one of the following values:  
  - Y[es]  
  - N[o]  
  - T[rue]  
  - F[alse]  
  - 1  
  - 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 /outfile. The default value is stored in book. |
<p>| /prefile= | Specifies the path and file name to prepend to a combined export book. The default value is stored in book. |
| /pstfile= | Specifies the path and file name to append to a combined export book. The default value is stored in book. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/fhier=</td>
<td>Specifies the From (Previous) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/thier=</td>
<td>Specifies the To (Current) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/fabbrev=</td>
<td>Specifies the name for the top node of the From (Previous) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/tabbrev=</td>
<td>Specifies the name for the top node of the To (Current) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/hier=</td>
<td>Specifies the name for the hierarchy to export. Used for Hierarchy Export and other similar exports, but not used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/nabbrev=</td>
<td>Specifies the name of the top node to export. Used for Hierarchy Export and other similar exports, but not used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>/encoding</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

**Table 20 Import Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/infile=</td>
<td>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. This parameter is only required for imports configured to import from a client file. <strong>Caution!</strong> Do not use this parameter for imports configured to use an input file from an external server connection, otherwise an error occurs.</td>
</tr>
<tr>
<td>/iabbrev=</td>
<td>Specifies the name for the import to run. This parameter is required.</td>
</tr>
<tr>
<td>/implog=</td>
<td>Specifies the name of the import log file (not the same as the Batch Client log file.) This parameter is required.</td>
</tr>
<tr>
<td>/vabbrev=</td>
<td>Specifies the version name to be created by import. The default value is stored in the import profile.</td>
</tr>
<tr>
<td>/vdescr=</td>
<td>Specifies the description of the version to be created by import. The default value is stored in the import profile.</td>
</tr>
</tbody>
</table>
Parameter: /save=

Specifies whether to save the imported version or leave it detached. Specify one of the following values:

- Y[es]
- N[o]
- T[ru]e
- F[alse]
- 1
- 0

The default value is False.

For information on saving versions, see "Saving Versions in the Batch Client" on page 175.

---

**Table 21  Open Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vabbrev=</td>
<td>Specifies the name of the version to open. This parameter is required.</td>
</tr>
</tbody>
</table>

**Table 22  Refresh Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vabbrev=</td>
<td>Specifies the name of the version to refresh (close and re-open). This parameter is required.</td>
</tr>
</tbody>
</table>

**Table 23  Delete Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vabbrev=</td>
<td>Specifies the name for the version to delete. This parameter is required.</td>
</tr>
</tbody>
</table>

**Table 24  Restore Version from File Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/filename=</td>
<td>Specifies the name of the file to locate using the external connection (/conn).</td>
</tr>
<tr>
<td>/conn=</td>
<td>Specifies the name of the external connection.</td>
</tr>
<tr>
<td>/sver=</td>
<td>Specifies the names of versions in the source file to restore; delimited by /delim if there are multiple versions.</td>
</tr>
<tr>
<td>/tver=</td>
<td>Specifies the names of new versions, correlating to the /sver parameter; delimited by /delim if there are multiple versions.</td>
</tr>
<tr>
<td>/delim=</td>
<td>Printable delimiter character. Uses a single character such as colon (:), comma (,), dash (-), and so on.</td>
</tr>
<tr>
<td>/save=</td>
<td>Save versions after restoring - Y: Yes, N: No, T: True, F: False, 1, 0</td>
</tr>
</tbody>
</table>
Table 25   Backup Version to File Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vabbrev=</td>
<td>Specifies the names of versions to backup; delimited by /delim if there are multiple versions.</td>
</tr>
<tr>
<td>/conn=</td>
<td>Specifies the name of the external connection.</td>
</tr>
<tr>
<td>/filename=</td>
<td>Specifies the name of the backup file to create using the external connection (/conn).</td>
</tr>
<tr>
<td>/delim=</td>
<td>Printable delimiter character. Uses a single character such as colon (:), comma (,), dash (-), and so on.</td>
</tr>
</tbody>
</table>

**Configuration File Syntax**

In addition to using command-line parameters, the Batch Client can be run with a configuration file. 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.

Below is an example configuration file.

```plaintext
[General]
Operation=Export
UserName=mylogin
Password=mypassword
URL=net.tcp://localhost:5210/Oracle/Drm/ProcessManager
LogFileName=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 configuration file with multiple operations. Note that when using multiple operations, the operation key, for example Operation=Import, must be included in each section.

```plaintext
[General]
Operation=Multiple
UserName=mylogin
Password=mypassword
URL=net.tcp://localhost:5210/Oracle/Drm/ProcessManager
LogFileName=C:\Oracle\Data Relationship Management\rdsc.log
```

**Note:** Sections, keys, and values are not case sensitive.
[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"

The following tables describe the keys that can be used in each section of the configuration file.

**Table 26 - General Section of Configuration File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>Specifies the operation to perform. Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>● ActionScript</td>
</tr>
<tr>
<td></td>
<td>● BackupVersionToFile</td>
</tr>
<tr>
<td></td>
<td>● Blend</td>
</tr>
<tr>
<td></td>
<td>● CloseVersion</td>
</tr>
<tr>
<td></td>
<td>● Export</td>
</tr>
<tr>
<td></td>
<td>● Import</td>
</tr>
<tr>
<td></td>
<td>● Multi</td>
</tr>
<tr>
<td></td>
<td>● OpenVersion</td>
</tr>
<tr>
<td></td>
<td>● RefreshVersion</td>
</tr>
<tr>
<td></td>
<td>● RestoreVersionFromFile</td>
</tr>
<tr>
<td></td>
<td>● DeleteVersion</td>
</tr>
<tr>
<td></td>
<td>This key is required</td>
</tr>
<tr>
<td>UserName</td>
<td>Specifies the Data Relationship Management user name for login. This key is required.</td>
</tr>
<tr>
<td>Password</td>
<td>Specifies the Data Relationship Management password for login. This key is required.</td>
</tr>
</tbody>
</table>
### Table 27  Multiple Section of Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationCount</td>
<td>Specifies the number of operation sections in the file. Specify an integer greater than 0. This key is required.</td>
</tr>
<tr>
<td>OperationN</td>
<td>Specifies the label for an operation section where ( N ) is a number between 1 and the OperationCount value. This key is required.</td>
</tr>
</tbody>
</table>

### Table 28  Action Script Section of Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InFile</td>
<td>Specifies the path and file name for the input file. It is recommended that you put the value in double quotes. This key is required.</td>
</tr>
<tr>
<td>StripQuotedStrings</td>
<td>Specifies whether to remove quoted strings. Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- Y[es]</td>
</tr>
<tr>
<td></td>
<td>- N[o]</td>
</tr>
<tr>
<td></td>
<td>- T[ue]</td>
</tr>
<tr>
<td></td>
<td>- F[alse]</td>
</tr>
<tr>
<td></td>
<td>- 1</td>
</tr>
<tr>
<td></td>
<td>- 0</td>
</tr>
<tr>
<td></td>
<td>The default value is True.</td>
</tr>
</tbody>
</table>
### Key | Description
---|---
**Delimiter** | Specifies the delimiter. Specify one of the following values:  
- Tab  
- Comma  
- #nnn  
where #nnn represents any valid 3-digit ASCII character value (0–255).  
Tab is the default value.

**ColumnOrder** | Specifies the column order. If supplied, must contain all column identifiers. A12345 is the default value.

---

**Table 29  Blend Section of Configuration File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlendName</td>
<td>Specifies the name of blender to run. This key is required.</td>
</tr>
<tr>
<td>BlendLogFileName</td>
<td>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.</td>
</tr>
<tr>
<td>SourceVersionAbbrev</td>
<td>Specifies the name for the source version to blend. This key is required.</td>
</tr>
<tr>
<td>TargetVersionAbbrev</td>
<td>Specifies the name for the target version to blend. This key is required.</td>
</tr>
<tr>
<td>NewVersionAbbrev</td>
<td>Specifies the name for the new version to create.</td>
</tr>
<tr>
<td>NewVersionDescription</td>
<td>Specifies a description for the new version. You must enclose the value in double quotes.</td>
</tr>
</tbody>
</table>
| AutoSave | Specifies whether to save the blended version or leave it detached. Specify one of the following values:  
- Y[es]  
- N[o]  
- T[ru]e  
- F[alse]  
- 1  
- 0  
The default value is False.  
For information on saving versions, see "Saving Versions in the Batch Client" on page 175. |

---

**Table 30  Close Section of Configuration File**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersionAbbrev</td>
<td>Specifies the name for the version to close. This key is required.</td>
</tr>
</tbody>
</table>
### Table 31  Export Section of Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
</table>
| ExportType        | Specifies the type of export: individual export or book. Specify one of the following values:  
|                   | - E[xport]  
|                   | - B[ook]  
|                   | This key is required.                                                                                                                                                                                   |
| OutFile           | 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. This key is required.                             |
| ExportName        | Specifies the name of the saved export to run. This key is required if running a single export.                                                                                                |
| CurrentVersionAbbrev | Specifies the name for the current version. If not specified, the name of the default version is used.                                                                                                    |
| PrevVersionAbbrev | Specifies the name for the previous version. If not specified, the name of the default version is used.                                                                                                  |
| BookName          | Specifies the name of the saved book to run. This key is required if exporting a book. Enclosing value in double quotes is recommended.                                                                 |
| CombineExportOutput | Specifies whether to combine the export output into the file named in /outfile. Specify one of the following values:  
|                   | - Y[es]  
|                   | - N[o]  
|                   | - T[ru]e  
|                   | - F[alse]  
|                   | - 1  
|                   | - 0  
<p>|                   | 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 /outfile. The default value is stored in book. |
| PreFileName       | Specifies the path and file name to prepend to a combined export book. The default value is stored in book.                                                                                          |
| PostFileName      | Specifies the path and file name to append to a combined export book. The default value is stored in book.                                                                                        |
| FromHierarchy     | Specifies the From (Previous) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.                                                                                             |
| ToHierarchy       | Specifies the To (Current) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.                                                                                                 |
| FromTopNodeAbbrev | Specifies the name of the top node of the From (Previous) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.                                                                  |
| ToTopNodeAbbrev   | Specifies the name of the top node of the To (Current) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports                                                                  |
| HierarchyCount    | Specifies the number of hierarchy/top node pairs defined.                                                                                                                                             |</p>
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy$n$</td>
<td>Specifies the Hierarchy to export, where $n$ is a number (1 through HierarchyCount) identifying the current hierarchy/top node pair (for example, Hierarchy1). Used for Hierarchy Export and other similar exports, but not used for Hierarchy Compare and Hierarchy Difference exports.</td>
</tr>
<tr>
<td>TopNodeAbbrev$n$</td>
<td>Specifies the name of the top node 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.</td>
</tr>
<tr>
<td>FromDate</td>
<td>Specifies the From date/time for transaction log exports (Optional)</td>
</tr>
<tr>
<td>ToDate</td>
<td>Specifies the To date/time for transaction log exports (Optional)</td>
</tr>
</tbody>
</table>
| CurrentVersionType | Specifies the version type for the current version. Valid values are:  
  - A[s Of]  
  - N[ormal]  
  
The default is N[ormal].                                                                                                                                                                                                                                                           |
| CurrentAsOfDate  | Specifies the As Of Date/Time to use when creating an As Of Version for the current version for the export.                                                                                                                                                                                                                                  |
| PrevVersionType  | Specifies the version type for the previous version. Valid values are:  
  - A[s Of]  
  - N[ormal]  
  
The default is N[ormal].                                                                                                                                                                                                                                                           |
| PrevAsOfDate     | Specifies the As Of Date/Time to use when creating an As Of Version for the previous version for the export.                                                                                                                                                                                                                                 |

Table 32  Import Section for Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InFile</td>
<td>Specifies the path and file name for the input file. It is recommended that you put the value in double quotes. This key is required.</td>
</tr>
<tr>
<td>ImportAbbrev</td>
<td>Specifies the name for the import to run. This key is required.</td>
</tr>
<tr>
<td>ImportLogFileName</td>
<td>Specifies the path and file name of the import log file. This is not the same log file as defined by the LogFileName key in the [General] section.) The default file name is DRM_Batch_ClientMMDDYYYY.log, where MM = month, DD = day, YYYY = year</td>
</tr>
<tr>
<td>VersionAbbrev</td>
<td>Specifies the version name to be created by import. The default value is stored in the import profile.</td>
</tr>
<tr>
<td>VersionDescription</td>
<td>Specifies the description of the version to be created by import. The default value is stored in the import profile.</td>
</tr>
</tbody>
</table>
Key | Description
---|---
AutoSave | Specifies whether to save the imported version or leave it detached. Specify one of the following values:
  - Y[es]
  - N[o]
  - T[rue]
  - F[alse]
  - 1
  - 0
The default value is False.
For information on saving versions, see “Saving Versions in the Batch Client” on page 175.

Table 33  Open Section for Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersionAbbrev</td>
<td>Specifies the name of the version to open. This key is required.</td>
</tr>
</tbody>
</table>

Table 34  Refresh Section of Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersionAbbrev</td>
<td>Specifies the name of the version to refresh (close and re-open). This key is required. This key is required.</td>
</tr>
</tbody>
</table>

Table 35  Delete Section of Configuration File

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersionAbbrev</td>
<td>Specifies the name for the version to delete. This parameter is required.</td>
</tr>
</tbody>
</table>

**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_mmddyyyy.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 by a scheduling program or batch file and appropriate action can automatically be taken.

Table 36  Error Level Values

<table>
<thead>
<tr>
<th>Error Level Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful/normal termination</td>
</tr>
<tr>
<td>Error Level Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Unknown error</td>
</tr>
<tr>
<td>2</td>
<td>Batch operation completed but generated warnings</td>
</tr>
<tr>
<td>3</td>
<td>Batch operation did not complete due to error</td>
</tr>
<tr>
<td>100</td>
<td>Error communicating to or returned from the Oracle Data Relationship Management Server</td>
</tr>
<tr>
<td>200</td>
<td>Error occurred during initialization of the Batch Client</td>
</tr>
<tr>
<td>210</td>
<td>An invalid parameter value was passed</td>
</tr>
<tr>
<td>220</td>
<td>Invalid URL</td>
</tr>
<tr>
<td>230</td>
<td>Invalid user name or password</td>
</tr>
</tbody>
</table>

**Note:** If other error level values are returned, the error is from the Windows operating system. See Windows documentation for more information.
Glossary

**alternate hierarchy**  A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

**ancestor**  A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

**application**  1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. 2) A related set of dimensions and dimension members that are used to meet a specific set of analytical requirements, reporting requirements, or both.

**application administrator**  A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

**artifact**  An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

**attribute**  A characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

**authentication**  Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

**book**  1) In Financial Reporting, a container that holds a group of similar documents. Books may specify dimension sections or dimension changes. 2) In Data Relationship Management, a collection of exports that can be run together as a group. Export results can be combined together or output separately.

**business rules**  Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

**categories**  Groupings by which data is organized. For example, Month.

**child**  A member with a parent above it in the database outline.

**column**  In Data Relationship Management, a field of data associated with an import source or the results of a query, compare, validation, or export.

**content**  Information stored in the repository for any type of file.

**custom property**  A property of a dimension or dimension member that is created by a user.

**data model**  A representation of a subset of database tables.

**database connection**  A file that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

**descendant**  Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

**drill-down**  Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

**encoding**  A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. See also code page, locale.
**external authentication** Logging on to Oracle EPM System products with user information stored outside the application. The user account is maintained by the EPM System, but password administration and user authentication are performed by an external service, using a corporate directory such as Oracle Internet Directory (OID) or Microsoft Active Directory (MSAD).

**file delimiter** A character, such as a comma or tab, that separates fields in a data source.

**filter** A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

**format string** 1) In Essbase, a method for transforming the way cell values are displayed. 2) In Data Relationship Management, a parameter of a Format or Formatted Date derived property that indicates the format in which a property value should be returned.

**formula** In Data Relationship Management, business logic used by a derived property to dynamically calculate a property value.

**function** In Data Relationship Management, a syntactic element of a derived property formula that accepts parameters and returns dynamic values.

**generation** A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

**group** A container for assigning similar access permissions to multiple users.

**identity** A unique identification for a user or group in external authentication.

**jobs** Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

**lifecycle management** The process of migrating an application, a repository, or individual artifacts across product environments.

**migration** The process of copying applications, artifacts, or users from one environment or computer to another; for example, from a testing environment to a production environment.

**migration log** A log file that captures all application migration actions and messages.

**native authentication** The process of authenticating a user name and password from within the server or application.

**null value** A value that is absent of data. Null values are not equal to zero.

**parents** The entities that contain dependent entities that report directly to them. Because parents are entities associated with at least one node, they have entity, node, and parent information associated with them.

**permission** A level of access granted to users and groups for managing data or other users and groups.

**precision** Number of decimal places displayed in numbers.

**provisioning** The process of granting users and groups specific access permissions to resources.

**record** In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

**repository** Storage location for metadata, formatting, and annotation information for views and queries.

**resources** Objects or services managed by the system, such as roles, users, groups, files, and jobs.

**role** The means by which access permissions are granted to users and groups for resources.

**security platform** A framework enabling Oracle EPM System products to use external authentication and single sign-on.

**services** Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

**Shared Services Registry** The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.
**sibling**  A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.