

## **Oracle® Hyperion Data Relationship Management, Fusion Edition**

*Oracle® Hyperion Data Relationship Steward*

*Oracle® Hyperion Data Relationship Management for Oracle Hyperion Enterprise Planning Suite*

*Oracle® Hyperion Data Relationship Management for Oracle Hyperion Financial Close Suite*

*Oracle® Hyperion Data Relationship Management for Customer Hub*

*Oracle® Hyperion Data Relationship Management Read Only Access*

### **User's Guide**

RELEASE 11.1.2.1

Data Relationship Management User's Guide, 11.1.2.1

Copyright © 1999, 2011, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited. The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS:**

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

# Contents

---

<b>Documentation Accessibility</b> .....	9
<b>Chapter 1. Getting Started</b> .....	11
About Data Relationship Management .....	11
Accessing Data Relationship Management .....	11
Changing Passwords .....	12
Security .....	12
Naming Metadata Objects .....	13
<b>Chapter 2. User Experience Differences</b> .....	15
Release Overview .....	15
New Feature Terms .....	16
Changes to Existing Feature Terms .....	16
Obsolete Terms .....	17
Task Group Descriptions .....	17
Key User Experience Differences .....	18
Logon .....	19
Browse .....	19
Query .....	22
Compare .....	23
Script .....	23
Import .....	24
Blend .....	24
Export .....	24
Audit .....	25
Administer .....	25
<b>Chapter 3. Managing Versions</b> .....	27
Data Relationship Management Version Life Cycle .....	27
Version Type and Status .....	29
Creating Versions .....	29
Copying Versions .....	29
Saving Versions .....	30

Creating As-Of Versions .....	30
Editing Version Properties .....	31
Assigning Validations to Versions .....	31
Unloading and Deleting Versions .....	32
<b>Chapter 4. Managing Hierarchies .....</b>	<b>33</b>
Creating Hierarchies .....	33
Editing Hierarchy Properties .....	34
Enabling and Disabling Node Types .....	35
Opening Hierarchies .....	36
Creating Hierarchy Groups .....	36
Assigning Hierarchies to Hierarchy Groups .....	37
Assigning Controlled Properties .....	37
Assigning Controlled Node Access Groups .....	37
Assigning Validations .....	38
Deleting Hierarchies .....	38
<b>Chapter 5. Working with Nodes .....</b>	<b>41</b>
About Nodes .....	41
Adding and Inserting Nodes .....	42
ID Function .....	43
Modeling a Node .....	44
Moving and Ordering Nodes .....	45
Relaxed Move Option .....	46
Sorting Nodes .....	46
Node Deletion Methods .....	47
Merging Nodes .....	47
Deleting Nodes .....	48
Removing Nodes .....	48
Annulling Nodes .....	49
Inactivating Nodes .....	49
Reactivating Nodes .....	50
Assigning Validations .....	50
Orphan Nodes .....	50
Viewing Orphan Nodes .....	51
Deleting Orphan Nodes .....	51
Navigating in Tree View .....	51
Finding Nodes .....	52
Node Security .....	53
Using Shared Nodes .....	53

Shared Node Properties . . . . .	54
Enabling Shared Nodes . . . . .	55
Adding and Inserting Shared Nodes . . . . .	55
Removing and Deleting Shared Nodes . . . . .	56
Renaming Shared Nodes . . . . .	56
Converting a Shared Node to a Primary Node . . . . .	56
Displaying Shared Nodes . . . . .	57
<b>Chapter 6. Managing Properties . . . . .</b>	<b>59</b>
Property Categories . . . . .	59
System Properties . . . . .	60
Stats Properties . . . . .	60
Property Value Scope and Origin . . . . .	61
Property Status . . . . .	62
Editing Property Values . . . . .	63
<b>Chapter 7. Validating Data . . . . .</b>	<b>65</b>
Validating Versions . . . . .	65
Validating Hierarchies . . . . .	66
Validating Nodes . . . . .	66
Navigating Validation Results . . . . .	67
Navigating Version and Hierarchy Results in List View . . . . .	67
Navigating Node Results in List View . . . . .	67
Navigating Node Results in Tree View . . . . .	67
Downloading Node Results . . . . .	68
Clearing Validation Results . . . . .	68
<b>Chapter 8. Using Queries . . . . .</b>	<b>69</b>
Query Types . . . . .	69
Creating Queries . . . . .	70
Navigating Query Results . . . . .	71
Navigating in List View . . . . .	72
Navigating in Tree View . . . . .	72
Downloading Query Results . . . . .	72
Managing Queries . . . . .	73
Opening Queries . . . . .	73
Copying Queries . . . . .	73
Deleting Queries . . . . .	74
<b>Chapter 9. Comparing Hierarchies . . . . .</b>	<b>75</b>
Compare Types . . . . .	75

Visual Compare .....	75
Structure Compare .....	75
Property Compare .....	76
Rename Compare .....	76
Creating Compares .....	76
Navigating Compare Results .....	78
Navigating in List View .....	78
Navigating in Tree View .....	78
Downloading Compare Results .....	79
Managing Compares .....	79
Opening Compares .....	79
Copying Compares .....	80
Deleting Compares .....	80
<b>Chapter 10. Working with Action Scripts .....</b>	<b>81</b>
Loading Action Scripts From Flat Files .....	81
Downloading Action Script Results .....	82
Required Action Script Parameters .....	83
Loading Action Scripts From a Transaction Log .....	86
Loading Action Scripts From a Node Model .....	87
<b>Chapter 11. Working with Imports .....</b>	<b>89</b>
Import Sections .....	89
Creating Imports .....	90
Running, Copying, and Deleting Imports .....	93
<b>Chapter 12. Blending Versions .....</b>	<b>95</b>
Creating Blenders .....	95
Running, Copying, and Deleting Blenders .....	99
<b>Chapter 13. Working with Exports .....</b>	<b>101</b>
Creating Exports .....	101
Creating Hierarchy Exports .....	102
Creating Hierarchy XML Exports .....	107
Creating Generation Exports .....	108
Creating Compare Exports .....	112
Creating Difference Exports .....	116
Creating Version Exports .....	119
Creating Property List Exports .....	122
Creating Property Lookup Exports .....	123
Creating Transaction Log Exports .....	125

Creating Merge Log Exports .....	128
Creating Export Books .....	130
Running, Copying, and Deleting Exports and Export Books .....	131
<b>Chapter 14. Working with Change Requests .....</b>	<b>133</b>
<b>Chapter 15. Auditing Transaction and Request History .....</b>	<b>135</b>
Auditing Transaction History .....	135
Data Actions in Transaction History .....	136
Administrative Actions in Transaction History .....	138
Auditing Requests .....	141
<b>Chapter 16. Using the Data Relationship Management Batch Client .....</b>	<b>143</b>
Securing the Batch Client .....	144
Setting up Batch Client Credentials .....	144
Running the Batch Client .....	144
Dependencies .....	145
Saving Versions in the Batch Client .....	145
Command Line Syntax .....	145
Configuration File Syntax .....	151
Outputs and Result Codes .....	157
<b>Glossary .....</b>	<b>159</b>
<b>Index .....</b>	<b>163</b>





---

# Documentation Accessibility

---

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

## Access to Oracle Support for Hearing-Impaired Customers

Oracle customers have access to electronic support through My Oracle Support or by calling Oracle Support at 1.800.223.1711. Hearing-impaired customers in the U.S. who wish to speak to an Oracle Support representative may use a telecommunications relay service (TRS). Information about the TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html/>, and a list of telephone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>. International hearing-impaired customers should use the TRS at +1.605.224.1837. An Oracle Support engineer will respond to technical issues according to the standard service request process.



# 1

# Getting Started

---

## In This Chapter

About Data Relationship Management .....	11
Accessing Data Relationship Management.....	11
Security .....	12
Naming Metadata Objects .....	13

## About Data Relationship Management

Oracle Hyperion Data Relationship Management, Fusion Edition 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.

- 3 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 CSS.

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

## Security

User access is determined by:

- 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.
- 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.

- User role assignment — For more information, see the *Oracle Hyperion Data Relationship Management Administrator's Guide*.
- 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.

## Naming Metadata Objects

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



---



# User Experience Differences

---

---

## In This Chapter

<a href="#">Release Overview</a> .....	15
<a href="#">Task Group Descriptions</a> .....	17
<a href="#">Key User Experience Differences</a> .....	18

## Release Overview

In the 11.1.2 release of Data Relationship Management, a new user interface was introduced. This user interface replaces the Data Relationship Management Win32 client and the ActiveX Web Publishing client.

The new user interface:

- Provides one user experience for all Data Relationship Management user types
- Offers Web accessibility to all Data Relationship Management Win32 features except the Console
- Allows user-authenticated and anonymous access (previously provided by Web Publishing)
- Simplifies the presentation of data and access to features for less-technical users
- Provides a task-driven approach to user interaction vs. a tool-driven approach
- Minimizes any degradation in richness of features due to Web accessibility
- Supports internationalization of the product
- Eliminates the need for additional client installation to access the Data Relationship Management user interface
- Minimizes the need for training new users or retraining existing users

For more information, see:

- [New Feature Terms](#)
- [Changes to Existing Feature Terms](#)
- [Obsolete Terms](#)

## New Feature Terms

Term	Definition
Application	An instance of Data Relationship Management. The user interface can access multiple applications.
As Sibling	An Add/Insert/Move/Put action allows placement of a node as a sibling to another node.
Clipboard	Copy nodes here to work with them across multiple hierarchy windows.
Connection	External locations relative to the server that can be read from and written to.
Namespace	Uniquely identifies a set of names to retain referential integrity when metadata objects having different origins but the same names are combined into the same application.
Object Access	User metadata objects can now be: <ul style="list-style-type: none"> <li>● User – Personal objects that are only available to an individual user to view and run. All user roles have the ability to create and manage objects of this access level.</li> <li>● Standard – Public objects that are available to all users to view and run. Only Data Manager role users have the ability to create and manage objects of this access level.</li> <li>● System – Restricted objects that are only available to Application Administrator role users to view and run. Only those users have the ability to create and manage objects of this access level.</li> </ul>
Server File	A type of connection used to access a network file system or an FTP directory.
Shortcuts	Context-sensitive links that navigate the user to another task group retaining the current selection.
Use Fast Deletes	During an export, enables a bulk deletion on a target database table.
User Roles	Permission-based roles that control user access to product features. Roles: <ul style="list-style-type: none"> <li>● Anonymous User</li> <li>● Workflow User</li> <li>● Interactive User</li> <li>● Data Creator</li> <li>● Data Manager</li> <li>● Access Manager</li> <li>● Application Administrator</li> </ul>
Version and Hierarchy Owner	The Owner has full management privileges to a version/hierarchy. A user with the permission to create a version or hierarchy is assigned as the Version Owner or Hierarchy Owner.

## Changes to Existing Feature Terms

Term	Description of Change
Action Script	Formerly the Automator interface or an Automator script.
Controlled Property	For a hierarchy, formerly assigning a controlling hierarchy to a property
Download	Formerly Save to File or Print



<b>Term</b>	<b>Description of Change</b>
Hierarchy Group	Formerly a System Category, which is used for grouping hierarchies
Locate	Formerly the Synchronize By Name feature
Load Status	Formerly represented as bold-faced print for an open/closed version
Match	Formerly the Synchronize By Property feature
Paste Properties	Properties can be pasted for any node that is taken and copied to the clipboard.
Put	Extended to support reordering children in addition to Insert and Move
Search	Formerly the Find Node feature
Take/Copy Properties	Includes the former Take and Copy Properties features
Validation (Batch)	Formerly Verification (and Verify)

## Obsolete Terms

<b>Term</b>	<b>Comment</b>
Abbrev	Replaced by Name. Except for the Abbrev() formula function.
Automator	Replaced by Action Script
Export Preview	Replaced by Download
System Category	Replaced by Hierarchy Group
User Types <ul style="list-style-type: none"> <li>● User</li> <li>● Functional Administrator</li> <li>● Security Administrator</li> <li>● System Administrator</li> </ul>	Replaced by User Roles and Permissions
Verification (and Verify)	Replaced by Batch Validation (and Validate)
Web Publishing	Replaced by Anonymous Access

## Task Group Descriptions

Data Relationship Management features and content are organized into logical task groups in a navigation menu. Only one task group is in focus at once. All task groups are displayed so you can navigate to and change focus from a task group. Following are descriptions of each task group in the Data Relationship Management Web user interface.

- **Browse** – The Browse task group is the default task group in focus when you enter the Data Relationship Management user interface. In the Browse task, you can browse versions,

hierarchies, nodes, and their properties. You also can modify the objects to which you have access. The Browse task is a common starting point for both authenticated and anonymous users. When you browse a hierarchy for a given version, each hierarchy opens in a separate tab. You can have multiple hierarchies open but only one in focus at once. Administrative actions for versions and hierarchies also can be performed here.

- **Query** – In the Query task group, you manage and run property queries and work with the results of the query process. The details of each new or existing query open in a separate tab. You can have multiple queries open but only one in focus at once. The results of each query can be edited (based on a user’s rights to the nodes and properties that are returned), so you can make changes without navigating to the Browse task group.
- **Compare** – In the Compare task group, you manage and run compares and view and work with the results of the comparison process. The details of each new or existing compare open in a separate tab. You can have multiple compares open, but only one can be in focus at once. The results of each compare can be edited (based on a user’s rights to the nodes and properties that are returned), so you can make changes without navigating to the Browse task group.
- **Script** – In the Script task group, you create and run Action Scripts and view the results of these scripts. There is no concept of saved profiles for action scripts and therefore there is no list of existing action scripts from which to select. You can have only one action script (and its results) open at once. An action script that has been loaded from another source can be edited directly in the Script task.
- **Import** – In the Import task group, you manage and run imports and view the results of the import process. The details of each new or existing import open in a separate tab. You can have multiple imports open but only one in focus at once.
- **Blend** – In the Blend task group, you manage and run blenders and view the results of the blender process. The details of each new or existing blender open in a separate tab. You can have multiple blenders open but only one in focus at once.
- **Export** – In the Export task group, you manage and run exports and view the results of the export process. The details of each new or existing export open in a separate tab. You can have multiple exports open but only one in focus at once.
- **Audit** – In the Audit task group, you query and view history for transactions and requests. You can select an audit type to view transactions or requests, but only one type can be queried and viewed at once.
- **Administer** – In the Administer task group, you can manage metadata, security, and preferences for the Data Relationship Management application. The default task is creating a new metadata/security object or opening an existing object to administer. If the user has application administrator permissions, the default task is managing metadata. If the user is an access manager, then the default task is managing security.

## Key User Experience Differences

The following sections describe how major actions are performed in Data Relationship Management releases before 11.1.2 and in release 11.1.2 and after:

- [Logon](#)
- [Browse](#)
- [Query](#)
- [Compare](#)
- [Script](#)
- [Import](#)
- [Blend](#)
- [Export](#)
- [Audit](#)
- [Administer](#)

## Logon

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Logging on	You create a connection to a Data Relationship Management application from the Data Relationship Management Client at this URL: <code>http://servername/mdm_ntier</code> .	You access the Data Relationship Management Web Client from a Web browser at this URL: <code>http://servername:port/drm-web-client</code> . Select a Data Relationship Management application to log on to.

## Browse

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Managing versions and hierarchies	Functional and System Admin user types can create and manage all versions and hierarchies.	Data Creator and Data Manager roles can create versions and hierarchies. Data Creators become the owner of any version/hierarchy they create.  Data Creator users can manage any version of which they are the owner.  Data Manager users can manage all versions and hierarchies.
Grouping hierarchies	System Admin users can create/delete System Categories from the Admin menu.  Hierarchies can be assigned to System Categories from the Admin menu or individually by using hierarchy-level properties in the Version Control panel.	Data Manager users can create hierarchy groups and assign hierarchies to them directly from the Browse task group.  Hierarchy groups can be managed from the Administer task group.
Viewing hierarchies	You can have one or more hierarchies open in separate windows and displayed side-by-side horizontally or vertically.	You can have one or more hierarchies open in separate tabs, but only one tab can be in focus at a time.  You can view multiple hierarchies side-by-side in the results of a Compare.

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing node properties	<p>Within a hierarchy tree, you can view nodes by name, description, or a custom Display By String.</p> <p>In the Property Grid, you can view all properties for a selected node by property category. You can position the Property Grid for each tree to accommodate different user perspectives (single vs. multiple hierarchies).</p>	<p>Within a hierarchy tree, you can use View By to view nodes by any combination of properties in multiple columns.</p> <p>In the Property Grid, on the Properties tab, you can view all properties for a selected node by property category. The single Property Grid is used for one hierarchy during a Browse task and for multiple hierarchies during a Compare task.</p>
Inserting and moving nodes	<p>By dragging or by using Take/Put, you can insert or move a node in a hierarchy tree.</p> <p>By dragging from the hierarchy's Child List, you can insert or move multiple nodes. You cannot insert or move nodes from separate parents.</p> <p>You can insert or move a node only as a Child of a limb node.</p>	<p>Using Take/Put, you can insert a node into a hierarchy. By dragging or by using Take/Put, you can move a node within a hierarchy into a hierarchy during a Browse task. During a Compare task, you can insert a node by dragging or by using Take/Put.</p> <p>Using Take/Put from the Clipboard, you can insert or move multiple nodes. You can insert and move nodes from separate parents.</p> <p>You can insert a node as a Child or Sibling of another node.</p>
Ordering children	<p>Using the Order Children context menu, you can reorder the child nodes of a given parent node.</p> <p>By dragging or using arrow buttons, you can reposition nodes in the child list in a separate window.</p>	<p>When you Add/Insert/Move a node, you can order it as a Sibling of another node.</p> <p>By dragging or by using Take/Put from the Clipboard, you can reorder nodes in the hierarchy tree as Siblings.</p>
Copying and pasting properties	<p>When you add a node, you can copy the properties of another node.</p> <p>Using Copy/Paste, you can copy the properties of a node to another node.</p> <p>You must re-select the set of properties to be pasted for each target node.</p>	<p>Using Take/Paste, you can copy a node and its properties to the Clipboard and paste the properties of a node to another node.</p> <p>The set of properties selected is saved and available for pasting to multiple nodes.</p>
Viewing inactive nodes	<p>Inactive nodes are displayed in the hierarchy tree and are marked with an icon for easy identification.</p> <p>In the Web Publishing client, you can show/hide inactive nodes in the hierarchy tree.</p>	<p>Inactive nodes are displayed in the hierarchy tree and are marked with an icon for easy identification. You can hide or show inactive nodes in the hierarchy tree.</p> <p>You can display the Node Inactive property in a column for each hierarchy node in the tree.</p>
Removing and deleting nodes	<p>You can remove or delete multiple nodes from the Child List of a hierarchy window.</p>	<p>You can remove or delete single nodes except when using Annul All Below.</p>
Searching for a node	<p>You can use Global Find to search for nodes within a version. Search results are returned in a multi-tabbed hotlist, grouped by hierarchy.</p> <p>You can use Find to search for nodes within a hierarchy. Search results can be returned in a hotlist and/or marked tree.</p> <p>You can use multi-character wildcard and exact match searches.</p> <p>You can use Goto to navigate from a node in the hotlist to a node in a hierarchy tree.</p>	<p>You can use Global Find on the Find Nodes tab to search for nodes within a version. Search results are returned in a list below the search box, grouped by hierarchy.</p> <p>You can use Find to search for nodes below a selected node within a hierarchy. Search results are returned on the Relationships tab.</p> <p>You can use single- and multi-character wildcards and escape characters for exact match searches.</p> <p>You can use Goto to navigate from a node on the Relationships tab to a node in a hierarchy tree.</p>

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Finding related nodes	<p>Using the Lists context menu, you can view lists of related nodes (descendants, children, links, and so on).</p> <p>Using Synchronize By Name, you can find nodes that have the same name in open hierarchies.</p> <p>Using Synchronize By Property, you can find nodes that have the same value for a selected property in open hierarchies.</p> <p>Synchronize results are returned as marked nodes in open hierarchy windows.</p>	<p>Using the Related menu, you can view lists of related nodes on the Relationships tab.</p> <p>Using Goto, you can navigate from a node on the Relationships tab to a node in a hierarchy tree.</p>
Viewing shared nodes	<p>Implicit shared nodes are displayed in the hierarchy tree and child list in italics.</p> <p>You can view primary and explicit/implicit shared nodes together in a list using Show Shared Nodes.</p>	<p>Explicit and implicit shared nodes are identified with different node icons. You can display the icon for the node in a list or hierarchy tree.</p> <p>You can view primary and explicit/implicit shared nodes together in a list by using the Shared Nodes menu item of the Related menu.</p>
Running a query/compare/export	<p>The current browsed node in a hierarchy is used as the default for a new local property query.</p>	<p>Using links on the Shortcuts tab, you can pass the current browsed version/hierarchy/node as context to a query, compare, or export by using links on the Shortcuts tab.</p>
Viewing transaction history	<p>You can use the View Transaction History context menu to view transaction history for a hierarchy node.</p>	<p>You can use the Transaction History shortcut on the Shortcuts tab to view transaction history for a hierarchy node.</p>
Assigning node access groups	<p>You assign node access groups in the Property Grid on the Leaf Access and Limb Access tabs.</p>	<p>You assign node access groups using the Assign Node Access menu item.</p>
Assigning controlling hierarchies	<p>For each version, you can define a controlling hierarchy for a property definition from the Admin menu.</p> <p>For each version, you can define a controlling hierarchy for a node access group from the Admin menu.</p> <p>You can view controlled properties for a selected hierarchy.</p>	<p>You can assign controlled properties to a hierarchy in a version using the Assign Control-Properties menu item.</p> <p>You can view the controlling hierarchy for a property in the Property Grid.</p>
Assigning validations	<p>You can assign validations and verifications at a version, hierarchy, or node level.</p> <p>Each validation and verification can run only as either real-time or batch and are assigned as enabled or disabled.</p> <p>Validations at a the node level are assigned in the Property Grid on the Validation and Verification tabs.</p>	<p>You can assign validations at a version, hierarchy, or node level using the Assign Validations menu item.</p> <p>You can run each validation as real-time and/or batch. You select the mode during assignment.</p>
Running batch validations	<p>You can run assigned verifications using Verify-Assigned.</p> <p>You can run a specific set of verifications using Verify-Selected.</p> <p>Verification results can be displayed in a hotlist and/or marked tree window.</p>	<p>You can run assigned batch validations using the Validate-Assigned menu item.</p> <p>You can run a specific set of batch validations in batch using Validate-Selected.</p> <p>Batch validation results are displayed on the Validation Results tab and marked in hierarchy trees.</p>

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Navigating validation results	<p>From the View Reasons context menu, you can view the reasons for failure for each node in a memo box.</p> <p>After they have been addressed, failed nodes can be hidden from a hotlist in a checklist and can be shown again in the entire list.</p>	<p>For each node that failed validation, you can view the reasons for failure as detail rows.</p> <p>You can copy failed nodes from the validation results list to the Clipboard using Take/Copy Properties. You can remove nodes from the Clipboard after they have been addressed.</p>
Saving and printing hierarchies	<p>You can print copies of hierarchy trees.</p> <p>Using the Print From Here menu, you can print a branch of a hierarchy.</p>	<p>You can download hierarchy trees downloaded to a XLS, RTF, or PDF file to be saved or printed.</p> <p>You can save or print a branch of a hierarchy by downloading the tree when viewing from the top node of the branch.</p>

## Query

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing and managing queries	<p>You can view and edit queries that are specific to a user.</p> <p>Any user can view standard queries, but only a Functional Admin user type or higher can manage them.</p>	<p>You can view and edit queries that are specific to a user.</p> <p>Any user can view standard queries, but only a user with the Data Manager role can manage them.</p> <p>Only users with the Application Administrator role can view and manage system queries.</p>
Selecting a query type	<p>You can create and select global property queries from the Tools-Global Property Query menu.</p> <p>You can create and select local property queries from the Tools-Property Query menu.</p> <p>Saved global and local queries are listed separately.</p>	<p>Global and local property queries are created/selected from the Query task group.</p> <p>Saved global and local queries are listed together and grouped by object access (User, Standard, System).</p>
Defining return properties	<p>Properties displayed in columns for query results are controlled by settings in the Tools-Options menu.</p> <p>In each query, you can optionally include criteria properties to display as additional columns.</p>	<p>When you run the query, you select properties to display in columns for query results.</p> <p>In each query, you can optionally include criteria properties to display as additional columns.</p>
Displaying query results	<p>You can display query results in a hotlist or in a marked tree in an open hierarchy window.</p>	<p>You can display query results in a list or in a marked tree on a separate Query tab.</p>
Saving and printing results	<p>You can print query results or save to a text file.</p>	<p>You can download query results to a CSV, XLS, RTF, or PDF file for saving or printing.</p>

## Compare

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing and managing compares	<p>You can view and edit compares that are specific to a user.</p> <p>Any user can view standard compares, but only users with the Functional Admin user type or higher can manage them.</p>	<p>You can view and edit compares that are specific to a user.</p> <p>Any user can view standard compares, but only Data Manager user roles can manage them.</p> <p>Only users with the Application Administrator user role can view and manage system compares.</p>
Selecting a compare type	<p>You create and select structure compares from the Tools-Compare menu.</p> <p>You create and select property compares from the Tools-Property Compare menu.</p> <p>You create and select renamed node compares from the Tools-Renamed Node Compare menu.</p> <p>Saved compares are listed separately for each compare type.</p>	<p>You create and select structure, property, and renamed node compares from the Compare-Select task group.</p> <p>Saved compares are listed together and grouped by object access (User, Standard, System).</p>
Defining return properties	<p>Settings in the Tools-Options menu determine the properties displayed in columns for compare results.</p>	<p>When you run a compare, you select the properties to display in columns for compare.</p> <p>For each compare, you can optionally include criteria properties to display as additional columns.</p>
Displaying query results	<p>You can display compare results in a hotlist or in a marked tree in an open hierarchy window.</p>	<p>You can display compare results in a list or in a marked tree on a separate Compare tab.</p>
Saving and printing results	<p>You can print compare results or save to a text file.</p>	<p>You can download compare results to a CSV, XLS, RTF, or PDF file for saving or printing.</p>

## Script

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Processing a script of changes	<p>From the File-Automator menu, you can load a script of incremental changes.</p>	<p>On the Script page, you can load an action script from a file.</p>
Loading a script from a file	<p>You can load an Automator script from a client file using the File-Automator menu item.</p>	<p>You can load an action script from a client file using the File source type.</p>
Creating a script from transaction history	<p>From the Tools-Transaction History menu, you can generate an Automator script from transaction history.</p>	<p>You can generate an action script from transaction history using the Transaction History source type.</p>
Creating a script from a node model	<p>From the Model After context menu, you can generate an Automator script from a node model.</p>	<p>You can generate an action script from a node model using the Node Model source type.</p>
Editing a script of changes	<p>You can edit automator scripts only before loading from a file.</p>	<p>You can edit action scripts after they are loaded from a file or generated from transaction history or a node model.</p>

## Import

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing and managing imports	Only a Function Admin user type or higher can view and manage Standard imports.	Data Creator, Data Manager, and Application Admin user roles can view and edit imports that are specific to a user.  Data Creator, Data Manager, and Application Admin user roles can view Standard imports but they can be managed only by a Data Manager user role.  Only an Application Admin user role can view and manage System imports.
Defining an import source	You can use imports to load data from a client file.	You can use imports to load data from a client or server file.  Server file (NFS and FTP) imports use common external Connections which centrally maintain connectivity and authentication information.
Saving and printing results	Import results are written to a client log file.	You can download import results to a CSV, XLS, RTF, or PDF file to be saved or printed.

## Blend

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing and managing blenders	Only a Function Admin user type or higher can view and manage Standard blenders.	Data Creator, Data Manager, and Application Admin user roles can view and edit blenders that are specific to a user.  Data Creator, Data Manager, and Application Admin user roles can view Standard blenders, but only a Data Manager user role can manage them.  Only an Application Admin user role can view and manage System blenders.
Defining blender details	Blenders have a source and target version, and a third version can be created.  You individually select processing options for each blender.	Blenders have From/To source versions and a target version that can be the To source version, or a third version can be created.  Processing options specific to structural changes or properties are grouped and defaulted for convenience purposes.
Saving and printing results	Blender results are written to a client log file.	You can download blender results to a CSV, XLS, RTF, or PDF file to be saved or printed.

## Export

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Viewing and managing exports	You can view and edit exports that are specific to a user.  Any user can view Standard exports, but only a Functional Admin user type or higher can manage them.	You can view and edit exports that are specific to a user.  Any user can view Standard exports, but only a Data Manager user role can manage them.  Only an Application Admin user role can view and manage System exports.



Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Defining an export target	Exports can output data to a client file or a database table. Database table exports require connection and authentication information to be defined for each export.	Exports can output data to a client file, server file, or database table. Server file (NFS and FTP) and database table exports use common external connections, which centrally maintain connectivity and authentication information.
Saving and viewing export results	Export results are directly written to a client file or can be viewed using the Preview output mode. You can preview export results with an internal or external viewer controlled by settings from the Tools-Options menu.	Export results can be saved to a file, downloaded via the browser, or written to a database table.

## Audit

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Querying for transactions	All transactions can be queried by any user.	Users can query any transactions they have performed. Only a Version/Hierarchy owner or Data Manager user role can query data transactions for versions and hierarchies. Only an Access Manager or Application Admin user role can query System transactions.
Viewing admin transaction detail	Detail for admin transactions is stored as XML and can be viewed in an explorer mode or XML viewer. You can view the From and To values separately.	You can view detail for admin transactions as two side-by-side trees for convenience purposes.
Querying and deleting requests	All requests can be queried by any user or can be restricted to admin users by system preferences. All requests can be deleted by any user or can be restricted to admin users by system preferences.	Users can query any requests they own. All requests can be queried and deleted only by the Data Manager user role.

## Administer

Action	Action Before Release 11.1.2	Action for Release 11.1.2 and Later
Grouping hierarchies	System Admin users can create/delete System Categories from the Admin menu. You can assign hierarchies to System Categories from the Admin menu or individually using hierarchy-level properties in the Version Control panel.	Data Manager users can manage Hierarchy Groups from the Administer task group. You can assign hierarchies to Hierarchy Groups directly from the Browse task group.
Creating property formulas	You can create formulas for derived properties in the Formula Editor. You can select functions and properties from a list to insert into the formula. Function parameters can be entered only into the formula string itself.	You can create formulas for derived properties on the Parameters tab of a property definition page. You can select functions and properties to insert into the formula. Function parameters can be entered at the time of insertion into a formula.

<b>Action</b>	<b>Action Before Release 11.1.2</b>	<b>Action for Release 11.1.2 and Later</b>
Creating batch validations	Validations and Verifications can be created to only run as either real-time or batch.	<p>Each validation can be created to be run as real-time and/or batch and the mode of operation is chosen during assignment.</p> <p>A default mode is defined to support a Boolean assignment value.</p>
Assigning user permissions	<p>Users can be assigned a single User Type that controls permissions to access features within the product.</p> <p>Security and System Admin user types have the ability to create users and assign user types.</p>	<p>Users can be assigned one or more User Roles that control permissions to access features within the product.</p> <p>The Access Manager user role only has the ability to create users and assign user roles.</p>

# 3

## Managing Versions

### In This Chapter

Data Relationship Management Version Life Cycle .....	27
Version Type and Status.....	29
Creating Versions .....	29
Copying Versions.....	29
Saving Versions .....	30
Creating As-Of Versions .....	30
Editing Version Properties .....	31
Assigning Validations to Versions .....	31
Unloading and Deleting Versions.....	32

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. Nodes cannot be copied or moved across versions and their properties cannot refer to nodes of other versions. Compare, blend, and export are the only features that work across versions.

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 within each version is categorized into hierarchies, nodes, and properties. All data-related operations require selection of a version.

### 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**, select **Versions**, then **New**.
- 2 Enter a name and description for the version and 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.

## 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. Examples include the version name, description, and status.

► 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.

## 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 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.



# 4

## Managing Hierarchies

### In This Chapter

Creating Hierarchies .....	33
Editing Hierarchy Properties.....	34
Opening Hierarchies .....	36
Creating Hierarchy Groups .....	36
Assigning Hierarchies to Hierarchy Groups .....	37
Assigning Controlled Properties.....	37
Assigning Controlled Node Access Groups .....	37
Assigning Validations .....	38
Deleting Hierarchies .....	38

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 Select **Hierarchies**, then **New**, then **Hierarchy**.
- 3 Enter a name and description.
- 4 **Optional:** Select a hierarchy group.
- 5 **Optional:** Select a top node for the hierarchy and provide a description.

## 6 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 1** Hierarchy Properties

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

Hierarchy Property	Definition
User Display By	Determines how hierarchy nodes display for the current user
User Hierarchy Sort	Points to a node-level property to determine sort order within the hierarchy for the current user
User Match By	Property by which to synchronize nodes in the selected hierarchy for the current user
User Paste Properties	<p>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.</p> <p><b>Note:</b> If this property and Default Paste Properties are both set, the user property takes precedence over the default property.</p>

► To edit hierarchy properties:

1 From **Browse**, select a version.

2 Select **Hierarchies**, then **Properties**.

Hierarchy 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 3 and 4 as needed to edit hierarchy properties.

6 Click **Save** in the property grid.

## Enabling and Disabling Node Types

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

► 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 Select a hierarchy or hierarchy group.

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

- 3 Select **Hierarchies**, then **Open**.

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

- 4 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

## Creating Hierarchy Groups

Hierarchy groups allow related hierarchies to be organized together for easy browsing and selection. A hierarchy can only exist in a single hierarchy group in a version.

► 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 Use the arrows to select hierarchies from the **Available** list 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 Select a hierarchy.
- 3 Select **Hierarchies**, then **Assign to Group**.
- 4 Use the arrows to select hierarchies from the Available list 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 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 Select a hierarchy that will control a global inheriting property.
- 3 Select **Hierarchies**, then **Assign Control**, then **Properties**.  
A list of global inheriting properties are displayed.
- 4 From the **Available** list, select properties by moving them to the **Selected** list and 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 Select a hierarchy.
- 3 Select **Hierarchies**, then **Assign Control**, then **Node Access Groups**.

A list of global inheriting access groups are displayed.

- 4 From the **Available** list, select groups by moving them to the **Selected** list and 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 Select a hierarchy.
- 3 Select **Hierarchies**, then **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 Select a hierarchy.
- 3 Select **Hierarchies**, then **Delete**.

- 4 Click **OK** to confirm the deletion.

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





# 5

## Working with Nodes

### In This Chapter

About Nodes.....	41
Adding and Inserting Nodes .....	42
Modeling a Node.....	44
Moving and Ordering Nodes .....	45
Node Deletion Methods .....	47
Assigning Validations .....	50
Orphan Nodes.....	50
Navigating in Tree View.....	51
Finding Nodes.....	52
Node Security .....	53
Using Shared Nodes.....	53

## About Nodes

A node is a point within a hierarchy. Every point in a hierarchy is a node. For example, within a hierarchy that represents an organizational structure, a node might represent a department or a cost center.

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 2** Node Terminology

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

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

## 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 Select a node.
- 5 From **Nodes**, select **New**.
- 6 **Optional:** Select **Leaf Node** if the node is a leaf. Otherwise, it is created as a limb node.
- 7 Enter a unique name and a description for the node.

**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 Select whether to add the node as a child or sibling of the selected node.
- 9 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 Select a node.
- 5 From **Nodes**, select **Insert**.
- 6 Select the location of the node:

- Existing Hierarchy
- Orphan Node List

7 Do one of the following:

- If you selected **Existing Hierarchy** in step 6, do the following:
  - a. Select a hierarchy.
  - b. Select the node to insert by doing one of the following:
    - Enter text in **Find** and click  to locate the node.

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

Click  to find the next instance of the search item.

- Expand the Nodes tree to locate the node.
- 8 Select whether to insert the node as a child or sibling of the selected node.
  - 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.

## 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 3** Node Deletion Methods

Action	Result
Destroy	Destroy deletes a node from the version and overrides any required merge logic if enabled. <b>Note:</b> The Destroy option is only available from the Orphans list.
Delete	Deletes the node from all parents in all hierarchies.
Delete & Merge	Deletes the node from all parents in all hierarchies and allows you to select a node to use in place of the deleted node. <b>Note:</b> This option is only available if the UseMerge system preference is enabled.
Annul	Annul removes the node from all hierarchies where it has the same parent, and also deletes the node if it is then an orphan. <ul style="list-style-type: none"> <li>● 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.</li> <li>● If the node does not have the same parent in all hierarchies, the Annul produces the same result as Remove.</li> </ul>
Annul All Below	Removes all the children of the selected node. If any of the children become orphans, they are also deleted.
Remove	Removes the node from the current parent but not from other parents in other hierarchies. The node is not deleted from the version and is available for re-insertion.
Inactivate	Flags the node as inactive so that it can be filtered from an export.
Inactivate & Merge	Flags the node as inactive and allows you to select a node to use in place of the inactivated node. <b>Note:</b> This option is only available if the UseMerge system preference is enabled.

## 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

► 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. If ReqMerge is not enabled, then a merge node is optional unless the node approved property is TRUE. The node approved property is set to TRUE when a version is finalized or when it is specifically set to TRUE by a user with appropriate access.

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

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

## Removing Nodes

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

► To remove a node:

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

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

- 4 Select a node.
- 5 From **Nodes**, select **Remove**.
- 6 Select **Remove Node** to confirm node removal.

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



## Annuling Nodes

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

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

► To annul a node:

1 From **Browse**, select a version.

2 Select a hierarchy.

3 From **Hierarchies**, select **Open**.

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

4 Select a node.

5 From **Nodes**, select an action:

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

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

6 Select **Annul Node**.

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

## Inactivating Nodes

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

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

► To inactivate a node:

1 From **Browse**, select a version.

2 Select a hierarchy.

3 From **Hierarchies**, select **Open**.

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

4 Select a node.

5 From **Nodes**, select **Inactivate**.

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

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 re-assignment to hierarchies in the version.

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

## Viewing Orphan Nodes


► To view orphan nodes:

- 1 **Select the Orphans tab.**

For each orphan node, the number of descendants is displayed.

- 2 **Select a node.**

- 3 **Perform an action:**

- Select **Related** and then **Descendants** or **Children** to view a node's relationships.
- Select **Node** and then **Properties** to view a node's properties.
- Click  to search for an orphan node by name, description, or number of descendants.
- Select **Node** and then **Take/Copy** to copy the selected node to the clipboard.
- Select **Download** and then choose a format in which to download the list of orphan nodes.

## Deleting Orphan Nodes

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

► To delete orphan nodes:

- 1 **Select the Orphans tab.**

- 2 **Select one or more nodes.**

- 3 **From Node, select Delete or Destroy.**




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

- 4 **Confirm the Delete or Destroy command.**

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

## Navigating in Tree View

Use the following toolbar buttons to navigate in tree view.

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

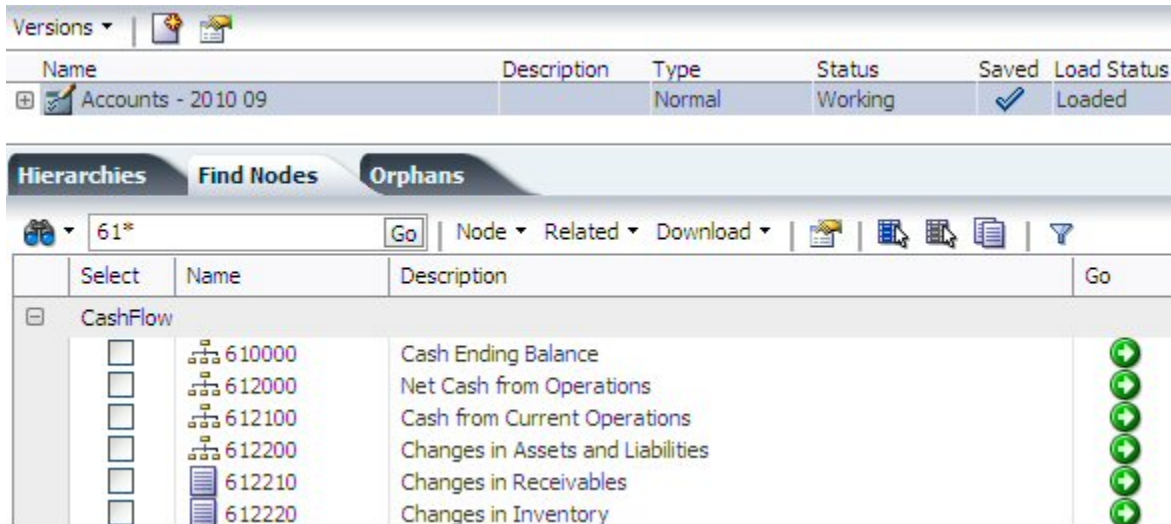
## Finding Nodes

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.



Name	Description	Type	Status	Saved	Load Status
Accounts - 2010 09		Normal	Working	✓	Loaded



Select	Name	Description	Go
<input type="checkbox"/>	CashFlow		
<input type="checkbox"/>	6 10000	Cash Ending Balance	✓
<input type="checkbox"/>	6 12000	Net Cash from Operations	✓
<input type="checkbox"/>	6 12100	Cash from Current Operations	✓
<input type="checkbox"/>	6 12200	Changes in Assets and Liabilities	✓
<input type="checkbox"/>	6 12210	Changes in Receivables	✓
<input type="checkbox"/>	6 12220	Changes in Inventory	✓

3 Click  and select how you want to search:

- **By Name**
- **By Description**

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.

- To find a node in a hierarchy:
  - 1 From **Browse**, select a version.
  - 2 Select a hierarchy, then click .
  - 3 Click  and select how you want to search and return results:
    - **Find By – Name** or **Description**
    - **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 If you selected to show results in the hierarchy tree, you can move to the next result by clicking .

## 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 Hyperion 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.

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

You can:

- Automate the alternate naming of the shared nodes
- Create system-level properties that map shared nodes

- Display the structure below a shared node
- Move between shared nodes and the primary node
- Remove and delete primary shared nodes
- Rename the primary shared node (thus renaming all the shared nodes)
- Find and display a shared node list to be used for navigation and analysis purposes

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

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

## Shared Node Properties

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

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

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

## Enabling Shared Nodes

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

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

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

## Adding and Inserting Shared Nodes

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

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

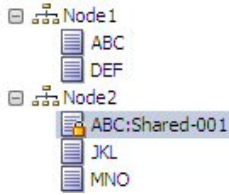
➤ To add a shared node:

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

➤ To insert a shared node:

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

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



## Removing and Deleting Shared Nodes

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

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

► To remove or delete a shared node:

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

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

## Renaming Shared Nodes

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

► To rename a shared node:

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

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

## Converting a Shared Node to a Primary Node

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.



# 6

# Managing Properties

## In This Chapter

Property Categories .....	59
Property Value Scope and Origin .....	61
Property Status .....	62
Editing Property Values .....	63

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, Planning or 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 4** Default Property Categories

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

Category	Description
Leaf Access	Node security groups and their leaf access levels for the node – one property for each group This category is available for nodes only.
Limb Access	Node security groups and their limb access levels for the node – one property for each group This category is available for nodes only.

You can have different levels of access to different property categories. Categories that can be viewed are available in the drop-down list. Categories that can be edited allow modification of properties that support overridden values. The Validation, Leaf Access, and Limb Access categories are available only to users with the Data Manager role.

## System Properties

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

**Table 5** System Properties

Property	Definition
Description	Description of node
Display By String	Value of combined properties used for viewing nodes in a hierarchy. Controlled by the Default Display By and User Display By properties.
Leaf	<ul style="list-style-type: none"> <li>● If True, node is a leaf node and cannot have children</li> <li>● If False, node is a limb node and may have children</li> </ul>
Leaf Access	Indicates level of access to node for current user (only applies if node is leaf)
Limb Access	Indicates level of access to node for current user (only applies if node is limb)
Name	Unique name of node within a version
Node ID	Unique numeric identifier for the node generated by the system

## Stats Properties

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

**Table 6** Stats Properties





Property	Definition
# Children	Number of nodes, one level beneath
# Descendants	Total number of nodes beneath (at all levels)
# Leaf Children	Number of child nodes that cannot have children

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







## Property Value Scope and Origin

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

**Table 7** Property Scope Icons

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

**Table 8** Property Origin Values

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

## 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 9** Property Status

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

## Editing Property Values

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

- The property has a derived value. For example, a property indicating the total number of child nodes is derived by 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, choose an action:**

- **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.
- 4 Take an action:
    - Click **Save** to save your changes.
    - Click **Cancel** to exit without saving changes.





# 7

## Validating Data

### In This Chapter

Validating Versions.....	65
Validating Hierarchies .....	66
Validating Nodes.....	66
Navigating Validation Results.....	67

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.

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, see the *Oracle Hyperion 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 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 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.

# Navigating 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 Node** — 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:

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

## Downloading Node Results

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

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

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

► To download query results in list view:

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

## Clearing Validation Results

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

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

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

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

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

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



# Using Queries

---

## In This Chapter

Query Types.....	69
Creating Queries .....	70
Navigating Query Results.....	71
Managing Queries.....	73

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, 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 From the Data Relationship Management navigation bar, select **Query**.

2 Click .

3 Select the **Source** tab and choose a version 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.

**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 From the Available list, select the properties to display as columns in the query results by moving them to the Selected list.**

**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 either or both of the following:**

- Click  to save the query.

You are prompted to define a name, description, and object access for the query:

- **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 queries.
- **System** – Restricted objects that are only available to Application Administrator role users to view and run.

- Click  to run the query.

## Navigating Query Results

Depending on the display options selected for the query, the results are displayed in a list, marked in the tree, or both. Global query results are read only but you can make edits directly in the

query results. Property edits can be performed for nodes displayed in a list of results for a local query. Structural changes and property edits are allowed for local query results displayed as marked nodes in the hierarchy tree.

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 Node — 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:

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

## Downloading Query Results

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



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

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

➤ To download query results in tree view:

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

## Managing Queries



You can open, copy, and delete saved queries. For more information see:

- [Opening Queries](#)
- [Copying 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 Level**, select a level:
  - **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 queries.
- **System** – Restricted objects that are only available to Application Administrator role users to view and run.

5 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.

# 9

## Comparing Hierarchies

### In This Chapter

Compare Types .....	75
Creating Compares .....	76
Navigating Compare Results .....	78
Managing Compares.....	79

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 From the Data Relationship Management navigation bar, select **Compare**.

2 Click .

3 Click **Source** and select the version 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

**6** 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.

**7** Select the **Filter** tab and do one of the following:

- From **Include 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.

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

**8** From **Limb/Leaf Filter**, select nodes to include.

**9** Select the **Columns** tab and choose a **Category**.


**10** From the **Available** list, select the properties to display as columns in the compare results by moving them to the **Selected** list.

**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.

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

## 12 Do either or both of the following:

- Click  to save the compare.

You are prompted to define a name, description, and object access for the compare:

- **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.

## Navigating 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 Node — 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.

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

## Downloading 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](#)
- [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 Select the **Compare** tab and choose a compare.
  - 2 Click  to open the selected compare.
  - 3 Select the tabs to modify the compare parameters, and then click  to run the compare.

## Copying Compares

- To copy a compare:
- 1 Select the **Compare** tab and choose a compare to copy.
  - 2 From **Compares**, select **Copy**.
  - 3 Enter a new name and description.
  - 4 From **Object Access Level**, select a level:
    - **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**.

## Deleting Compares

- To delete a compare:
- 1 Select the **Compare** tab and choose 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.



---

## In This Chapter

Loading Action Scripts From Flat Files .....	81
Loading Action Scripts From a Transaction Log .....	86
Loading Action Scripts From a Node Model .....	87

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.

---

## 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 From the Data Relationship Management navigation bar, 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** – UTF8, ANSI, ASCII, UTF16BE, UTF16LE
- **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**.

## Required Action Script Parameters

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

**Table 10** Action Script Parameters

Column A	Column B	Column C	Column D	Column E	Column F
<b>Activate</b> Activates the specified node in the hierarchy	Version	Hierarchy	Node		
<b>Add</b> Adds the specified node to the specified hierarchy (This action can only be used for a node that does not already exist within the specified version)	Version	Hierarchy	Node	Parent node	Leaf property
<b>AddHier</b> Adds a hierarchy to a version	Version	Hierarchy	Description	Top Node	
<b>AddInsert</b> Adds the specified node if it does not already exist in the specified version, otherwise the node is inserted	Version	Hierarchy	Node	Parent node	(Leaf property)
<b>AddOrphan</b> Adds an orphan node to a version	Version	Node	Leaf Property		
<b>Annul</b> Annuls the specified node	Version	Hierarchy	Node		
<b>AnnulAllBelow</b> Annuls all nodes following the specified node	Version	Hierarchy	Node		
<b>AssignHierValidations</b> Assigns validations at the hierarchy level	Version	Hierarchy	Validation		
<b>AssignVersionValidations</b> Assigns validations at the version level	Version	Validation			

Column A	Column B	Column C	Column D	Column E	Column F
<b>ChangeHierTopNode</b> Changes the top node for a specified hierarchy	Version	Hierarchy	New Top Node		
<b>ChangeProp</b> Updates the value of the specified property of the specified node	Version	Hierarchy	Node	Property label or name  Controlled by the Property References By Label option.	Value
<b>ClearPropsBelow</b> Clears the values of the specified property in all of specified node's children (Corresponds to the <i>Clear All Below</i> command that is available to users within Data Relationship Management)	Version	Hierarchy	Node	Property label	
<b>Delete</b> Deletes the specified node from the version, and merges it with the merge node if one is specified	Version	Hierarchy	Node	(Merge node)	
<b>DeleteHier</b> Deletes a hierarchy from a version	Version	Hierarchy			
<b>DeleteOrphan</b> Deletes an orphan node	Version	Node			
<b>Destroy</b> Destroys the specified node from the version	Version	Hierarchy	Node		
<b>Inactivate</b> Inactivates the specified node in the hierarchy, and merges it with the merge node if one is specified	Version	Hierarchy	Node	(Merge node)	
<b>Insert</b> Inserts the specified node into the specified hierarchy (This action can only be used for nodes that already exist in another hierarchy within the specified version)	Version	Hierarchy	Node	Parent node	
<b>LockProp</b> Locks or unlocks the specified property of the specified node	Version	Hierarchy	Node	Property label	Is Locked (0=false, 1=true)
<b>Move</b> Moves the specified node within the hierarchy	Version	Hierarchy	Node	Destination parent node	
<b>NodeExists</b> Returns True if the specified node exists in the specified version, otherwise returns False	Version	Node			

Column A	Column B	Column C	Column D	Column E	Column F
<b>NodeExistsInHier</b> Returns True if the specified node exists in the specified hierarchy, otherwise returns False	Version	Hierarchy	Node		
<b>NodeHasAsParent</b> Returns True if the specified parent node is the parent of the specified node, otherwise returns False	Version	Hierarchy	Node	Parent node	
<b>PropCompare</b> Returns True if the value of the specified property of the specified node equals the value listed in the flat file, otherwise returns False	Version	Hierarchy	Node	Property label	Value
<b>PropQuery</b> Returns the value of the specified property of the specified node in the Action Script result set	Version	Hierarchy	Node	Property label	
<b>Rem</b> Indicates a remark/comment row in the flat file (Any arbitrary text can be entered on other columns of this row for documentation purposes—it is ignored by the Action Script process)	Remark	Remark	Remark	Remark	Remark
<b>Remove</b> Removes the specified node from the hierarchy	Version	Hierarchy	Node		
<b>RemoveDupProps</b> Removes (clears) duplicate properties for the specified node (Corresponds to the <i>Remove Duplicates</i> command that is available to users within Data Relationship Management)	Version	Hierarchy	Node	Property label	
<b>RemoveHierPropValue</b> Removes (clears) the value of a hierarchy property	Version	Hierarchy	Property label		
<b>RemoveProp</b> Removes (clears) the value of the specified property of the specified node (The property itself is not deleted because many other nodes may be utilizing this property)	Version	Hierarchy	Node	Property label	
<b>RemoveVersionPropValue</b> Removes (clears) the value of a version property.	Version	Property label			
<b>UpdateHierPropValue</b> Updates the value of a hierarchy property	Version	Hierarchy	Property label	Value	
<b>UpdateVersionPropValue</b> Updates the value of a version property	Version	Property label	Value		

# 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 From the Data Relationship Management navigation bar, 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 From the Data Relationship Management navigation bar, 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.



# 11

## Working with Imports

### In This Chapter

Import Sections .....	89
Creating Imports .....	90
Running, Copying, and Deleting Imports.....	93

Using Data Relationship Management, you can import data extracted from external systems 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 can contain any or all of the following sections:

**Table 11** Import Sections

Import Section	Description
Version	Version property values <i>Required column: Version Name</i>
Hierarchy	Structural information about each hierarchy and top node to be imported and, perhaps, hierarchy property values. <i>Required columns: Hier Name, Top Node</i>

Import Section	Description
Node	Structural information about each node to be imported and, perhaps, global node property values <i>Required column: Node</i>
Relationship	Parent-child information about each relationship to be imported and, perhaps, global or local node property values for the child in each pair. <i>Required columns: Parent Node Name, Node</i>
Hierarchy Node	Local node property values specific to a specified hierarchy <i>Required columns: HierName, Node</i>

## Creating Imports

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

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 From the Data Relationship Management navigation bar, 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.

**4 Select File Format options:**

- Character Encoding — UTF8, ANSI, ASCII, UTF16BE, UTF16LE.
- 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** — Select 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 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 node 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).

**8** On the **Columns** tab, from the **Section** list, select a file section.

**9** From the **Available** list on the **Select Columns** tab, select the columns for the import.

**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:** 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.

**10** On the **Column Options** tab, you can map a column to hierarchies. Double-click a column name row and select from these options:


- Select **Apply Value to All Hiers** if values in the column should be applied to all hierarchies. By default, this option is selected.
- In the Hierarchy Listing field, enter the name of each hierarchy to which the column applies. The property value is set only in the hierarchies designated in the list.

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


- If you selected Fixed Width on the Source tab, enter a value in the Fixed Width column.

**11** On the **Filter** tab, make selections to skip blank and use defaults for certain property types.

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

- 12 On the **Target** tab, enter a name and, optionally, a description for the new version that will be created by the import.
- 13 Select a value for the maximum number of errors that can occur during processing before the import stops.
- 14 Do either or both of the following:
  - Click  to save the import.



You are prompted to define a name, description, and object access for the import:

    - **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 imports.
    - **System** – Restricted objects that are only available to Application Administrator role users to view and run.
  - Click  to run the import.

## Running, Copying, and Deleting Imports

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

► To run an import:

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

► To copy an import:

- 1 On the **Import** tab, an import.
- 2 From **Imports**, select **Copy**.
- 3 Enter a new name and description for the import.
- 4 From **Object Access Level**, select a level:
  - **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 imports.
  - **System** – Restricted objects that are only available to Application Administrator role users to view and run.
- 5 Click **OK**.

- ▶ To delete an import:
  - 1 On the **Import** tab, select an import.
  - 2 From **Imports**, select **Delete**.

---

## In This Chapter

Creating Blenders .....	95
Running, Copying, and Deleting Blenders .....	99

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 From the Data Relationship Management navigation bar, select **Blend**.
- 2 Click .
- 3 On the **Source** tab, select source and target versions 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**, click **New** and select hierarchy and top node mappings to select hierarchy/top node pairs to blend.

Hierarchy/top node pairs are mappings from the source to the target indicating which hierarchy/top nodes in the target are to receive values from the source. Top nodes in the target can be created or mapped to existing top nodes, depending on the Blender settings.

**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.

**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

## 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 Hyperion 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.
- **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 Do either or both of the following:**

- Click  to save the blender.

You are prompted to define a name, description, and object access for the blender:



- **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.

- Click  to run the blender.

## Running, Copying, and Deleting Blenders

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

► To run a blender:

- 1 From the Data Relationship Management navigation bar, select **Blend**.
- 2 Select a blender and click .
- 3 Click the tabs to modify the blender parameters, and then click  to run the blender.

► To copy a blender:

- 1 From the Data Relationship Management navigation bar, select **Blend**.
- 2 Select a blender.
- 3 From **Blenders**, select **Copy**.
- 4 Enter a new name and description for the blender.
- 5 From **Object Access Level**, select a level:
  - **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.
- 6 Click **OK**.

- ▶ To delete a blender:
  - 1 From the Data Relationship Management navigation bar, select **Blend**.
  - 2 Select a blender.
  - 3 From **Blenders**, select **Delete**.

**In This Chapter**

Creating Exports.....	101
Creating Export Books.....	130
Running, Copying, and Deleting Exports and Export Books.....	131

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.

Export Type	Description	Usage Frequency	Output Formats
Hierarchy Export	Outputs a record for each node and its properties for hierarchies. This export can be used to create a parent-child type format or a basic list of hierarchy nodes.	High	File, Database Table
Hierarchy Export XML	Outputs nodes, their relationships, and properties from hierarchies in an XML format.	High	File
Generation Export	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.	Medium	File
Compare Export	Compares two hierarchies and outputs a record for each node that met the comparison criteria. This export can be used to provide incremental changes to target systems.	Medium	File, Database Table
Difference Export	Compares two hierarchies and outputs a record for each difference found. This export can be used for reporting incremental changes to users.	Low	File
Version Export	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.	Medium	File, Database Table

Export Type	Description	Usage Frequency	Output Formats
Property List Export	Outputs a record for each item in the list of values for a property definition. This is a special use case export type.	Low	File
Property Lookup Export	Outputs a record for each key-value pair in the lookup table for a property definition. This is a special use case export type.	Low	File
Transaction Log Export	Outputs a record for each item in the transaction log that met the filter criteria. This export can be used for audit purposes.	Medium	File, Database Table
Merge Export	Outputs a record for each item in the merge log that met the filter criteria. This export is only applicable if the Merge feature is used.	Low	File

## Creating Hierarchy Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a hierarchy export:

1 From the Data Relationship Management navigation bar, select **Export**.

2 Click .

3 Select **Hierarchy** as the export type.

4 Select the **Source** tab and select a version.

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 Select the **Style** tab and choose 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 Repeat Parameters section, choose 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 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).

**8 Select the Filter tab and 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.

**9 Select the Columns tab and from the Available list, select the properties to display as columns in the export results by moving them to the Selected list.**

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

**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.

#### 10 Select the **Column Options** tab, and choose 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).

#### 11 Select the **Target** tab and choose where to export the results:

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

#### 12 Perform an action:

- If you selected to export to a **Client File**, choose options:
  - Column Headings — Includes column headings in the export 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE.
  - 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 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - Replace options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
  - Header/Footer — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

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


- Delimiter options — To select a field delimiter character and a record delimiter character

**13 Do either or both of the following:**

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## 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 12** Customization Tags

Customization Tag	Description
<%Date[(format)]%>	Returns current date based on the user's regional settings, where format = MM/DD/YYYY
<%Time[(format)]%>	Returns current time based on the user's regional settings, where format = HH:MM:SS AMPM
<%Timestamp%>	Returns combination of date and time.
<%CurrentUser%>	Returns current Data Relationship Management user.
<%CurrentVersion%>	Returns current Data Relationship Management version for this export.
<%PreviousVersion%>	Returns previous Data Relationship Management version for this export.
<%ExportName%>	Returns name of export.
<%FileName%>	<ul style="list-style-type: none"> <li>● Returns output file name if specified</li> <li>● Returns default file name if not specified</li> </ul>
<%RecordCount%>	Returns count of records exported (only available in footer)
<%AppVersion%>	Returns the Data Relationship Management product number (for example, 11.1.2)
<%PropValue()%>	Returned value depends on the specified export type as described following:

Customization Tag	Description
<%PropValue([Previous   Current(Default)], PropName)%>	For Compare/Difference Exports, returns value of first node in CompareCriteria for Previous/Current version
<%PropValue([HierName], PropName)%>	<ul style="list-style-type: none"> <li>● For Hierarchy Exports, returns value from first node in list</li> <li>● If HierName is supplied, return value from the first node that matches HierName</li> </ul>
<%PropValue(PropName)%>	For Version Exports, returns only version properties

## Creating Hierarchy XML Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a hierarchy XML export:

**1 From the Data Relationship Management navigation bar, select **Export**.**

**2 Click** .

**3 Select **Hierarchy XML** as the export type.**

**4 On the **Source** tab, select a version.**

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 Select the **Style** tab, and choose 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 Select the **Filter** tab and 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.

- 8 Select the **Columns** tab and from the **Available** list, select the properties to display as columns in the export results by moving them to the **Selected** list.

**Note:** Use the Category list to select a property category to make navigation of the Available list easier.

**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.


- 9 Select the **Target** tab and choose where to export 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**, 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.

- 11 Do either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.
- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Generation Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a generation export:

1 From the Data Relationship Management navigation bar, select **Export**.

2 Click .

3 Select **Generation** as the export type.

4 On the **Source** tab, select a version.

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 From **Repeat Parameters**, 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** — determine 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 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).

## 8 From Ancestor Options section, select from these options:

- **Max Ancestor Levels** — Defines the maximum 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 property in the ancestor columns.

## 9 Select the Filter tab and choose 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/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, from the Column Set list, select one of the following options:

- **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

## 11 From the Available list, select the properties to display as columns in the export results by moving them to the Selected list.

**Note:** Use the Category list to select a property category to make navigation of the Available list easier.

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

## 12 Select the **Column Options** tab and choose 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.

## 13 Select the **Target** tab and choose where to export the results:


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

## 14 Perform an action:

- If you selected to export to a **Client File**, choose options:
  - Column Headings — Includes column headings in the export
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
- Replace options — Select up to three characters from the lists to be replaced. For example, commas can be replaced by tabs
- Header/Footer — Add a line between the header/footer and the body or enter text to be printed on the header and footer of the export results. Enter lines of text into the text box or click  to open a text editor.

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


- Delimiter options — To select a field delimiter character and a record delimiter character

**15 Do either or both of the following:**

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Compare Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a compare export:

**1 From the Data Relationship Management navigation bar, select **Export**.**

**2 Click** .



- 3 Select **Compare** as the export type.
- 4 Select the **Source** tab and 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 Select the **Style** tab and select from these 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 Select the **Filter** tab and choose options:

- From **Validation**, select a validation to apply to the export.
- From **Query**, select a query to apply to the export or, to define a query, click .

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

7 On the **Columns** tab, from the **Available** list, select the properties to display as columns in the export results by moving them to the **Selected** list.

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

**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.

**8 Select the **Column Options** tab and choose 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, from the **Device** list, select where to export the results: **Client File, Database Table, or Server File.****

**10 Perform an action:**

- If you selected to export to a **Client File**, choose options:
  - Column Headings — Includes column headings in the export
  - Quoted Strings — Puts quotation marks around 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.
- Click  to run the export.


If you selected Client File, you have a download option where you can download the file locally.

## Creating Difference Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a difference export:

- 1 From the Data Relationship Management navigation bar, 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 Select the **Filter** tab and choose options:**

- From **Validation**, select a validation to apply to the export.
- From **Query**, select a query to apply to the export or, to define a query, click .

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

**7 On the **Columns** tab, from the **Available** list, select the properties to display as columns in the export results by moving them to the **Selected** list.**

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

**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.


**8 Select the **Column Options** tab and choose 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, from the **Device** list, select where to export the results: **Client File** or **Server File**.**

**10 Perform an action:**

- If you selected to export to a **Client File**, choose options:
  - Column Headings — Includes column headings in the export.
  - Quoted Strings — Puts quotation marks around 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE.

- 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Version Exports

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

The export wizard is used to define the parameters and criteria for the export. 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 a version export:

1 From the Data Relationship Management navigation bar, select **Export**.

2 Click .

3 Select **Version** as the export type.


4 Select the **Source** tab and choose a version.

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 Select the **Style** tab and choose 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 Select the **Filter** tab and choose 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.

- 7 On the **Columns** tab, from the **Available** list, select the properties to display as columns in the export results by moving them to the **Selected** list.

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

**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.

- 8 On the **Target** tab, from the **Device** list, select where 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.
- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Property List Exports

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

The export wizard is used to define the parameters and criteria for the export. The wizard content is organized across Source 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 property list export:

**1 From the Data Relationship Management navigation bar, select **Export**.**

**2 Click** .

**3 Select **Property List** as the export type.**

**4 Select the **Source** tab and choose a version.**

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 the **Available** list, select a list property upon which to base the export.**


**6 Select the **Target** tab and choose where to export 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**, choose 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 either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.
- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Property Lookup Exports

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

The export wizard is used to define the parameters and criteria for the export. The wizard content is organized across Source, Style, 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 a property lookup export:

1 From the Data Relationship Management navigation bar, select **Export**.

2 Click .

3 Select **Property Lookup** as the export type.

4 Select the **Source** tab and choose a version.

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 the **Available** list, select a lookup property upon which to base the export.

6 **Optional:** Select the **Style** tab, and choose **Differences Only** to filter out lookup pairs where the Lookup Key and Result Value are the same.

7 Select the **Columns** tab and choose 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 arrows to position the column in the Selected list. The first item in the list is the first column, the second item in the list is the second column, and so on.

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

9 Perform an action:

- If you selected to export to a **Client File**, choose options:
  - Column Headings — Includes column headings in the export
  - Quoted Strings — Puts quotation marks around 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 either or both of the following:**

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Transaction Log Exports

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

The export wizard is used to define the parameters and criteria for the export. The wizard content is organized across Source, 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 a transaction log export:

**1 From the Data Relationship Management navigation bar, select **Export**.**


**2 Click** .

**3 Select **Transaction Log** as the export type.**


**4 Select the **Source** tab and choose a version.**

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 To select hierarchies and nodes for the export, click** .

**6 Select the **Filter** tab and choose 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 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE

- 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 either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.

If you selected Client File, you have a download option where you can download the file locally.

## Creating Merge Log Exports

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

The export wizard is used to define the parameters and criteria for the export. The wizard content is organized across Source, 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 a merge log export:

1 From the Data Relationship Management navigation bar, select **Export**.

2 Click .

3 Select **Merge Log** as the export type.

4 On the **Source** tab, select a version.

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 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
  - 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 — To export the results as UTF8, ANSI, ASCII, UTF16BE, UTF16LE
- 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

## 9 Do either or both of the following:

- Click  to save the export.

You are prompted to define a name, description, and object access for the export:

- **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.

- Click  to run the export.


If you selected Client File, you have a download option where you can download the file locally.

## 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 From the Data Relationship Management navigation bar, select **Export**.
- 2 Click .
- 3 Select the **Source** tab and choose a version from which to select exports for the book.

4 From **Object Access Level**, select a level for the book:

- **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 exports can only reference standard or system exports.

5 Select exports from the **Available** list and move them to the **Selected** list to include them in the book.

**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, select **Combine Export Output** and specify if the combined file will be a server file or client file. If you are combining into a server file, an external connection must be set up.

Pre File and Post File can be used to specify data that is prepended and appended to the combined output file. External connections must be set up to these files.

**Note:** When a book contains both exports configured to output to a file and some configured to output to a database table, the Combine option affects only the exports that are writing to files.

7 Click  and define a name and description for the book.


## Running, Copying, and Deleting Exports and Export Books

You can run, copy, 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 Do one of the following:

- On the **Export** tab, double-click an export.
- On the **Books** tab, double-click an export book.

2 Click the tabs to modify the export or export book parameters, and then click .

➤ To copy an export or export book:

1 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 Level**, select a level:
    - **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 delete an export or export book:
- 1 Select an export or export book.
  - 2 From **Exports**, select **Delete**.

Change requests are proposed data changes to a Data Relationship Management version. Requests consist of multiple transactions that are validated against and committed to a version as a group. Requests greatly extend the ability to integrate approval change management with Data Relationship Management by allowing transactions within a request to be grouped as a single entity and allow the request to be validated without committing changes to the Data Relationship Management version.

You can create change requests to:

- Add and delete hierarchies
- Add and delete nodes
- Insert and move nodes
- Activate, inactivate and remove nodes
- Update properties
- Remove property values

Change requests can be stored in a draft state for approval and validated against a Data Relationship Management version without committing the changes to the version immediately. Change requests in this pending approval state can be updated by multiple users at different times and re-validated as needed. The transactions in a request are committed to a Data Relationship Management version when the request is approved.

**Note:** After a change request has been approved, the request cannot be modified and the request cannot be deleted until the associated version is deleted.

A change request consists of the following elements:

- Target Data Relationship Management version.
- Owner of the request — A valid Data Relationship Management user ID.
- Custom workflow ID — Identifier for the request in a workflow application.
- Custom workflow label — Short description for the request in a workflow application.
- Custom workflow status — Manages the status of the request in a workflow application.
- Custom workflow info — Stores extra information needed by a workflow application.
- Request comments — Annotation for the request.

- Created by — User who created the initial request.
- Created date — Date when the request was created.
- Updated by — User who last updated the request.
- Updated date — Date when the request was last updated.
- Approved By — User who approved the request.
- Approved Date — Date when the request was approved.
- Validated Flag — Indicates whether the request has been validated since it was last updated.
- Approved Flag — Indicates whether the request has been approved.
- Additional batch validations that should be applied to only the actions in the request during a validate or approve operation
- List of action items that affect hierarchies and nodes for the current request

Change requests are exposed via the Data Relationship Management API for use with the Workflow Development Kit or third-party applications to support request and approval change management processes. For more information on requests, see the *Oracle Hyperion Data Relationship Management Application Programming Interface Guide*.

## In This Chapter

Auditing Transaction History .....	135
Data Actions in Transaction History .....	136
Administrative Actions in Transaction History .....	138
Auditing Requests.....	141

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.

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 and requests using the Audit task. For more information, see:

- [Auditing Transaction History](#)
- [Auditing Request History](#)

## Auditing Transaction History

► To query and view transaction history:

- 1 From the Data Relationship Management navigation bar, select **Audit**.
- 2 From **Audit Type** select **Transactions**.
- 3 On the **Source** tab, select whether 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 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 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:

### Version, Hierarchy, and Node Data Changes

- Add Hierarchy
- Add Node
- Add Orphan Node
- Add Version
- Annul Node



- Annul All Below Node
- Assign Hierarchy Validations
- Assign Version Validations
- Change Hierarchy Top Node
- Clear All Below
- Copy Version
- Delete Hierarchy
- Delete Node
- Delete Orphan
- Delete Version
- Detach Version
- Inactivate Node
- Insert Node
- Lock Node Property
- Move Node
- Node Property Data Loss
- Order Children
- Reactivate Node
- Remove Duplicates
- Remove From Orphan
- Remove Hierarchy Property Value
- Remove Node
- Remove Node Property Value
- Remove Version Property Value
- Save Version
- Unlock Node Property
- Update Hierarchy Property Value
- Update Node Property Value
- Update Version Property Value

#### **Logging Actions**

- Admin Change User Password
- Admin Rename Compare Criteria
- Admin Rename Criteria List
- Admin Update Hierarchy Validations

- Admin Update Property Lookup Value
- Admin update Version Validations
- Migration Export
- Migration Import
- Action Script Run
- BCP Run
- Blender Run
- Close Version
- Compare Run
- Export Run
- Global Query
- Import Run
- Log In
- Log Out
- Open Version
- Property Compare Run
- Query
- Rebuild Hierarchy Node Table
- Rename Compare Run
- Batch Validation Run

## Administrative Actions in Transaction History

The Transaction History records all changes to metadata such as property definitions, system preferences, and export profiles. The following administrative actions are logged in the transaction history:

### Node Access Groups

- Add Node Access Group
- Update Node Access Group
- Update Access Group Members
- Update Access Group Controlling Hiers
- Delete Node Access Group

### Node Types

- Add Node Type
- Update Node Type

- Apply Node Type Props
- Apply Node Type Validations
- Delete Node Type

### **Properties**

- Add Property Definition
- Update Category List for Prop Def
- Update Property Definition
- Update Property Definition Parameters
- Update Controlling Hierarchy
- Update Property List Values
- Add Property Lookup Value
- Delete Property Lookup Value
- Delete Property Definition

### **Property Categories**

- Add Category
- Update Category
- Update Category Users
- Update Category Properties
- Delete Category

### **System Preferences**

- Update System Preference

### **Hierarchy Group**

- Add Hierarchy Group
- Update Hierarchy Group
- Delete Hierarchy Group

### **User Management**

- Add User
- Set User Password
- Update User
- Assign User Node Access Groups
- Assign User Categories
- Delete User

- Update User Lockout

### **Validations**

- Add Validation
- Set Validation Params
- Update Validation
- Delete Validation

### **Import**

- Save Import
- Delete Import

### **Export**

- Save Export
- Delete Export

### **Book**

- Save Book
- Delete Book

### **Blender**

- Save Blender
- Delete Blender

### **Property Query**

- Update Criteria List
- Delete Criteria List

### **Compare**

- Save Compare Criteria
- Update Compare Criteria
- Delete Compare Criteria

### **Glyph**

- Add Glyph
- Delete Glyph
- Update Glyph

## Request

- Approve Request
- Delete Request
- Save Request

## Auditing Requests


All change requests are logged in the Request History. This data can be viewed by selecting Tools, then Request History from the Data Relationship Management main menu. Each request is recorded with timestamps, the user ID of the request owner, the actions being performed, the validation and approval status, and other request level information. When you first access the Request History screen, a default query displays all unapproved requests that you own.

The top section of the Request History screen allows you to define query criteria. The bottom section displays the list of change requests that meet the current query criteria. Query criteria are grouped into four categories:

- Summary
  - Request ID — Cannot be changed; automatically generated when the request is created.
  - Owned By user ID — Cannot be changed; defined when request is created.
  - Committed By user ID — Cannot be changed; defined when request is approved.
  - Target version for the request.
  - Perform exact match on string values option.
  - Maximum number of records to return setting.
- Workflow — Used to define applicable data.
  - Custom ID
  - Custom Label
  - Custom Status
  - Comments
- History
  - Created By, Updated By, and Approved By fields for managing user-related information for the workflow application.
  - Create Date, Update Date, and Approval Dates for when a request is created, updated, and approved.

➤ To query and view request history:

- 1 From the Data Relationship Management navigation bar, 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 select the format for the file : TSV, CSV, PDF, RTF, or XLS.

## In This Chapter

Securing the Batch Client .....	144
Running the Batch Client.....	144
Command Line Syntax.....	145
Configuration File Syntax.....	151
Outputs and Result Codes .....	157

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\EPMSysstem11R1\products\DataRelationshipManagement\client\batch-client\.`

By passing command-line parameters appropriate to the operation, the Batch Client is able to log into 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 32 on page 157](#).

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.

## 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\EPMSysstem11R1\products\DataRelationshipManagement\server\bin`.**
- 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.

## 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 Hyperion 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 13** All Operations

Parameter	Description
/op=	<p>Specifies an operation to perform:</p> <ul style="list-style-type: none"> <li>● A[ction Script]</li> <li>● B[lend]</li> <li>● C[lose]</li> <li>● E[xport]</li> <li>● I[mport]</li> <li>● O[pen]</li> <li>● R[efresh]</li> <li>● D[elete]</li> </ul> <p>This parameter is required</p>
/u=	Specifies the Data Relationship Management user name for login.
/pw=	Specifies the Data Relationship Management password for login.
/url=	<p>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, "net.tcp://localhost:5210/Oracle/Drm/ProcessManager"</p> <p>This parameter is required.</p>
/log=	<p>Specifies the path and file name for the log file. If the file exists, the log is appended to the end of the existing file. It is recommended that you put the value in double quotes.</p> <p><b>Note:</b> The specified log file may not be created if the /log parameter is placed after a problematic parameter such as a parameter with a missing or invalid value. To ensure that the specified log file is created, the /log parameter should be placed as the first parameter.</p> <p>This parameter is optional. If not specified, the following name is used: <code>DRM_Batch_ClientMMDDYYYY.log</code>, where <i>MM</i> = month, <i>DD</i> = day, <i>YYYY</i> = year</p>
/cfgfile=	Specifies the path and file name of the configuration file describing desired options. It is recommended that you put the value in double quotes.
/h[elp]   /?	Displays a list of command-line parameters.
/gmtoffsetminutes	<p>Local minutes offset from Greenwich Mean Time (GMT). For example, U.S. CST offset minutes from GMT is -360.</p> <p>This parameter is optional. The default is the server's time zone.</p>
/culturename	<p>Local culture name, in the format: <i>languagecode-country/regioncode</i> where <i>languagecode</i> is a lowercase two-letter code derived from ISO 639-1 and <i>country/regioncode</i> is an uppercase two-letter code derived from ISO 3166. For example, U.S. English is en-US.</p> <p>This parameter is optional. The default is the server's culture.</p>
/objectaccess	<p>Object access method for accessing import, export, and blender objects. Valid values are Standard, System, or User.</p> <p>This parameter is optional. The default is Standard.</p>

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

**Table 14** Action Script Parameters

Parameter	Description
<code>/infile=</code>	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.
<code>/delim=</code>	Specifies the delimiter. Specify one of the following values: <ul style="list-style-type: none"><li>● Tab</li><li>● Comma</li><li>● <i>#nnn</i></li></ul> where <i>#nnn</i> represents any valid 3-digit ASCII character value (0 - 255). Tab is the default value.
<code>/cols=</code>	Specifies the column order. If supplied, must contain all column identifiers. A12345 is the default value.
<code>/stripq=</code>	Specifies whether to remove quoted strings. Specify one of the following values: <ul style="list-style-type: none"><li>● Y[es]</li><li>● N[o]</li><li>● T[rue]</li><li>● F[alse]</li><li>● 1</li><li>● 0</li></ul> The default value is True.
<code>/propertyrefbylabel</code>	Look up property references by label (instead of name). Specify one of the following values: <ul style="list-style-type: none"><li>● Y[es]</li><li>● N[o]</li><li>● T[rue]</li><li>● F[alse]</li><li>● 1</li><li>● 0</li></ul>
<code>/encoding</code>	Character encoding of input file. Specify one of the following values: <ul style="list-style-type: none"><li>● UTF8</li><li>● UTF7</li><li>● UTF32</li><li>● UTF16LE]</li><li>● UTF16BE</li><li>● ASCII</li><li>● ANSI</li></ul> The default is UTF8.

**Table 15** Blend Parameters

Parameter	Description
/bl=	Specifies the name of blender to run. This parameter is required.
/sver=	Specifies the name for the source version to blend. This parameter is required.
/tver=	Specifies the name for the target version to blend. This parameter is required.
/nvera=	Specifies the name for the new version to create.
/nverd=	Specifies a description for the new version. You must enclose the value in double quotes.
/save=	Specifies whether to save the blended version or leave it detached. Specify one of the following values: <ul style="list-style-type: none"> <li>● Y[es], T[rue], or 1</li> <li>● N[o], F[alse], or 0</li> </ul> <p>The default value is False. For information on saving versions, see <a href="#">“Saving Versions in the Batch Client” on page 145</a>.</p>
/blendlog=	Specifies the name of the blend log file (not the same as the Batch Client log file).

**Table 16** Close Parameters

Parameter	Description
/abbrev=	Specifies the name for the version to close. This parameter is required.

**Table 17** Export Parameters

Parameter	Description
/xtype	Specifies the type of export: individual export or book. Specify one of the following values: <ul style="list-style-type: none"> <li>● E[xport]</li> <li>● B[ook]</li> </ul> <p>This parameter is required.</p>
/fdate=	Specifies the From date/time for transaction log exports (Optional) <b>Note:</b> Date/time formats are determined by the regional settings on the client computer.
/tdate=	Specifies the To date/time for transaction log exports (Optional) <b>Note:</b> Date/time formats are determined by the regional settings on the client computer.
/outfile=	For exports configured to output to a client file, this parameter specifies the path and file name for the output file. If the output file exists, it is overwritten. It is recommended that you put the value in double quotes. If this parameter is specified and the export is configured to send output to an external server connection, this parameter overrides the external server connection and the export is written to the file specified with /outfile.
/xname=	Specifies the name of the saved export to run. This parameter is required if running a single export.
/cver=	Specifies the name for the current version. This parameter is optional. If not specified, the system uses the value of the DefaultCurrentVersion system preference.

Parameter	Description
/pver=	Specifies the name for the previous version. This parameter is optional. If not specified, the system uses the value of the DefaultPreviousVersion system preference.
/cvertype=	Specifies the version type for the current version. Valid values are: <ul style="list-style-type: none"> <li>● A[s Of]</li> <li>● N[ormal]</li> </ul> N[ormal] is the default.
/pvertype=	Specifies the version type for the previous version. Valid values are: <ul style="list-style-type: none"> <li>● A[s Of]</li> <li>● N[ormal]</li> </ul> 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. <b>Note:</b> 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. <b>Note:</b> 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: <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> 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.
/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.
/fhier=	Specifies the From (Previous) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.
/thier=	Specifies the To (Current) Hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.
/fabbrv=	Specifies the name for the top node of the From (Previous) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.
/tabbrv=	Specifies the name for the top node of the To (Current) hierarchy. Used for Hierarchy Compare and Hierarchy Difference exports.

Parameter	Description
/hier=	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.
/nabbrev=	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.

**Table 18** Import Parameters

Parameter	Description
/infile=	For imports configured to import from a client file, this parameter specifies the path and file name for the input file. It is recommended that you put the value in double quotes. This parameter is only required for imports configured to import from a client file.  <b>Caution!</b> Do not use this parameter for imports configured to use an input file from an external server connection, otherwise an error occurs.
/iabbrev=	Specifies the name for the import to run. This parameter is required.
/implog=	Specifies the name of the import log file (not the same as the Batch Client log file.)
/vabbrev=	Specifies the version name to be created by import. The default value is stored in the import profile.
/vdescr=	Specifies the description of the version to be created by import. The default value is stored in the import profile.
/save=	Specifies whether to save the imported version or leave it detached. Specify one of the following values: <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> <p>The default value is False.</p> <p>For information on saving versions, see <a href="#">“Saving Versions in the Batch Client” on page 145</a>.</p>

**Table 19** Open Parameters

Parameter	Description
/vabbrev=	Specifies the name of the version to open. This parameter is required.

**Table 20** Refresh Parameters

Parameter	Description
/vabbrev=	Specifies the name of the version to refresh (close and re-open). This parameter is required.

**Table 21** Delete Parameters

Parameter	Description
/vabbrev=	Specifies the name for the version to delete. This parameter is required.

## 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.

```
[General]
Operation=Export
UserName=mylogin
Password=myspassword
URL=http://localhost/MDM/
LogFileName=C:\Hyperion\Master Data 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.

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

```
[General]
Operation=Multiple
UserName=mylogin
Password=myspassword
URL=net.tcp://localhost:5210/Oracle/Drm/ProcessManager
LogFileName=C:\Oracle\Data Relationship Management\rdsc.log
[Operations]
OperationCount=3
Operation1=Weekly_Import
Operation2=Update_CostCenters
Operation3=Merge_Data
[Weekly_Import]
```

```

Operation=Import
InFile="COA_data.txt"
ImportAbbrev="RDSC Import"
ImportLogFileName="C:\Oracle\Data Relationship Management\Batch Client\import_log.txt"
VersionAbbrev="COA Current"
VersionDescription="COA for this month"
[Update_CostCenters]
Operation=Action Script
InFile=CostCenter_changes.txt
StripQuotedStrings=True
Delimiter=comma
[Merge_Data]
Operation=Blend
Blendname="COA Blender"
SourceVersionAbbrev="COA Current"
TargetVersionAbbrev="COA Previous"
NewVersionAbbrev="COA Blended"
NewVersionDescription="Current and Previous COA"

```

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

**Table 22** General Section of Configuration File

Key	Description
Operation	<p>Specifies the operation to perform. Specify one of the following values:</p> <ul style="list-style-type: none"> <li>● M[ultiple]</li> <li>● A[ction Script]</li> <li>● B[lend]</li> <li>● C[lose]</li> <li>● E[xport]</li> <li>● I[mport]</li> <li>● O[pen]</li> <li>● R[efresh]</li> <li>● D[elete]</li> </ul> <p>This key is required</p>
UserName	Specifies the Data Relationship Management user name for login. This key is required.
Password	Specifies the Data Relationship Management password for login. This key is required.
URL	<p>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, <code>net.tcp://localhost:5210/Oracle/Drm/ProcessManager</code>.</p> <p><b>Note:</b> It is recommended that you put the value in double quotes.</p> <p>This key is required.</p>
LogFileName	<p>Specifies the path and file name for the log file. If the file exists, the log is appended to the end of the existing file. It is recommended that you put the value in double quotes.</p> <p>The default file name is <code>DRM_Batch_ClientMMDDYYYY.log</code>, where <i>MM</i> = month, <i>DD</i> = day, <i>YYYY</i> = year</p>



Key	Description
TimezoneOffset	Local minutes offset from GMT. For example, U.S. CST offset minutes from GMT is -360.  This parameter is optional. The default is the server's time zone.
CultureName	Local culture name, in the format: <i>languagecode-country/regioncode</i> where <i>languagecode</i> is a lowercase two-letter code derived from ISO 639-1 and <i>country/regioncode</i> is an uppercase two-letter code derived from ISO 3166. For example, U.S. English is en-US.  This parameter is optional. The default is the server's culture.
/objectaccess	Object access method for accessing import, export, and blender objects. Valid values are Standard, System, or User.  This parameter is optional. The default is Standard.

**Table 23** Multiple Section of Configuration File

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

**Table 24** Action Script Section of Configuration File

Key	Description
InFile	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.
StripQuotedStrings	Specifies whether to remove quoted strings. Specify one of the following values: <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> <p>The default value is True.</p>
Delimiter	Specifies the delimiter. Specify one of the following values: <ul style="list-style-type: none"> <li>● Tab</li> <li>● Comma</li> <li>● #<i>nnn</i></li> </ul> <p>where #<i>nnn</i> represents any valid 3-digit ASCII character value (0 – 255). Tab is the default value.</p>
ColumnOrder	Specifies the column order. If supplied, must contain all column identifiers. A12345 is the default value.

**Table 25** Blend Section of Configuration File

Key	Description
BlendName	Specifies the name of blender to run. This key is required.
BlendLogFileName	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.
SourceVersionAbbrev	Specifies the name for the source version to blend. This key is required.
TargetVersionAbbrev	Specifies the name for the target version to blend. This key is required.
NewVersionAbbrev	Specifies the name for the new version to create.
NewVersionDescription	Specifies a description for the new version. You must enclose the value in double quotes.
AutoSave	<p>Specifies whether to save the blended version or leave it detached. Specify one of the following values:</p> <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> <p>The default value is False.</p> <p>For information on saving versions, see <a href="#">“Saving Versions in the Batch Client” on page 145</a>.</p>

**Table 26** Close Section of Configuration File

Key	Description
VersionAbbrev	Specifies the name for the version to close. This key is required.

**Table 27** Export Section of Configuration File

Key	Description
ExportType	<p>Specifies the type of export: individual export or book. Specify one of the following values:</p> <ul style="list-style-type: none"> <li>● E[xport]</li> <li>● B[ook]</li> </ul> <p>This key is required.</p>
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.

Key	Description
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	<p>Specifies whether to combine the export output into the file named in <code>/outfile</code>. Specify one of the following values:</p> <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> <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 <code>/outfile</code>. The default value is stored in book.</p>
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.
Hierarchy $n$	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.
TopNodeAbbrev $n$	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.
FromDate	Specifies the From date/time for transaction log exports (Optional)
ToDate	Specifies the To date/time for transaction log exports (Optional)
CurrentVersionType	<p>Specifies the version type for the current version. Valid values are:</p> <ul style="list-style-type: none"> <li>● A[s Of]</li> <li>● N[ormal]</li> </ul> <p>The default is N[ormal].</p>

Key	Description
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: <ul style="list-style-type: none"> <li>● A[s Of]</li> <li>● N[ormal]</li> </ul> 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 28** Import Section for Configuration File

Key	Description
InFile	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.
ImportAbbrev	Specifies the name for the import to run. This key is required.
ImportLogFileName	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 <code>DRM_Batch_ClientMMDDYYYY.log</code> , where <i>MM</i> = month, <i>DD</i> = day, <i>YYYY</i> = year
VersionAbbrev	Specifies the version name to be created by import. The default value is stored in the import profile.
VersionDescription	Specifies the description of the version to be created by import. The default value is stored in the import profile.
AutoSave	Specifies whether to save the imported version or leave it detached. Specify one of the following values: <ul style="list-style-type: none"> <li>● Y[es]</li> <li>● N[o]</li> <li>● T[rue]</li> <li>● F[alse]</li> <li>● 1</li> <li>● 0</li> </ul> The default value is False. For information on saving versions, see <a href="#">“Saving Versions in the Batch Client” on page 145</a> .

**Table 29** Open Section for Configuration File

Key	Description
VersionAbbrev	Specifies the name of the version to open. This key is required.

**Table 30** Refresh Section of Configuration File

Key	Description
VersionAbbrev	Specifies the name of the version to refresh (close and re-open). This key is required. This key is required.

**Table 31** Delete Section of Configuration File

Key	Description
VersionAbbrev	Specifies the name for the version to delete. This parameter is required.

## 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_mmmddyyyy.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 32** Error Level Values

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

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



---

# 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.



---

# Index

---

## A

### Action Scripts

- loading from flat file, [81](#)
- loading from node model, [87](#)
- loading from transaction logs, [86](#)
- required parameters, [83](#)

### ancestor node

- defined, [42](#)

### annulling nodes, [49](#)

### As-Of versions

- creating, [30](#)

### assigning

- controlled node access groups to hierarchies, [37](#)
- controlled properties to hierarchies, [37](#)

### auditing

- request history, [141](#)
- transaction history, [135](#)

## B

### batch client

#### command line syntax

- Action Script parameters, [147](#)
- all operations, [146](#)
- blend parameters, [148](#)
- close parameters, [148](#)
- delete parameters, [151](#)
- export parameters, [148](#)
- import parameters, [150](#)
- open parameters, [150](#)
- refresh parameters, [150](#)

#### configuration file syntax, [151](#)

- action script section, [153](#)
- blend section, [154](#)
- close section, [154](#)
- delete section, [157](#)
- export section, [154](#)
- general section, [152](#)

#### import section, [156](#)

#### multiple section, [153](#)

#### open section, [156](#)

#### refresh section, [156](#)

#### dependencies, [145](#)

#### outputs and result codes, [157](#)

#### saving versions, [145](#)

### blenders

- copying, [99](#)
- creating, [95](#)
- deleting, [100](#)
- running, [99](#)

## C

### child node

- defined, [41](#)

### compare exports

- creating, [112](#)
- defined, [101](#)

### compares

- copying, [80](#)
- creating, [76](#)
- deleting, [80](#)
- navigating results, [78](#)
- opening, [79](#)
- property, [76](#)
- rename, [76](#)
- structure, [75](#)
- types, [75](#)
- visual, [75](#)

### creating

- As-Of versions, [30](#)
- hierarchies, [33](#)
- hierarchy groups, [36](#)
- versions, [29](#)

### custom sort order, [46](#)

**D**

Data Relationship Management

starting, 11

deleting

hierarchies, 38

nodes, 48

descendant node

defined, 41

difference exports, 101

creating, 116

**E**

editing

hierarchy properties, 34

export books

copying, 131

creating, 130

deleting, 132

running, 131

exports

copying, 131

deleting, 132

running, 131

types, 101

**F**

finding nodes, 52

**G**

generation exports

creating, 108

defined, 101

global nodes

defined, 42

**H**

hierarchies

assigning controlled node access groups to, 37

assigning controlled properties to, 37

assigning to hierarchy groups, 37

assigning validations to, 38

creating, 33

deleting, 38

editing properties, 34

opening, 36

validating, 66

hierarchy

defined, 33

hierarchy exports

creating, 102

defined, 101

hierarchy groups

assigning hierarchies to, 37

creating, 36

hierarchy properties, 34

hierarchy XML exports

creating, 107

defined, 101

**I**

imports

copying, 93

creating, 90

deleting, 94

running, 93

sections, 89

inactivating nodes, 49

**L**

leaf node

defined, 41

limb node

defined, 41

loading Action Scripts

from flat files, 81

from node model, 87

from transaction logs, 86

local nodes

defined, 42

**M**

merge exports

defined, 102

merge log exports

creating, 128

**N**

node types

enabling and disabling, 35

nodes

- adding, [42](#)
- annulling, [47, 49](#)
- annulling all below, [47](#)
- assigning validations to, [50](#)
- defined, [41](#)
- deleting, [47, 48](#)
  - merging, [47](#)
  - methods, [47](#)
  - orphans, [51](#)
- deleting and merging, [47](#)
- destroying, [47](#)
- finding, [52](#)
- ID function, [43](#)
- inactivating, [47, 49](#)
- inactivating and merging, [47](#)
- inserting, [42](#)
- modeling, [44](#)
- moving, [45](#)
  - relaxed move, [46](#)
- orphan, [50](#)
- reactivating, [50](#)
- removing, [47, 48](#)
- shared
  - adding and inserting, [55](#)
  - converting to primary, [56](#)
  - displaying, [57](#)
  - enabling, [55](#)
  - properties, [54](#)
  - removing and deleting, [56](#)
  - renaming, [56](#)
  - using, [53](#)
- sorting, [46](#)
- validating, [66](#)
- viewing orphan, [51](#)

**O**

- opening
  - hierarchies, [36](#)
- orphan nodes, [50](#)
  - defined, [42](#)
  - deleting, [51](#)
  - viewing, [51](#)

**P**

- parameters in Action Script, [83](#)
- parent node

- defined, [41](#)
- properties
  - hierarchy, [34](#)
- property category, [59](#)
- property list export
  - defined, [102](#)
- property list exports
  - creating, [122](#)
- property lookup exports
  - creating, [123](#)
  - defined, [102](#)
- property status, [62](#)

**Q**

- queries
  - copying, [73](#)
  - creating, [70](#)
  - deleting, [74](#)
  - global, [69](#)
  - local, [69](#)
  - navigating results, [71](#)
  - opening, [73](#)
  - overview, [69](#)
  - types, [69](#)

**R**

- reactivating nodes, [50](#)
- removing nodes, [48](#)
- request history
  - auditing, [141](#)

**S**

- shared nodes
  - adding and inserting, [55](#)
  - converting to primary, [56](#)
  - displaying, [57](#)
  - enabling, [55](#)
  - properties, [54](#)
  - removing and deleting, [56](#)
  - renaming, [56](#)
  - using, [53](#)
- sibling node
  - defined, [42](#)
- sort order
  - custom, [46](#)
- sorting nodes, [46](#)

starting Data Relationship Management, 11

## T

- transaction history
  - auditing, 135
- transaction log exports
  - creating, 125
  - defined, 102

## V

- validations
  - assigning to hierarchies, 38
  - assigning to nodes, 50
  - assigning to versions, 31
  - navigating results, 67
  - running on hierarchies, 66
  - running on nodes, 66
  - running on versions, 65
- version
  - defined, 27
- version exports
  - creating, 119
  - defined, 101
- version properties
  - editing, 31
- versions
  - assigning validations to, 31
  - closing, 32
  - copying, 29
  - creating, 29
  - creating As-Of, 30
  - deleting, 32
  - running validations, 65
  - saving, 30
  - status, 29
  - types, 29