

Oracle Data Relationship Management 11.1.2 Administration

Student Guide

D69373GC20
Edition 2.0
August 2013
D83577

ORACLE®

Author

Sergiy Pecherskyy

**Technical Contributors
and Reviewers**

Matt Lontchar

Brian Brocklesby

Kevin Mckenzie

Randy Wilson

Mark Brieden

Stanley Ziolkowski

Richard Conway

Rahul Kamath

Editors

Daniel Milne

Richard Wallis

Graphic Designer

Divya Thallap

Publishers

Sujatha Nagendra

Nita Brozowski

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Contents

1 Product Orientation

- Objectives 1-2
- Data Relationship Management 1-3
- Enterprise Dimensions 1-4
- Example Data Relationship Management System 1-5
- Data Relationship Management Architecture 1-6
- Implementation Process for Data Relationship Management Applications 1-8
- Users of Data Relationship Management 1-10
- Connecting to Applications with Web Client 1-11
- Accessing Task Groups 1-12
- Changing a Password 1-14
- Viewing Help 1-15
- Summary 1-16
- Quiz 1-17

2 Working with Versions and Hierarchies

- Objectives 2-2
- Working with Versions 2-3
- Creating and Saving Versions 2-5
- Changing Version Status 2-6
- Copying Versions 2-8
- Backing Up and Restoring Versions 2-10
- Deleting Versions 2-11
- Version Variables 2-12
- Viewing and Editing Version Properties 2-14
- Working with Hierarchies 2-15
- Creating, Opening, and Deleting Hierarchies 2-16
- Viewing and Editing Hierarchy Properties 2-18
- Implementation Considerations 2-19
- Summary 2-20
- Quiz 2-21

3 Working with Nodes

- Objectives 3-2
- About Nodes 3-3

Adding and Deleting Nodes	3-4
Removing and Inserting Nodes	3-6
Viewing Orphan Nodes	3-8
Moving Nodes	3-9
Ordering Nodes	3-10
Inactivating and Reactivating Nodes	3-11
Annulling Nodes	3-12
Copying Nodes and Properties	3-13
About Domains	3-15
Creating Domains	3-16
Assigning Nodes to Domains	3-18
Creating Shared Nodes	3-20
Viewing and Editing Node Properties	3-22
Implementation Considerations	3-24
Summary	3-25
Quiz	3-26

4 Defining Properties

Objectives	4-2
Custom Properties	4-3
Namespaces	4-4
Data Types	4-5
Node Data Types	4-6
Property Levels	4-8
Local and Global Properties	4-9
Property Types	4-10
Configuring Defined Properties	4-11
Configuring List Values	4-13
Configuring Lookup Values	4-14
Configuring Inheriting Properties	4-15
Assigning Controlling Hierarchies to Global Inheriting Properties	4-16
Creating Property Categories	4-18
Loading Metadata with the Migration Utility	4-20
Summary	4-22
Quiz	4-23

5 Importing Data

Objectives	5-2
About Importing Data	5-3
Importing from Files	5-4
Importing from Database Tables	5-6

Opening, Copying, and Deleting Imports 5-7
Creating Imports 5-9
Configuring the Source: Import Files 5-10
Configuring the Source: Database Tables 5-12
Configuring the Style 5-13
Configuring the Columns: Import Files 5-16
Configuring the Columns: Database Table 5-18
Configuring the Filters and Target 5-19
Filtering Source Tables 5-20
Saving Imports 5-21
Running Imports 5-22
Summary 5-24
Quiz 5-25

6 Querying and Comparing Data

Objectives 6-2
Searching for Nodes 6-3
Creating Queries 6-5
Configuring the Source 6-6
Configuring the Style 6-7
Configuring the Filters 6-8
Configuring the Columns 6-10
Running and Saving Queries 6-12
Creating Compares 6-13
Performing Visual Compares 6-14
Creating Structure Compares 6-15
Example: Structure Compare to Find Differences 6-17
Creating Rename Compares 6-18
Creating Property Compares 6-19
Working with Query and Compare Results 6-21
Summary 6-23
Quiz 6-24

7 Updating Data with Action Scripts

Objectives 7-2
Action Scripts 7-3
Creating Flat Files for Action Scripts 7-4
Loading Action Scripts from Flat Files 7-6
Running Action Scripts and Reviewing Results 7-8
Downloading Action Script Results 7-9
Automating Version Changes with Action Scripts from Transaction Logs 7-10

Modeling Nodes by Using Action Scripts 7-12
Summary 7-14
Quiz 7-15

8 Blending Versions

Objectives 8-2
Blending Versions 8-3
Opening, Copying, and Deleting Blenders 8-5
Creating Blenders 8-7
Configuring Blender Versions and Hierarchies 8-8
Configuring Blender Processing Options 8-10
Configuring Blender Filters 8-12
Configuring Blender Properties 8-14
Configuring Blender Target Options 8-16
Saving Blenders 8-18
Running Blenders 8-19
Summary 8-20
Quiz 8-21

9 Creating Derived Properties

Objectives 9-2
Derived Properties 9-3
Derived Properties Using Formulas 9-4
Formula Syntax 9-5
Functions 9-6
Logical Operators 9-8
Creating Derived Properties Using Formulas 9-9
Testing Formulas 9-11
Derived Properties Using Scripts 9-13
Script Syntax 9-14
Property Enumeration Constants 9-15
JavaScript Data Type Objects and Functions 9-16
Print and Format Functions 9-17
Data Relationship Management Objects 9-18
Creating Scripts 9-20
Summary 9-21
Quiz 9-22

10 Validating Data

Objectives 10-2
Validations 10-3

Validation Levels	10-4
Validation Classes	10-6
Creating Validations	10-8
Validation Using Scripts	10-10
Validation Enumeration Constants	10-12
Assigning Validations	10-13
Running Batch Validations	10-15
Working with Batch Validation Results	10-16
Editing and Deleting Validations	10-18
Summary	10-19
Quiz	10-20

11 Setting Up Node Types

Objectives	11-2
Node Types	11-3
Process for Setting Up Node Types	11-5
Creating, Editing, and Deleting Node Types	11-6
Creating, Editing, and Deleting Glyphs	11-8
Assigning Node Types to Nodes	11-10
Defined Node-Level Property	11-11
Inherited Node-Level Property	11-12
Lookup Node-Level Property	11-13
Derived Node-Level Property	11-14
Hierarchy-Level and Version-Level Properties	11-15
Configuring Hierarchy Node Type Property Values	11-16
Troubleshooting Node Type Assignments	11-17
Summary	11-18
Quiz	11-19

12 Exporting Data

Objectives	12-2
Exporting Data	12-3
Export Types	12-4
Opening, Copying, and Deleting Exports	12-6
Creating Exports	12-8
Creating External Connections	12-11
Starting New Exports and Selecting Export Types	12-13
Configuring a Hierarchy Export Source	12-14
Configuring a Hierarchy Export Style	12-15
Configuring a Hierarchy Export Filter	12-17
Configuring Hierarchy Export Columns	12-18

Configuring a Hierarchy Export Target 12-20
Saving Exports 12-22
Running Exports 12-23
Creating and Running Export Books 12-25
Summary 12-27
Quiz 12-28

13 Managing Security

Objectives 13-2
Security Levels 13-3
Managing User Security 13-5
User Authentication Modes 13-6
Creating Users 13-7
Locking and Unlocking Users 13-9
Changing User Passwords 13-10
Managing Access to Nodes 13-11
Interactive Group-Type Node Access Levels 13-12
Workflow Group-Type Node Access Levels 13-14
Global Node Access Groups 13-15
Creating, Editing, and Deleting Node Access Groups 13-16
Configuring Limb and Leaf Access for Node Access Groups 13-18
Assigning Access to Property Categories 13-19
Assigning Access to Metadata Objects 13-21
Creating, Editing, and Deleting Object Access Groups 13-22
Granting Anonymous Access 13-24
Summary 13-25
Quiz 13-26

14 Configuring Governance Workflows

Objectives 14-2
Data Governance 14-3
Governance Workflow Objects 14-4
Workflow Tasks 14-5
Workflow Models 14-7
Working with Requests 14-10
Navigating the Worklist 14-12
Request Life Cycle 14-14
Creating Requests 14-16
Calculating Request Items 14-18
Validating Requests 14-19
Submitting Requests for Approval 14-20

Claiming Requests 14-21
Enriching Requests 14-22
Approving and Rejecting Requests 14-23
Alerts and Notifications 14-24
Summary 14-25
Quiz 14-26

15 Analyzing Data Changes

Objectives 15-2
Auditing Data 15-3
Querying and Viewing Transaction History 15-4
Viewing Transaction History Details 15-6
Downloading Transaction History 15-7
Analyzing Changes Over Time 15-8
Viewing Baseline Versions 15-9
Creating As-Of Versions 15-10
Viewing Change-Tracking Properties 15-12
Summary 15-13
Quiz 15-14

A Migrating Metadata

Migrating Metadata A-2
Launching the Migration Utility A-3
Extracting Metadata A-4
Loading Metadata A-6
Comparing Metadata A-8
Viewing Metadata and Generating Reports A-9

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

1

Product Orientation

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the benefits of Data Relationship Management
- Explain the Data Relationship Management architecture
- Describe the process for implementing Data Relationship Management applications
- Identify and describe the users of Data Relationship Management
- Connect to applications with Web Client
- Navigate Web Client



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management

- Enables you to build and manage enterprise dimensions
- Streamlines the process for updating dimensional data
- Aligns critical dimensions across systems
- Publishes dimensions to downstream systems
- Simplifies compliance with regulatory standards
- Is technology-independent and supports multiple system vendors, data models, and architectures



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Data Relationship Management provides the following benefits:

- It enables you to build and manage enterprise dimensions.
- It streamlines the process for updating dimensional data.
 - Data is updated in one location. As a result, data stays current and accurate, and reporting inconsistencies are significantly reduced.
 - Business rules, technology constraints, and structural integrity rules across multiple hierarchies are automatically enforced. Attribute values from formulas or inheritance rules are automatically derived.
 - Business users manage their own data in a format that they can easily understand.
 - It integrates with appropriate workflow and approval processes.
- It aligns critical dimensions across systems, such as EPM, BI, ERP, and data warehouses.
- It enables you to publish dimensions to downstream systems when required.
- It simplifies compliance with regulatory standards, and as a result, reduces financial close times. It provides a complete audit trail with change control, and provides historical versions of dimensions for analysis and potential restatements.
- It is technology-independent and supports multiple system vendors, data models, and architectures.

Enterprise Dimensions

- Reflect how the enterprise measures value
- Are shared across multiple business functions and systems
- Classify transactional and analytical data (usually hierarchical data)
- Examples:

FINANCIAL	ORGANIZATIONAL
<ul style="list-style-type: none">• Charts of accounts: Support multiple accounting standards (for example, IFRS and US GAAP)• Cost centers	<ul style="list-style-type: none">• Organization hierarchies• Legal entity structures• Location hierarchies: Store locations and regional management



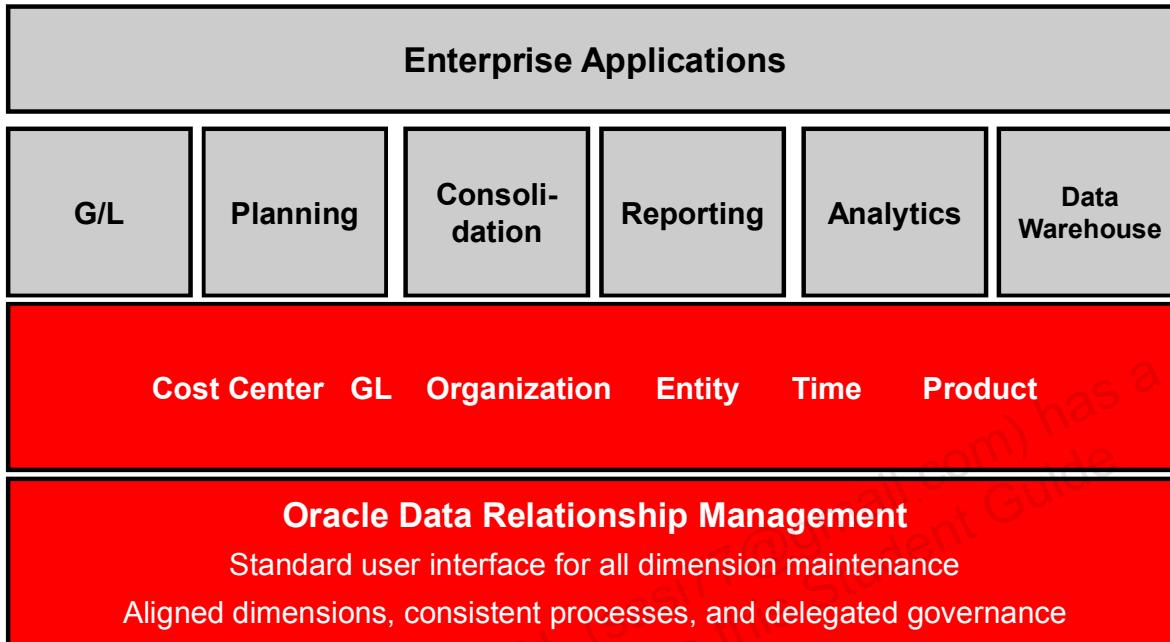
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Enterprise dimensions mostly define how a business measures its value. Therefore, it makes sense that these dimensions are shared as widely as possible, and are aligned across the enterprise. The dimensions classify transactional and analytical data (usually hierarchical data).

Examples of dimensions:

- **Financial dimensions:** Charts of accounts for supporting multiple accounting standards, such as International Financial Reporting Standards (IFRS) and United States Generally Accepted Accounting Principles (US GAAP), and cost centers
- **Organizational dimensions:** Organization hierarchies, legal entity structures, and location hierarchies (store locations and regional management)

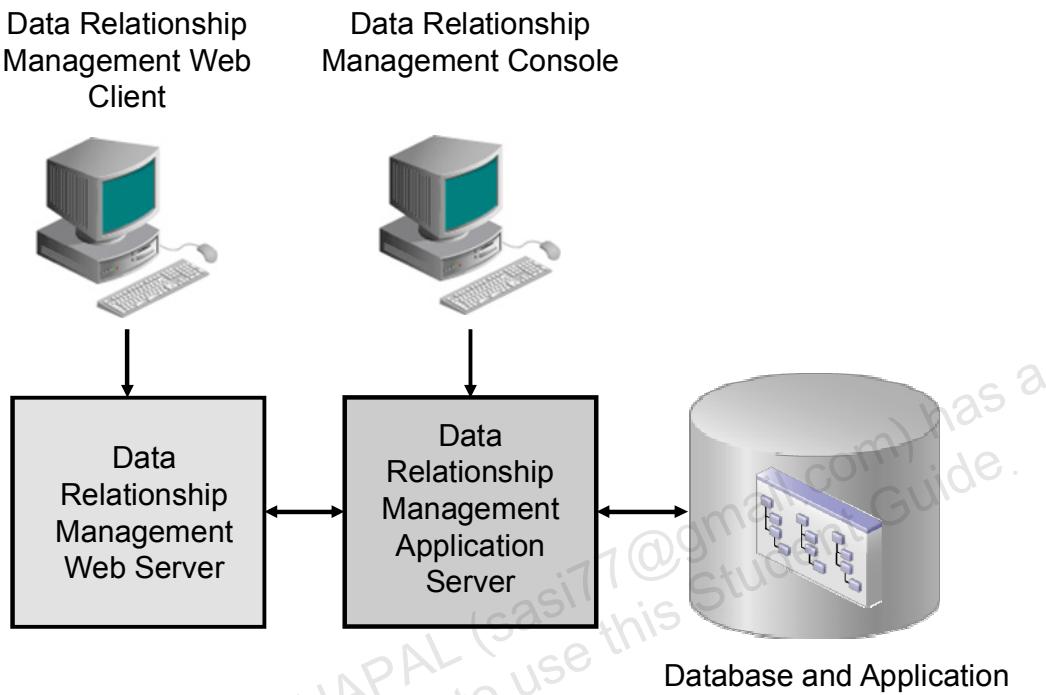
Example Data Relationship Management System



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the slide example, the general ledger (GL), planning, consolidation, reporting, analytics, and data warehouse applications all share the Cost Center, General Ledger, Organization, Entity, Time, and Product dimensions. Each use-case of a dimension may have a different subset or superset of the dimension, with different attribution, filters, and so on. Data Relationship Management is the standard user interface for managing these dimensions. You can align dimensions, have consistent processes, and delegate governance.

Data Relationship Management Architecture



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management architecture consists of the following components:

- **Data Relationship Management Web Client (Web Client)**
 - Is the primary user interface for managing data
 - Requires Mozilla Firefox or Internet Explorer
- **Data Relationship Management Console**
 - Is used to set up, configure, and monitor Data Relationship Management applications
 - Resides on the Application server
- **Data Relationship Management web server**
 - Brokers user requests
 - Requires Microsoft Internet Information Server
- **Application server**
 - Processes user requests and manages metadata and data for each application
 - Requires Windows 2003 SP1 (32- or 64-bit) or Windows 2008 Server (32-or 64-bit) and .NET Framework 4.0

- **Database**
 - Stores Data Relationship Management application data
 - Requires Oracle 10.1.x to 11.2.x or SQL Server 2005
- **Application**
 - Is a store of Data Relationship Management metadata and data
 - Resides in the database

The slide shows a high-level view of the information flow between the components. Web Client sends a request to the Data Relationship Management web server, which in turn communicates with the application server. The application server loads application data from the database, processes the user request, and returns the results of the request back to the Web Client through the web server.

Implementation Process for Data Relationship Management Applications

1. Create applications.
2. Create versions, hierarchies, and nodes.
3. Define properties.
4. Import data into applications.
5. Query and compare data.
6. Update data with action scripts.
7. Blend versions.
8. Create properties with formulas, and validate data.
9. Set up node types.
10. Export data.
11. Implement security, and set system preferences.
12. Audit and archive data.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The steps listed in the slide outline the process for implementing Data Relationship Management applications. The learning activities in this course follow this process in detail:

1. **Create applications:** In Data Relationship Management Console, you define one or more applications and build the necessary database structures. You can create multiple applications to support the many business needs for hierarchical data in an organization. Each application is independent of all other applications.
2. **Create versions, hierarchies, and nodes:** In Web Client, you begin the initial steps of designing an application by creating versions, hierarchies, and nodes. By accessing reports and other applications, you define the structure of your application.
 - Versions represent independent sets of data that is arranged into related hierarchies (for example, March 2013 data).
 - Hierarchies define structural relationships among data (for example, Geographic structure).
 - Nodes are points in hierarchies (for example, USA and California).
3. **Define properties and property categories:** You define properties for your application based on properties from other applications and reports. When properties are available from other applications, you streamline this step by using the Migration Utility to import the properties into your application.

- *Properties* define the attributes of versions, hierarchies, and nodes (for example, Name or Description).
 - *Property categories* organize properties into management subsets.
4. **Import data into applications:** With your application structure in place, you obtain relevant data from other applications and import it into your application. An *import* adds hierarchies, nodes, and properties to versions.
 5. **Query and compare data:** You review the data in your application by performing queries and comparisons. By examining how the data is structured and identifying conflicts, you identify the needed changes and fix it.
 - *Queries* find nodes based on their property values.
 - *Compares* find differences or similarities in hierarchy structures and property values.
 6. **Update data with action scripts:** You process bulk sets of incremental changes to data (add nodes and update properties) in an automated fashion with *action scripts*.
 7. **Blend versions:** With “clean” data, you combine versions to create a “single version of the truth” in your application. *Blenders* combine hierarchies, nodes, and properties from two different versions into the same version.
 8. **Create properties with formulas, and validate data:** You ensure data quality and enforce business rules by creating properties that consist of formulas (for example, account numbers must have eight numeric characters). You automate the process of checking and applying those rules by defining *validations*.
 9. **Set up node types:** You filter property lists for nodes by defining nodes types. You also assign validations to nodes through node types.
 10. **Export data:** If needed, you publish data to downstream dimensions.
 - *Exports* extract data to files and database tables.
 - *External connections* define server and database targets for exports.
 11. **Implement security, and set system preferences:** With your application built, you are ready to set up user access with users and node access groups. You can also configure *system preferences*, which are settings that control the behavior of Data Relationship Management, including security.
 - *Users* are authorized people in your organization who are explicitly granted access to data, validations, and functionality through roles.
 - *Node access groups* group users with similar access privileges and streamline the process of setting up user access.
 12. **Audit and archive data:** You can query and view transactions and requests. You can also use the Migration Utility to archive data.
 - *Transactions* are individual data changes to a version (for example, a node is added).
 - *Requests* are groups of proposed data changes, or transactions, to a version through a workflow application (for example, a proposal to update four properties and remove a node).

Users of Data Relationship Management

- End users
 - Interactive users
 - Data creators
 - Governance users
 - Workflow users
 - Anonymous users
- Administrators
 - Data managers
 - Access managers
 - Application administrators
- Implementation consultants



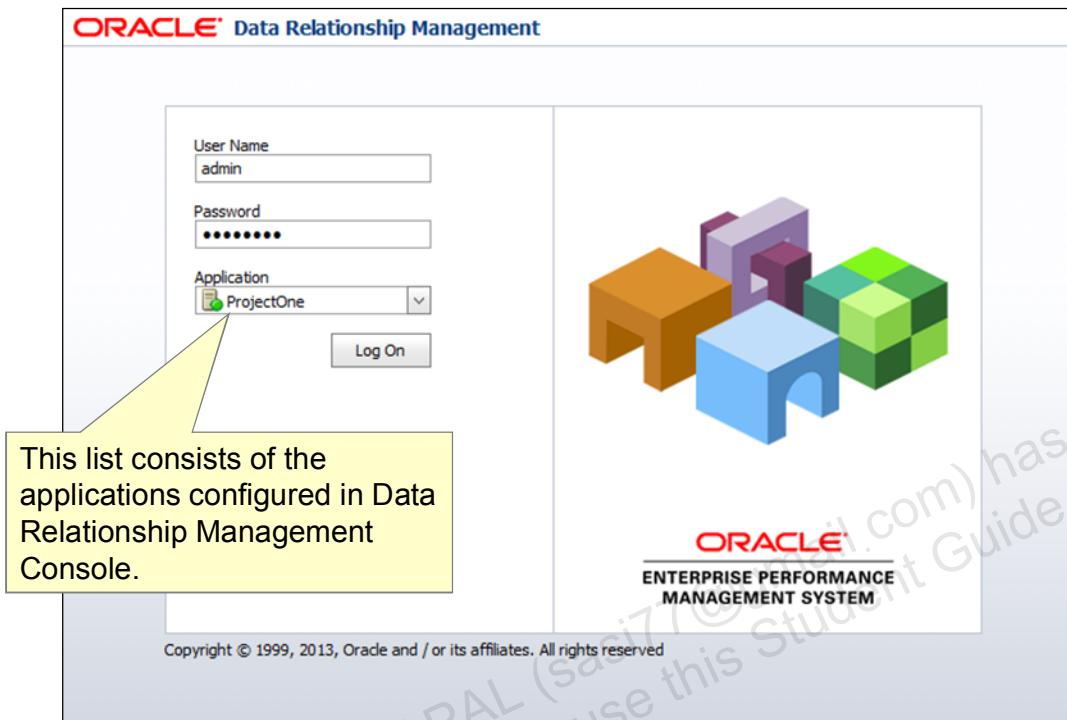
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management is designed for the following types of users:

- **End users**
 - *Interactive users* view, create, and edit nodes in hierarchies based on node access group and property category security.
 - *Data creators* create new versions and hierarchies, but cannot manage data created by other users.
 - *Governance users* submit and approve requests for changes to hierarchy nodes and their properties using workflow tasks and models.
 - *Workflow users* submit and approve requests for changes to hierarchies and nodes using an external workflow application.
 - *Anonymous users* access views of data anonymously through a URL.
- **Administrators**
 - *Data managers* manage all versions and hierarchies, user group access to the data, and standard metadata objects.
 - *Access managers* manage users and their assignment to security features.
 - *Application administrators* manage system metadata for applications.
- **Implementation consultants** install and configure Data Relationship Management, build initial applications for administrators and end users, and integrate products and develop custom Data Relationship Management applications and clients.

Connecting to Applications with Web Client



Copyright © 1999, 2013, Oracle and / or its affiliates. All rights reserved.

ORACLE®

To connect to applications with Web Client:

1. Select **Start**, then **Programs**, then **Oracle EPM System**, then **Data Relationship Management**, and then **Web Client**.
The logon page for Oracle Hyperion Data Relationship Management is displayed in your default web browser.
2. In the User Name box, enter the default user name: **admin**.
3. In the Password box, enter the default admin password (for example, **Welcome!**).
4. In the Application box, select one of your configured applications.
5. Click **Log On**.

Web Client displays the Home page by default.

Accessing Task Groups

Select links to task groups.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you log on to Web Client, the Home page is displayed. On the side of the Home page, you can access the following task groups:

- **Worklist**
 - You can create, view, and search for requests using governance workflows to process proposed changes to data.
 - When a request is created or opened, the request is displayed on a separate instance tab. Request-level actions are performed on the instance tab for each request.
- **Browse**
 - You can browse and manage versions, hierarchies, nodes, and their properties as well as search for nodes and orphan nodes.
- **Query**
 - You can create, open, copy, and delete queries
 - You can edit query results based on your user rights, which enables you to make changes without navigating to the Browse task group.
 - Multiple queries can be open, but only one can be in focus at a time.

- **Compare**
 - You can create, open, copy, and delete compares.
 - You can edit compare results based on your user rights, which enables you to make changes without navigating to the Browse task group.
 - Multiple compares can be open, but only one can be in focus at a time.
- **Script**
 - You can load an action script from a file, transaction log, or node model to apply a bulk set of incremental changes to a version.
 - You can edit action scripts after they are loaded.
- **Import**
 - You can create, open, copy, and delete imports.
 - You can open and run multiple imports in separate tabs, but only one can be in focus at a time.
- **Blend**
 - You can create, open, copy, and delete blenders.
 - You can open and run multiple blenders in separate tabs, but only one can be in focus at a time.
- **Export**
 - You can create, open, copy, and delete exports and books.
 - *Books* are exports that are grouped and run together.
- **Audit**
 - You can query and view data and system transactions.
 - You can query and view change requests submitted by users through a workflow process.
 - You can define query filters and select log fields to be displayed as columns in the query results.
- **Administer**
 - You can administer system metadata and security.
 - System metadata tasks include creating and managing property categories, hierarchy groups, property definitions, validations, node types, external connections, and glyphs. You can also manage system preferences.
 - Security tasks include creating and managing users and node access groups. You can also assign users to property categories.

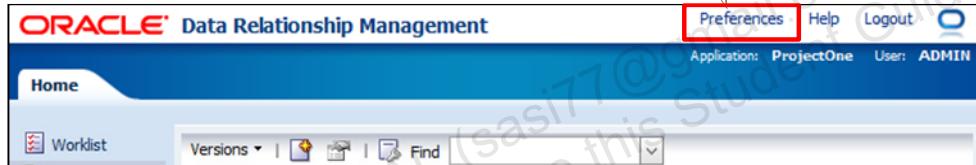
Note: Depending on the roles assigned to you, a limited set of task groups may be available for selection.

Changing a Password

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

Select Preferences to change your password.



ORACLE

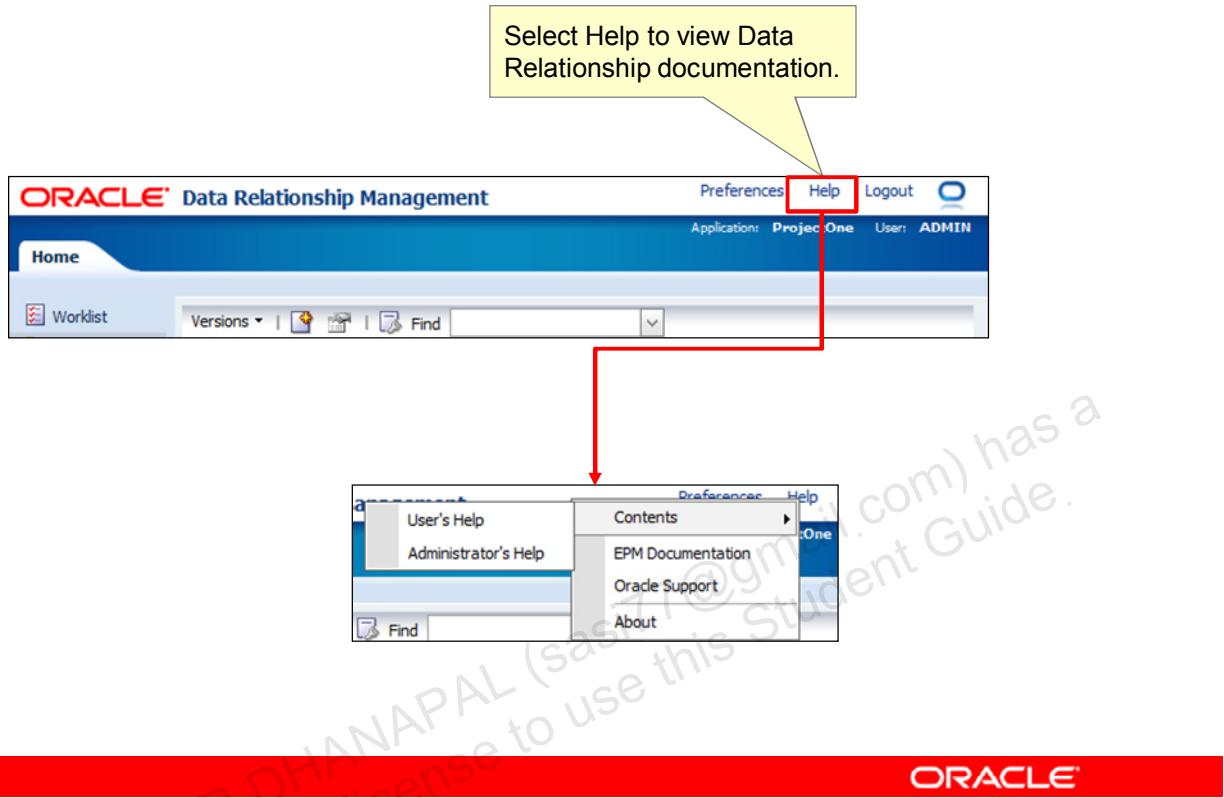
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the user authentication mode was defined as Internal during installation for you, then you can change your password from any page in Web Client.

To change your password:

1. On the header of a Data Relationship Management page, select **Preferences**.
2. Click **Change My Password**.
The Change Your Password dialog box is displayed.
3. Enter your current password.
4. Enter and reenter a new password, and click **OK**.
5. Select **Home** to return to the Home page.

Viewing Help



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

To view help:

1. On the Data Relationship Management Home page, select **Help**, and then select one of the following options:
 - **Contents:** Select **User's Help** or **Administrator's Help**, and then search for information in the table of contents (Content section), in the index (Index section), or by keywords (Search section).
 - **EPM Documentation:** Browse documentation on Oracle's website. You must be connected to the Internet.
 - **Oracle Support:** Browse Oracle's Support website. You must be connected to the Internet.
 - **About:** View version and copyright information.
2. View the information, and close the window.

Summary

In this lesson, you should have learned how to:

- Describe the benefits of Data Relationship Management
- Explain the Data Relationship Management architecture
- Describe the process for implementing Data Relationship Management applications
- Identify and describe the users of Data Relationship Management
- Connect to applications with Web Client
- Navigate Web Client



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

On the Home page in Web Client, you can perform which of the following tasks?

- a. Access task groups.
- b. Access user preferences.
- c. Access help.
- d. Create Data Relationship Management applications.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

When you implement Data Relationship Management, you usually define properties in an application before you import data.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Administrators manage which of the following items?

- a. Installation
- b. Application security
- c. Application data
- d. Application metadata



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b, c, d

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Working with Versions and Hierarchies



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Objectives

After completing this lesson, you should be able to:

- Create, save, change statuses of, copy, back up, restore, and delete versions
- Create version variables and assign them to versions
- Create, open, and delete hierarchies
- View and edit properties for versions and hierarchies
- Describe implementation considerations for versions and hierarchies

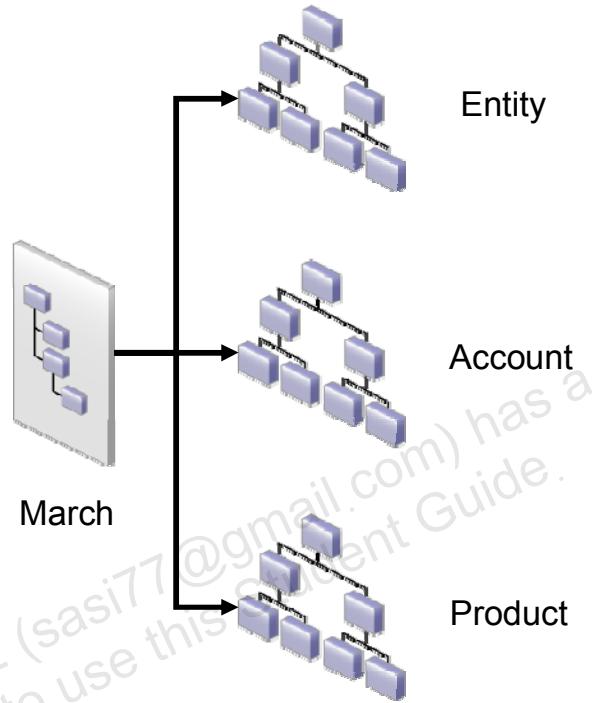


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Working with Versions

Versions:

- Represent independent sets of data
- Encapsulate hierarchies
- Are typically related to business time periods
- Support what-if scenarios
- Have types:
 - Normal
 - Baseline
 - As-Of



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management groups a set of hierarchies into versions. Versions are arranged into related hierarchies and are usually related to time periods or scenarios. Examples of versions are March, 3rd Quarter, and Plan. In the slide example, a version named March consists of three hierarchies: Entity, Account, and Product.

Versions are typically used for the following purposes:

- To represent a set of hierarchies used during a particular month (or other business cycle period). A new version created each month, for example, allows the historical records to be stored.
- To plan changes in the current version for the next main business cycle period (for example, for the next fiscal year)
- To support what-if scenarios during reorganizations, restructures, and mergers
- To differentiate between real data and test data during system testing
- To compare different versions to identify changes that are being made to the hierarchies over time

Version Types

The following version types are available:

- **Normal** is a version that you can edit and change its status over time.
- **Baseline** is a version that is a copy of a normal version at the time it was saved.
- **As-Of** is a version that is a generated copy of a normal version at a point in time before the current date but after the date the version was saved.

Normal versions are displayed in the user interface by default. You can view baseline and As-Of versions under each normal version.

Loading Versions

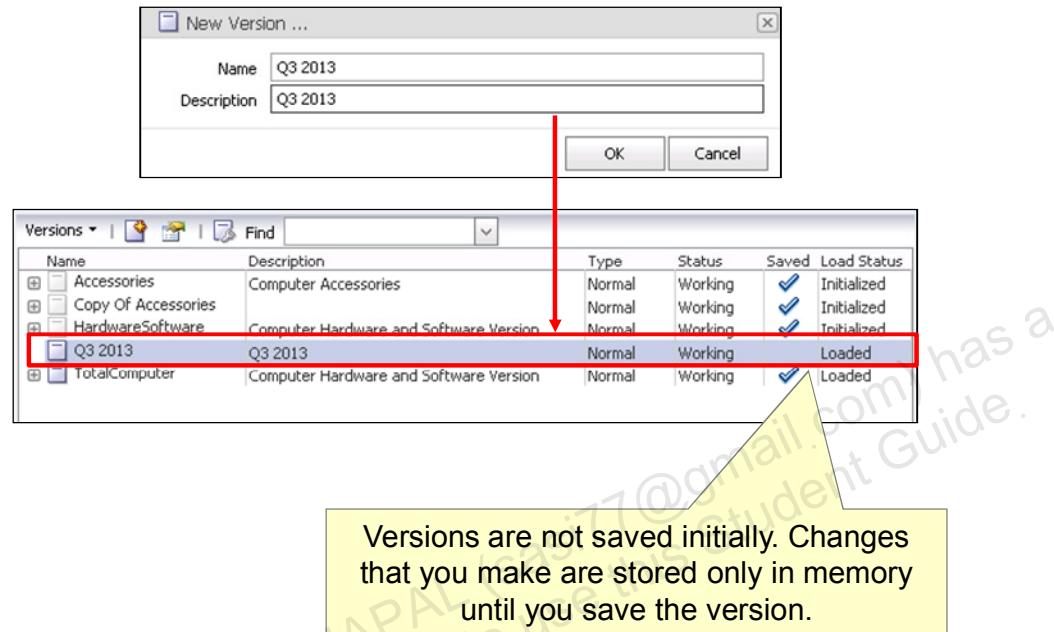
Data Relationship Management works with versions in memory on the application server. A version must be loaded into memory before you can create any hierarchies or edit any properties. Versions that are newly created or copied are automatically loaded for immediate use.

If a version is not loaded (load status of Initialized), then you can load the version by using the Load option or by requesting data from it.

To load versions:

- In the Browse task group, right-click a version, and select **Load**.
Loaded is displayed in the Load Status column.

Creating and Saving Versions



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To set up the initial environment in the application, you create a new empty version. New versions are created in an unsaved state. Until you save them, changes that you make are made in memory only and are not saved to the database.

To create and save versions:

1. In the Browse task group, select **Versions**, then **New**, and then **Version**.
2. Enter a name and description for the version, and click **OK**.

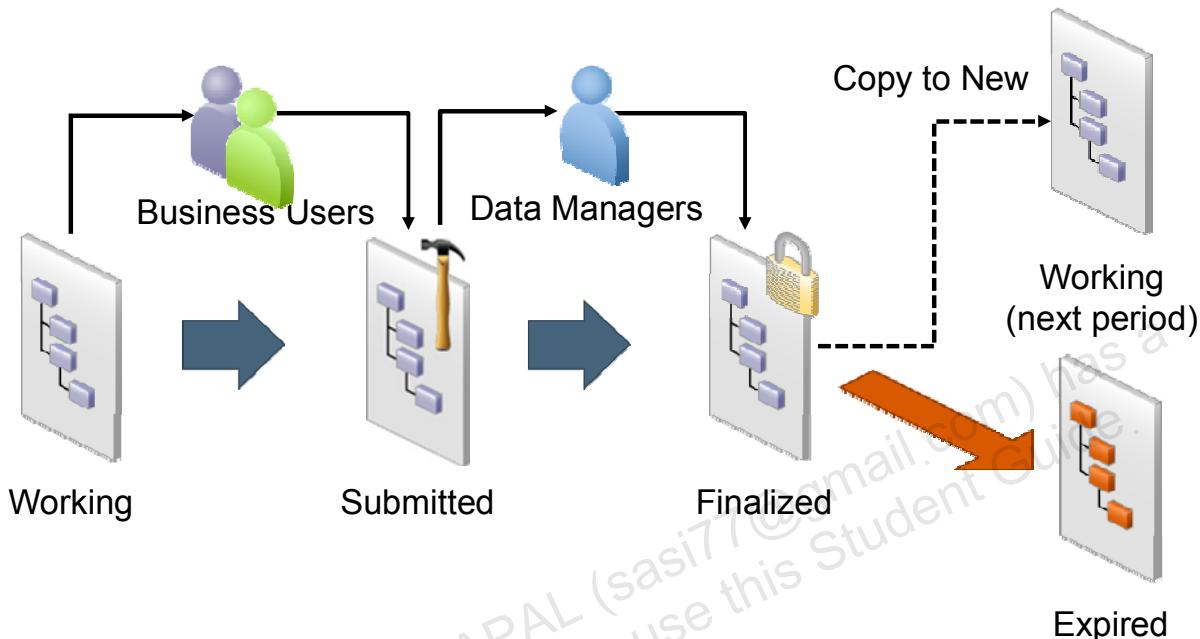
When the version tree refreshes, the new version is displayed in a detached (unsaved) state and you can start adding data to it.

3. Right-click the version, and select **Save**.

The version is saved, and a check mark is displayed in the Saved column.

Changing Version Status

Versions support data life cycles through status management.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Most organizations use data on a cyclical basis that matches their management reporting periods (typically every month). Within each reporting period, the use of data follows a predictable pattern that includes the following phases:

1. A version owner creates a new working version as a copy of the finalized version for the previous period.
2. End users make changes to the working version. Validations are automatically performed as users enter or modify data.
3. As the reporting period deadline looms, the version status is changed to Submitted. Version and hierarchy owners run comparisons to identify changes and perform validations to ensure the integrity of the data.
4. When data integrity is assured, the data manager changes the version status for the current reporting period to Finalized, and the version status for the previous reporting period from Finalized to Expired. The expired version is stored for possible future use in historical analyses or as an audit record.
5. The administrator performs exports from the new finalized version to feed hierarchy data to participating systems. After all exports are completed and loaded by the destination systems, participating systems have consistent hierarchical data as a basis for the period-end reporting process.

The version status controls the workflow of hierarchy maintenance. The version statuses are described in the following table:

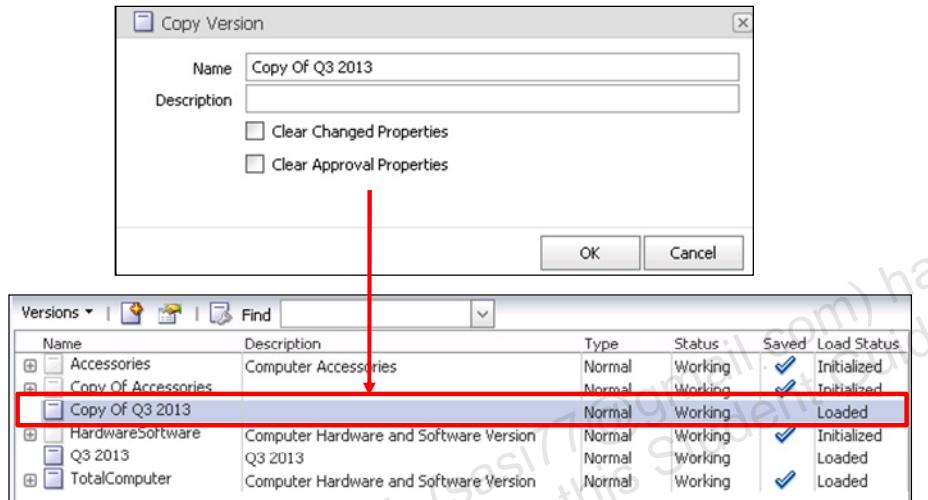
Version Status	Description
Working	End users can manage and change the data within the version within the limitations of configured constraints and validations. End users are limited by granular security.
Submitted	The version is submitted. Only version owners and data managers can change the data. End users are not allowed to make changes, but they can view the data. This status is used toward the end of a data cycle to allow for verification, corrections, and synchronized publishing of all changes.
Finalized	No one can change the data within the version. This status is used to create a “gold copy” of all data structures and properties as they are at the close of a life cycle period.
Expired	Only data managers and version owners can view the version and its data. This status is preserved for comparison and historical purposes. An attempt to publish (export) from an expired version generates a warning that the data may be out of date.

To change version status:

1. Select a version.
2. Select **Versions** and then **Properties**.
Properties for the selected version are displayed on the Properties tab.
3. Set the Version Status property to **Working**, **Submitted**, **Finalized**, or **Expired**.
4. Click **Save**.

Copying Versions

Create a separate version for what-if scenarios, future reorganizations, or future planning.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create versions by copying existing versions. The version copy is independent of other versions.

You typically copy versions to support the following business scenarios:

- Create a new working version for the next time period after finalizing the version for the previous time period.
- Create a separate version for what-if scenarios, future reorganizations, or future planning.
- Roll back invalid or problematic changes made to a version by copying an As-Of version of the normal version before the invalid changes were introduced.

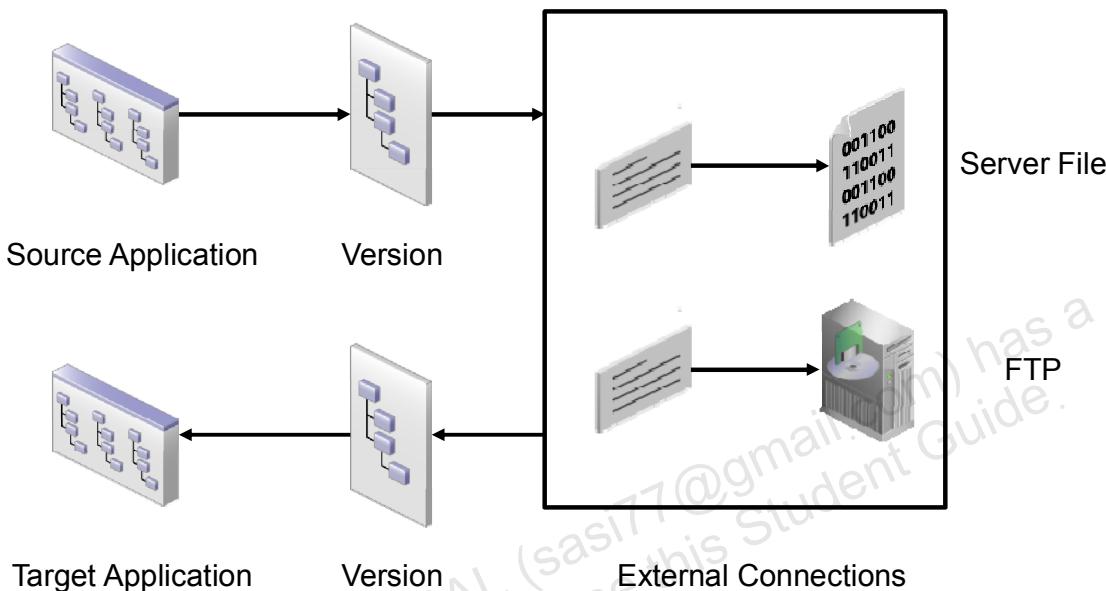
A version copy creates a normal, working version. Hierarchies, nodes, and properties are copied to the new version as well as access and validation assignments. Baseline versions, As-Of versions, and transaction history are not copied. If Change Tracking or Change Approval system preferences are enabled in the version that you want to copy, you can clear these settings in the version copy.

To copy versions:

1. In the versions list, right-click a version, and select **Copy**.
The Copy Version dialog box is displayed.
2. Enter a name and an optional description for the version.
3. **Optional:** Perform one or more of the following actions to clear Change Tracking and Change Approval system preferences:
 - Select **Clear Changed Properties**.
 - Select **Clear Approval Properties**.
4. Click **OK**.
5. Right-click the version, and select **Save**.

The version is saved, and a check mark is displayed in the Saved column.

Backing Up and Restoring Versions



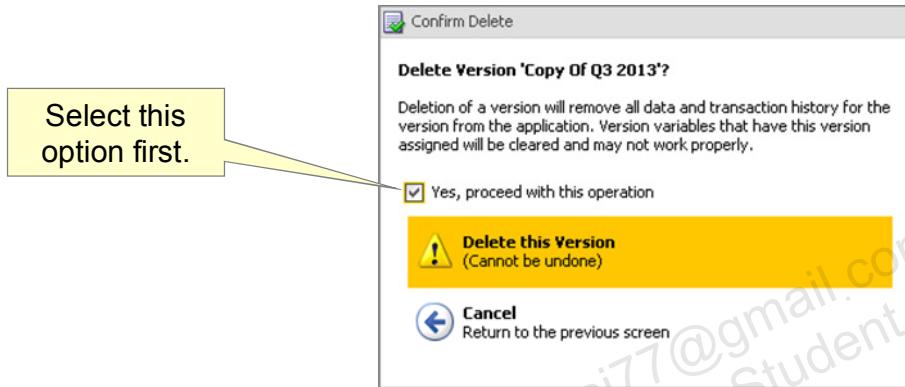
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can archive versions to a file system or migrate to another application using a backup-and-restore procedure. You can select versions and include them in a backup file that is written to a previously defined external connection. The backup file may later be restored to the same or a different Data Relationship Management application. During the restore process, Data Relationship Management identifies whether any incompatibilities exist between the properties stored in the file and the property definitions to which they will be loaded in the target application. Restore From File Warning transactions are logged for property incompatibilities encountered during the restore process.

Deleting Versions

The deletion of a normal version completely removes all data and transaction history for the version.



Select this option first.

ORACLE

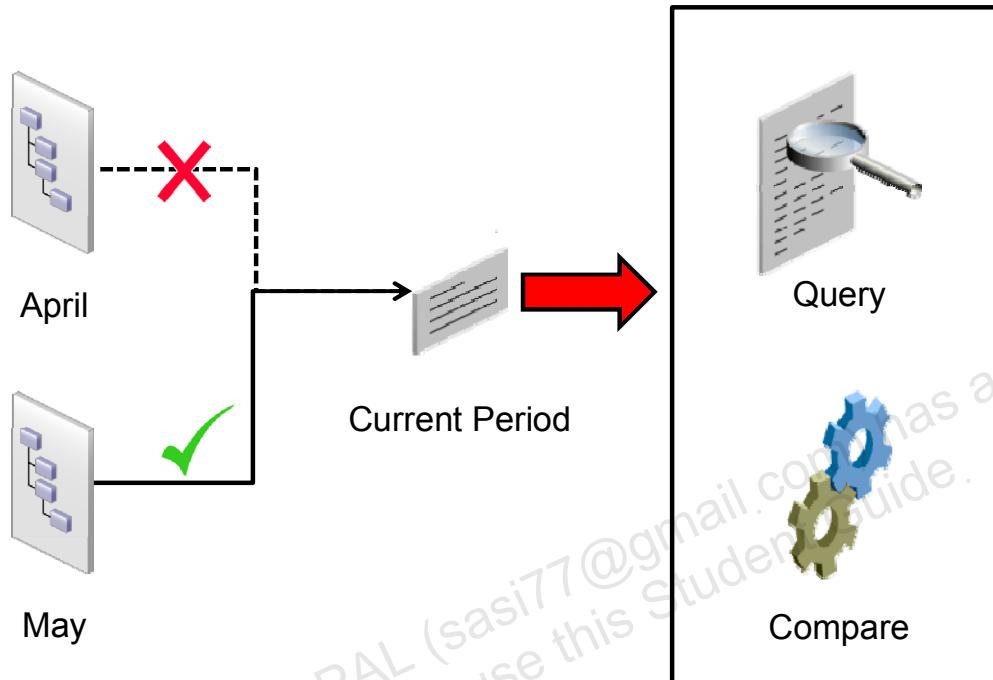
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can delete normal versions that are expired or that were created for what-if scenarios that never took place. The deletion of a normal version completely removes all data and transaction history for the version. Baseline versions and As-Of versions are also removed for the normal version when it is deleted.

To delete versions:

1. In the version list, right-click a version, and select **Delete**.
 - The Confirm Delete dialog box is displayed.
2. Select **Yes, proceed with this operation**.
3. Select **Delete this version**.
 - The version is deleted.

Version Variables



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Version variables allow dynamic references to versions, which are created, copied, and deleted. For example, a version variable can represent a period in time (current month, previous month), a business event or scenario (reorganization, plan, forecast, test), or a recurring set of data to be worked with (import new accounts from GL). You create a version variable, assign a version to it, and then you can use the variable in queries, compares, imports, blenders, and exports. You can also use version variables to quickly find a version in a large list of versions. In the slide example, the Current Period version variable used by a query and a compare originally pointed to the April version. When the Current Period variable is reset to point to the May version, both the query and compare will be redirected to use the May version instead.

To create version variables:

1. In the Browse task group, select **Versions**, then **New**, and then **Variable**.
2. Enter a name and description for the version variable and select an object access level (user, system, standard, or custom object access group). User variables are managed by users for their own personal use. Standard variables are managed by Data Managers for public use by all users. System variables are managed by Application Administrators for system integration purposes.
3. Click **OK**.

To assign versions to version variables:

- In the versions list, right-click a version, select **Set Variables**, and then select a version variable from the list.

The version is assigned to the selected version variable, and a check mark is displayed next to the version variable.

Note: A version may be assigned to multiple variables, but any variable may have only one version assigned.

To delete version variables:

1. In the Browse task group on the toolbar, click the Version Variables button ().
The Version Variables dialog box is displayed.
2. Select the row of the variable that you want to delete and click the Delete icon () in the Action column.
The Delete Version Variable dialog box is displayed.
3. Click **Delete the Variable** to confirm the deletion.
4. Click **Close** to close the Version Variables dialog box.

Viewing and Editing Version Properties

Version properties:

- Store information about versions
- Are grouped into property categories
- Are always global

The screenshot shows the 'Properties' tab of a Data Relationship Management application. The title bar says 'Properties' and 'Relationships'. The main area displays a table of properties for an object named 'TotalComputer'. The table has columns for 'Label', 'Value', and 'Status'. The properties listed are: Version ID (123), Version Name (TotalComputer), Version Description (Computer Hardware and Sol), Version Status (Working), Copy Time (7/17/2013 3:56:55 AM), Version Owner (ADMIN), Legacy Name (Accessories), and Domains in Use (Acquired Products). A 'Category' dropdown menu is set to 'System'.

Label	Value	Status
Version ID	123	
Version Name	TotalComputer	
Version Description	Computer Hardware and Sol	
Version Status	Working	
Copy Time	7/17/2013 3:56:55 AM	
Version Owner	ADMIN	
Legacy Name	Accessories	
Domains in Use	Acquired Products	



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Properties are the main data elements of a Data Relationship Management application. They store information about Data Relationship Management objects. Some property values are read-only; you can edit others. For example, you can change a version's description, but you cannot change its ID or name.

To view and edit properties for versions:

1. Select a version.
2. On the toolbar, click the Version Properties button ().

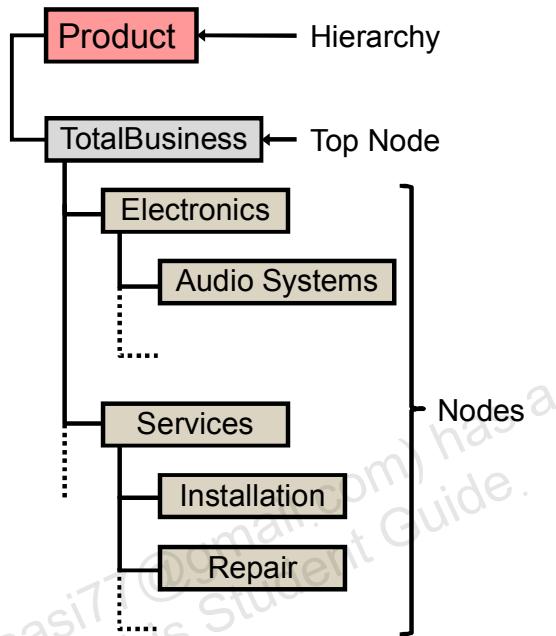
The Properties tab is displayed.

3. In the Category drop-down list, select a property category (for example, System).
4. View read-only property values.
5. Edit property values:
 - a. Select a property.
The Value box for the property becomes editable.
 - b. In the Value box, enter or select a property value.
 - c. At the bottom of the Properties tab, click **Save**.

Working with Hierarchies

Hierarchies:

- Define structural relationships among data (organization structure, accounts, geographic)
- Store alternate rollups
- Are uniquely defined by their top node



ORACLE

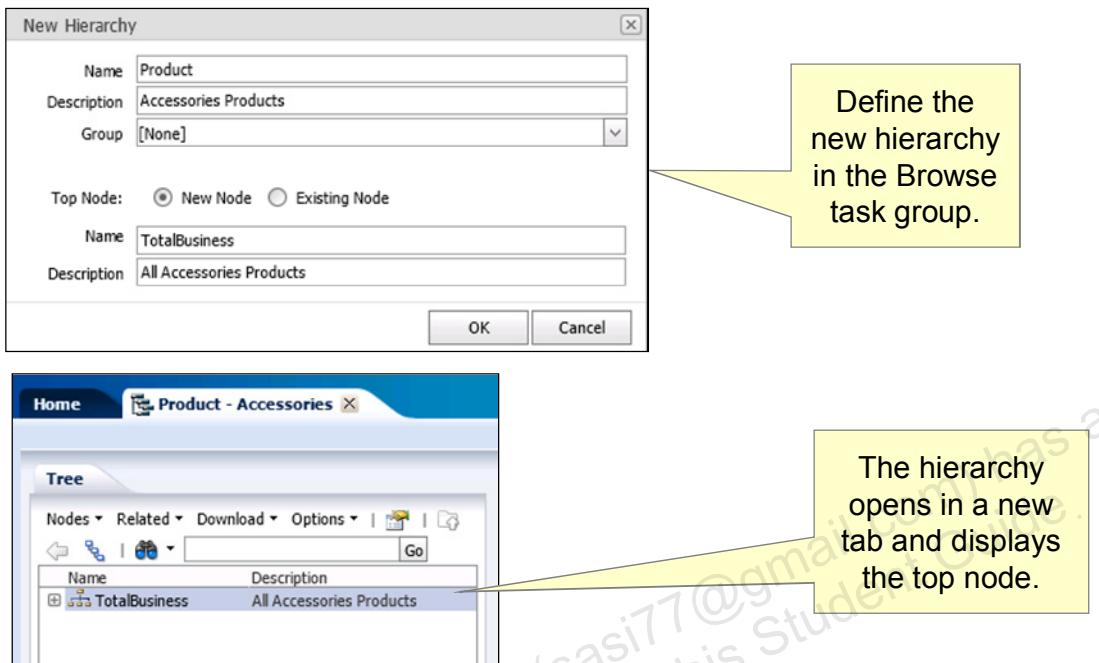
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A Data Relationship Management version can contain one or more hierarchies. A hierarchy represents a business view or reporting structure that groups and organizes a set of related data elements called nodes.

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

In this topic, you learn how to create hierarchies, open and delete hierarchies, and edit hierarchy properties.

Creating, Opening, and Deleting Hierarchies



Caution: You cannot restore a deleted hierarchy.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Creator and Data Manager users can create new hierarchies in loaded versions. When you create a hierarchy, the hierarchy is assigned a top node. The top node can be a new node or an existing node in the version. If an existing parent node is selected as the top node, the node and its descendants in the version are included in the hierarchy.

After creating a hierarchy, you must open it if you want to work with its structure. You can open one or more hierarchies, but you can focus on only one at a time.

When you delete a hierarchy, only its structure is deleted from the database. The hierarchy nodes that are not a part of another hierarchy become orphan nodes at the version level. You can reassign the orphan nodes to another hierarchy within the version, or you can delete the orphan nodes from the version.

Tip: You cannot restore a deleted hierarchy.

To create hierarchies:

1. In the Browse task group, select a version.
2. On the Hierarchies tab, select **Hierarchies**, then **New**, and then **Hierarchy**.
3. Enter a name and description.
4. **Optional:** In the Group drop-down list, select a hierarchy group.
5. Perform one of the following actions:
 - Define a new top node.
 - Select a top node from another hierarchy or from the orphan list.
6. Click **OK**.
 - When the hierarchy tree refreshes, the new hierarchy is displayed.

To open hierarchies:

- On the Hierarchies tab, perform one of the following actions:
 - Double-click a hierarchy.
 - On the toolbar, click the Open Selected button ().
 - Select **Hierarchies**, and then **Open**.

The hierarchy is displayed on a new tab. The nodes and structure of the hierarchy are displayed in a tree. The top node and its children are expanded.

To delete hierarchies:

1. On the Hierarchies tab, right-click a hierarchy, and select **Delete**.
The Confirm Delete dialog box is displayed.
2. Select **Delete this item**.
The hierarchy is deleted.

Viewing and Editing Hierarchy Properties

Hierarchy properties:

- Store information about hierarchies
- Are grouped into property categories
- Are always local

The screenshot shows the 'Properties' tab of the Oracle Data Relationship Management interface. The 'Category' dropdown is set to 'System'. The table lists the following properties:

Label	Value
Hierarchy ID	156
Hierarchy Name	Activity
Hierarchy Description	
Top Node ID	701
Top Node	
Hierarchy Group	Financial Management
Hierarchy Sort Order	0
Hierarchy Owner	ADMIN

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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.

To view and edit properties for hierarchy:

1. Select a hierarchy.
2. On the toolbar, click the Hierarchy Properties button ().
3. In the Category drop-down list, select a property category (for example, System).
4. View read-only property values.
5. Edit property values:
 - a. Select a property.
 - b. In the Value box, enter or select a property value.
 - c. At the bottom of the Properties tab, click **Save**.

Implementation Considerations

- Group related dimensions into the same versions.
- Organize unrelated dimensions into separate versions.
- Manage hierarchies for different time periods or business scenarios in separate versions.
- Maintain hierarchies with common nodes in the same version.
- Combine hierarchies with similar structures into the same hierarchy.
- Determine a naming convention for versions and hierarchies.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consider the following when implementing versions and hierarchies:

- Group related dimensions in the same versions to simplify cross-reference ability over time.
- Organize unrelated dimensions in separate versions to enable isolation and independence of data values.
- Manage hierarchies for different time periods or business scenarios in separate versions to prevent overlap and ensure accuracy of information.
- Maintain hierarchies with common nodes in the same version to synchronize changes to shared relationships and property values.
- Combine hierarchies with similar structures in the same hierarchy for easy management.
- Determine a naming convention for versions and hierarchies based on the dimensions being managed and how you want to resolve naming conflicts.

Summary

In this lesson, you should have learned how to:

- Create, save, change statuses of, copy, back up, restore, and delete versions
- Create version variables and assign them to versions
- Create, open, and delete hierarchies
- View and edit properties for versions and hierarchies
- Describe implementation considerations for versions and hierarchies



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

When a version status is set to Submitted, end users can view the data, but only version owners and data managers can change the data.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

You can assign a version to more than one version variable.

- a. True
- b. False



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Working with Nodes

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

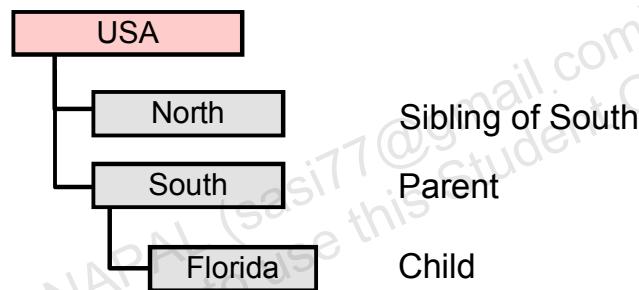
- Add, insert, delete, remove, annul, deactivate, reactivate, copy, move, and order nodes
- View orphan nodes
- Create domains and assign nodes to domains
- Create shared nodes
- View, edit, and copy properties for nodes
- Describe implementation considerations for nodes



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

About Nodes

- Nodes:
 - Are the hierarchy building blocks
 - Use familial relationship terms
- Node-level operations:
 - Create nodes (add, insert)
 - Remove nodes (delete, remove, annul, deactivate)
 - Move nodes



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Nodes are the building blocks of hierarchies. You maintain hierarchy structure by performing node-level operations, such as creating, inserting, moving, merging, sorting, removing, and deleting nodes. You can add nodes to hierarchies by defining parent-child relationships between them, as in the following examples:

- A node is a *child* if it is directly below another node.
- A node is a *parent* if it is directly above another node.
- *Descendants* are all nodes below a given node.
- *Ancestors* are all nodes between a given node and the top of a hierarchy.

All nodes that are descendants of a hierarchy's top node are included within the hierarchy. A node can exist in multiple hierarchies and always has the same descendants in every hierarchy in the same version to ensure consistency across hierarchies. A node can have the same or different parents in each hierarchy. A parent node can have different children only in different versions.

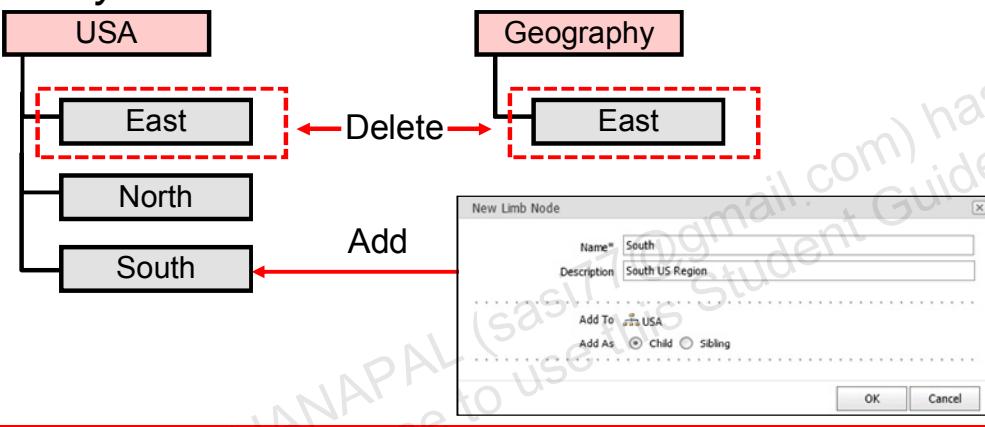
Adding and Deleting Nodes

Adding nodes:

- Node names must be unique within a version.
- The original instance of a node is referred to as the *primary node*.

Deleting nodes:

- Removes the node from all hierarchies
- Applies only to nodes with no descendants



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Adding Nodes

After creating hierarchies, you begin the process of populating them with nodes. One way to populate a hierarchy is to manually add nodes to it. Every node must be uniquely identified by its name within a version. Unless it is a shared node, a node can exist only in one location in a hierarchy (under a single parent). The original instance of a node is referred to as the *primary node*.

During creation, you specify whether the node is a *limb* or a *leaf*. Limb nodes can have children, whereas leaf nodes cannot. You also specify whether the node is to be a child of the currently selected node or its sibling (at the same level).

In the slide example, a new node named South is created as a limb and as a child under the USA node.

To add limbs:

1. Open the hierarchy in which you want to create the node.
2. In the hierarchy tree, select a node.
3. In the Nodes drop-down list, select **New**, and then select **Limb**.

The New Limb Node dialog box is displayed.

4. Enter a name and description.
5. Next to Add As, select **Child** or **Sibling** to add the node as a child or sibling to the selected node.
6. Click **OK**.

To add leafs:

1. Open the hierarchy in which you want to create the node.
2. In the hierarchy tree, select a node.
3. In the Nodes drop-down list, select **New**, and then select **Leaf**.
The New Leaf Node dialog box is displayed.
4. Enter a name and description.
5. Next to Add As, select **Child** or **Sibling** to add the node as a child or sibling to the selected node.
6. Click **OK**.

Deleting Nodes

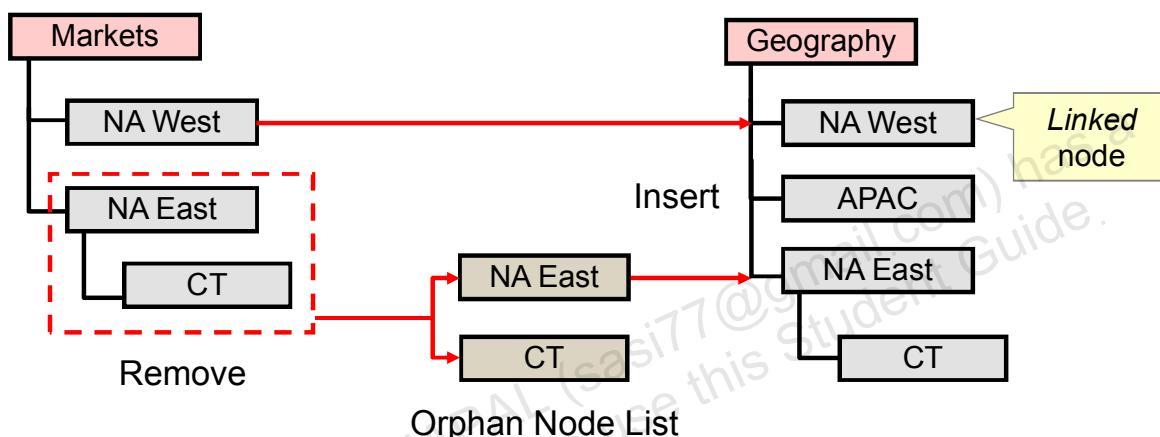
Deleting a node removes the node from all parents in all hierarchies and also deletes the node from the version. Deleting applies only to nodes without descendants. In the slide example, the East node is deleted from both the USA and Geography hierarchies.

To delete nodes:

1. Right-click a node, and select **Delete**.
The Delete Node dialog box is displayed.
2. Select **Delete node** to delete the node from all hierarchies permanently.

Removing and Inserting Nodes

- Removed nodes become *orphan nodes* if the removed node does not have a parent in another hierarchy after the removal.
- You can insert a node from another hierarchy or from the orphan node list.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Removing Nodes

Removing a node removes the node from the current hierarchy and any other hierarchy in which it has the same parent. The node is not deleted from the version so that it can exist in other hierarchies. Removed nodes become *orphan nodes* if the removed node does not have a parent in another hierarchy after the removal. If you remove a parent node with children, the entire branch of the hierarchy (the parent and its descendants) is orphaned. In the slide example, the NA East and CT nodes are removed from the Markets hierarchy and become orphan nodes.

To remove nodes:

- Right-click a node, and select **Remove**.
The Remove Node dialog box is displayed.
- Select **Remove node**.

Inserting Nodes

You can insert nodes from other hierarchies or from the orphan node list. If the source node exists in another hierarchy, a *linked node* is created in the target hierarchy. If the source node is from the orphan node list, a primary node is created in the target hierarchy.

In the slide example, the NA West node from the Markets hierarchy is inserted into the Geography hierarchy; therefore, the NA West node in the Geography hierarchy is considered a linked node. The NA East and CT nodes are inserted from the orphan node list as primary nodes.

To insert nodes:

1. Open the hierarchy in which you want to insert the node.
2. In the hierarchy tree, select a node.
3. In the Nodes drop-down list, select **Insert**.
The Insert Node dialog box is displayed.
4. Select **Existing Hierarchy** or **Orphan Node List** for the location of the node.
5. Perform one of the following actions:
 - If you selected Existing Hierarchy in step 4, select a hierarchy and node, and select whether to insert the node as a child or sibling of the selected node.
 - If you selected Orphan Node List in step 4, select an orphan node in the list, and select whether to insert the node as a child or sibling of the selected node.
6. Click **OK**.

Viewing Orphan Nodes

- Orphans are nodes within a version that are not located in any hierarchy.
- Orphans can result from a node being removed from a hierarchy or as the result of an import process.
- You can insert orphan nodes into a hierarchy or delete them.

Select	Name	Description	# Descendants
<input type="checkbox"/>	China	Country in Asia	0
<input type="checkbox"/>	TotalEntities	United States	17
<input type="checkbox"/>	UnitedStatesofAmerica	United States	16

It is possible for an orphan node to have descendants that are not orphans.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

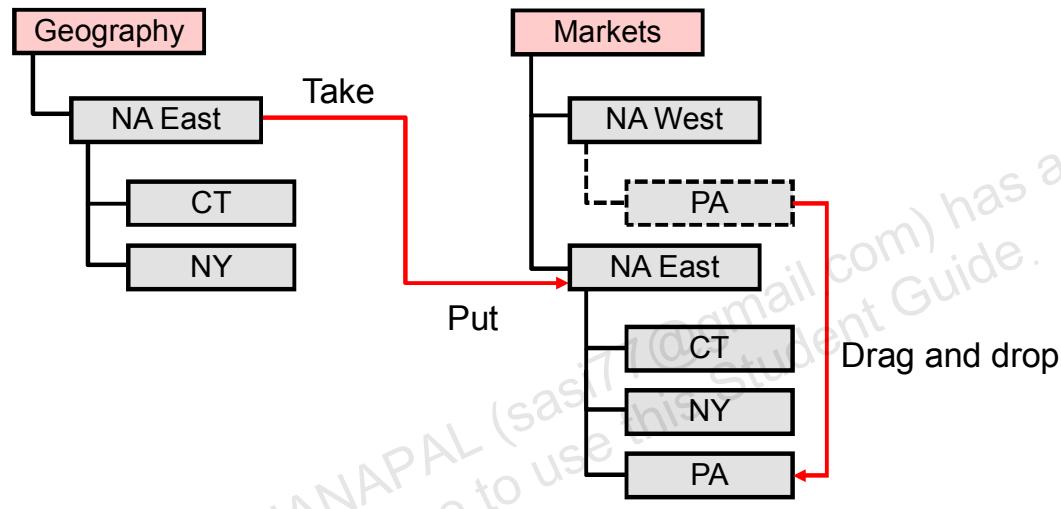
It is possible for an orphan node to have descendants that are not orphans. To avoid this situation, review the list of orphan nodes regularly.

To view orphan nodes:

1. In the Browse task group, click the **Orphans** tab.
For each orphan node, the number of descendants is displayed.
2. **Optional:** Select **Related** and then **Descendants** or **Children** to view node relationships.

Moving Nodes

- You can move nodes under valid limb nodes but not under leaf nodes.
- You can move nodes within the same hierarchy or to another hierarchy.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can move nodes under valid limb nodes but not under leaf nodes. You can move nodes within the same hierarchy or to another 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 another hierarchy is identical to inserting an existing node. This action may be unsuccessful for either of the following reasons:
 - A node with the same name already exists in the destination hierarchy.
 - A descendant of the moved node already exists in the destination hierarchy.

To move nodes:

1. Perform one of the following actions:
 - Drag a node to a new parent or sibling node.
 - Right-click a node, and select **Take/Copy Properties**. Right-click a target node, and select **Put**.
The Choose Node Destination dialog box is displayed.
2. Select **Put as Child** or **Put as Sibling**.
3. **Optional:** Select **Use this setting as the default**.
4. Click **OK**.

Ordering Nodes

Use the Core.SortOrder property to configure a custom sort order of nodes within a hierarchy.

Standard Hierarchy Sort (hierarchy-level property) = Core.SortOrder

Name	Description	Sort Order
100-30	Digital Speakers	0
100-10	Stereo Speaker System	1
100-20	Surround Sound Speakers	2
200	Mouse Devices	0
300	Microphones	0
400	Webcams	0
500	Drives & Storage	0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Data Relationship Management, nodes are sorted according to the following default rules:

- Limbs are sorted before leaves. This sort is controlled by the `SortLimbsFirst` system preference.
- Nodes are sorted by name in ascending order.

In some cases, nodes should be ordered in a particular way to support reporting requirements. You can apply a custom order to nodes in a hierarchy by using a sort order property. In the slide example, the Core.SortOrder property sorts the child nodes in the following order: 100-30, 100-10, and then 100-20.

To sort hierarchies based on a node-level sort order property:

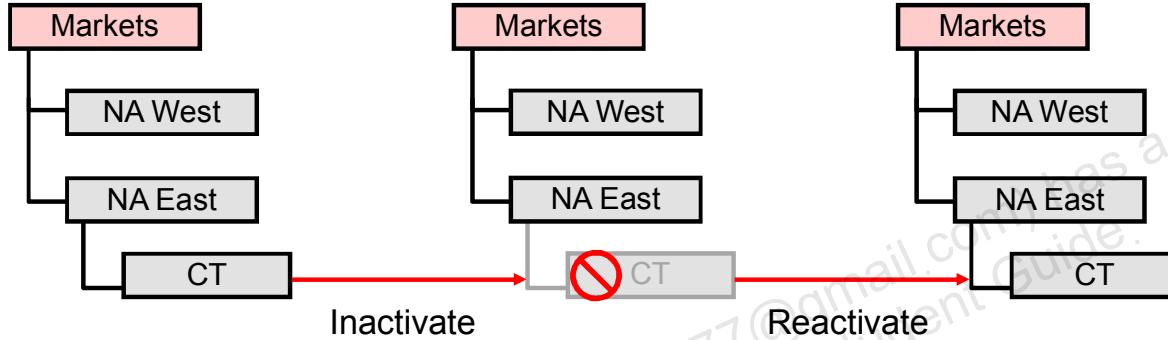
1. Set the value of the **Standard Hierarchy Sort** hierarchy-level property to point to the appropriate node-level property that contains the sort order.
2. Move the nodes in the hierarchy tree to create a custom sort order.
Note: Moving a node within a hierarchy and under the same parent as the target sibling is considered a reorder operation.

To disable sort orders:

- Remove the value of the hierarchy-level property that controls the sort order.

Inactivating and Reactivating Nodes

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



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Inactivating flags the node as inactive so that it can be filtered from browsing or from an export. You can deactivate leaf nodes or limb nodes that have no active children. When you deactivate a node it is deactivated in all hierarchies. You can reactivate the inactive node if needed.

To deactivate nodes:

- Right-click a node, and select **Inactivate**.

When the hierarchy tree refreshes, the node is grayed out and displays an inactive status icon.

To filter inactive nodes when browsing a hierarchy:

- From the Options menu on the hierarchy tree, select **Hide Inactive Nodes**.

When the hierarchy tree refreshes, inactive nodes are not displayed in the tree.

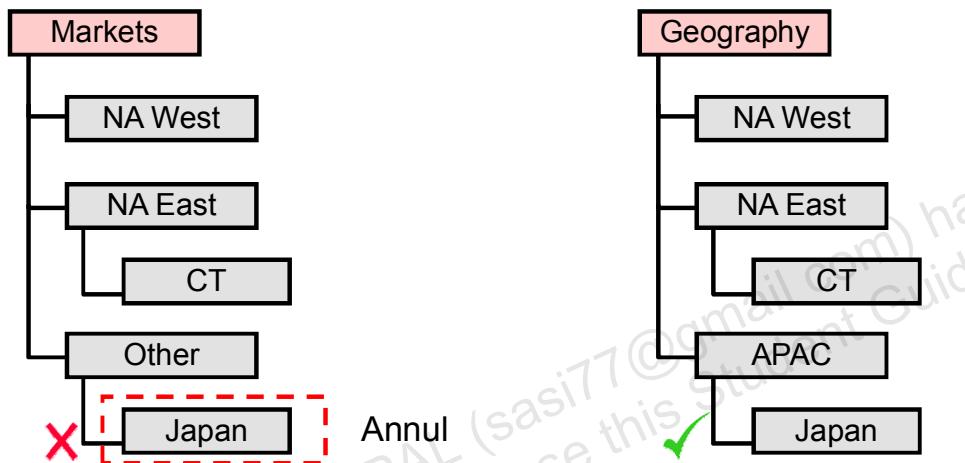
To reactivate nodes:

- Right-click an inactive node, and select **Reactivate**.

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

Annulling Nodes

Annulling removes the node from all hierarchies in which it has the same parent, and then deletes the node if the node becomes an orphan.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Annulling removes the node from all hierarchies in which it has the same parent, and then deletes the node if the node becomes an orphan. If a parent node is selected, the Annul operation is performed for the node and its descendants. In the slide example, the Japan node is annulled and, therefore, removed from the Other node in the Markets hierarchy. The Japan node in the Geography hierarchy, however, remains because it has a different parent (APAC).

To annul nodes:

1. Right-click a node, and select **Annul** or **Annul All Below** (for a parent node).

Note: The Annul All Below operation is applied to the node and its descendants.

The Annul Node or Annul All Below Node dialog box is displayed.

2. Select **Annul node** or **Annul All Below**. 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.

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

Copying Nodes and Properties

- **Take/Copy Properties** and **Paste Properties** copy a node's properties to another node.
- **Take/Copy Properties** and **Put** create linked nodes when used across hierarchies.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the following options to copy nodes and properties:

- **Take/Copy Properties** and **Paste Properties** copy a node's properties to another node.
- **Take/Copy Properties** and **Put** create linked nodes when used across hierarchies.

To copy node properties to other nodes:

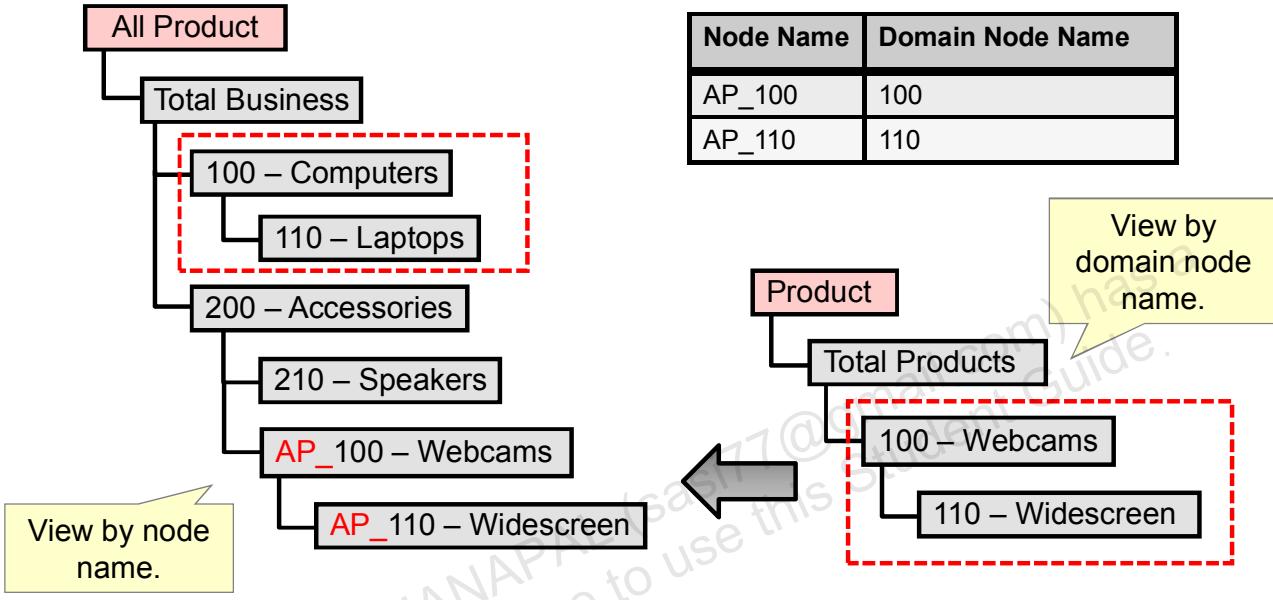
1. In a hierarchy tree, right-click the node that contains the properties that you want to copy, and select **Take/Copy Properties**.
2. Right-click another node, and select **Paste Properties**.
The Paste Properties dialog box is displayed.
3. Select one of the following options:
 - **Overridden Properties Only**
 - **All Properties**
 - **Select Properties** (If you select this option, the dialog box is expanded. Select properties, and click **OK**.)

To copy nodes to other hierarchies:

1. In a hierarchy tree, right-click a node, and select **Take/Copy Properties**.
2. In another hierarchy tree, right-click a node, and select **Put**.
The Choose Node Destination dialog box is displayed.
3. Select **Put as Child** or **Put as Sibling**.
4. **Optional:** Select **Use this setting as the default** to avoid the Choose Node Destination dialog box in the future.
Note: If needed, you can bypass this default setting: Select **Options**, then **Put As**, and then **Child of Target Node**, **Sibling of Target Node**, or **Prompt for Destination**.
5. Click **OK**.

About Domains

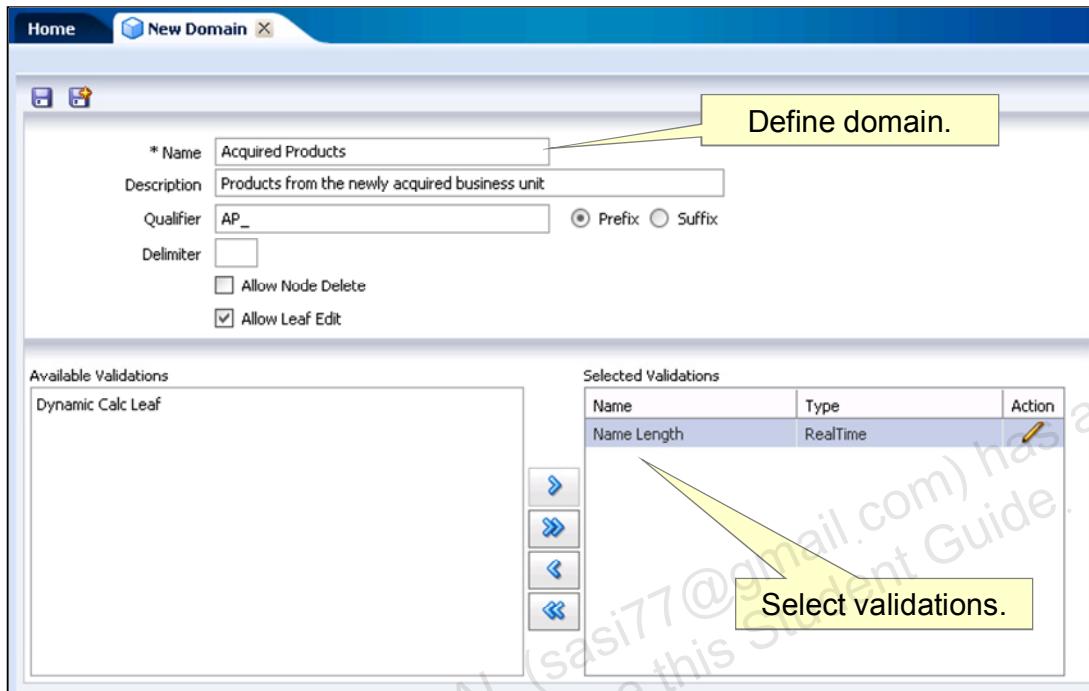
Domains are used to manage referential integrity for multiple sets of nodes from different sources.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Creating Domains



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To create a domain:

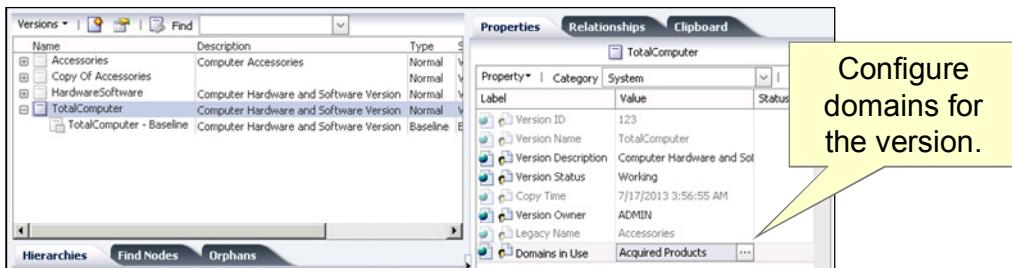
1. On the Home tab, select the **Administer** task group.
2. In the New drop-down list, select **Domain**.
The New Domain tab is displayed
3. Enter the name and optional description for the domain.
4. **Optional.** In the Qualifier box, enter text to be used for fully qualifying a node name. No two domains can use the same qualifier text. Select **Prefix** or **Suffix** to denote the location of the qualifier.
Note: After a domain has nodes assigned to it, the qualifier text cannot be changed.
5. **Optional.** In the Delimiter box, enter a single character used to separate the domain-qualifier text from the node name.
Note: After a domain has nodes assigned to it, the delimiter cannot be changed.
6. Select **Allow Node Delete** if you want to give users the ability to delete nodes from the version.
7. Select **Allow Leaf Edit** if you want to give users the ability to change the leaf system property value for nodes in the domain.

8. From the Available Validations list, select the node-level validations to be enforced for members of the domain, and move them to the Selected Validations list.
Note: Domain-level validation assignments override assignment values for the same validation that were set at the node or inherited from an ancestor node, hierarchy, or version level assignment.
9. On the toolbar, click the Save button ().

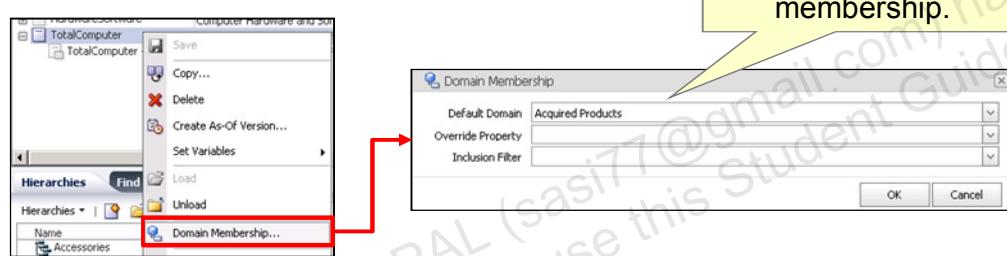
The New Domain tab is renamed to the domain name.

Assigning Nodes to Domains

1. Assign domains to a version.



2. Assign nodes to the domain.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Assigning nodes to domains is a two-step process. You first need to assign domains to a version, and then you configure domain membership for the version nodes.

To assign domains to a version:

1. In the Browse task group, right-click a version and select **Properties**.
The Properties tab is displayed.
2. In the Category drop-down list, select **System**.
3. Select the **Domain in Use** property.
The Value box for the property becomes editable.
4. Click the ellipsis button (**...**) in the Value column.
The Select Values dialog box is displayed.
5. From the Available list, select domains to assign to the version and then move them to the Selected list.
Note: You can use the up and down arrows in the Selected list to determine the order in which the domains are displayed.
6. Click **OK**.

To assign nodes for the domain:

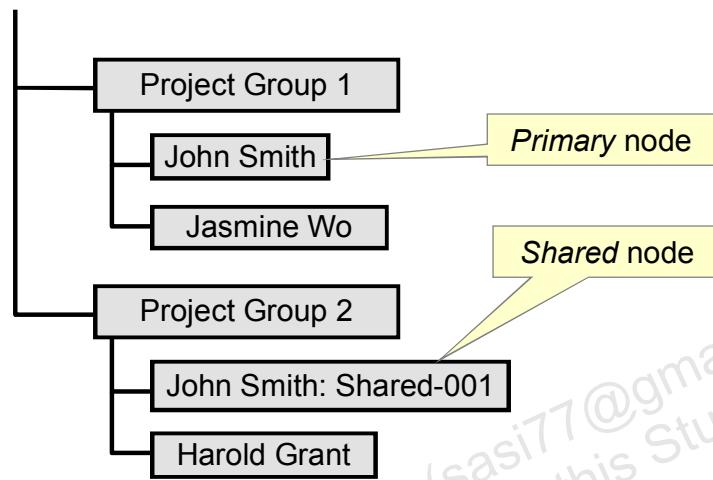
1. In the Browse task group, right-click a version and select **Domain Membership**.
The Domain Membership dialog box is displayed.
2. Select a domain to assign qualified nodes to by doing the following:
 - a. In the Default Domain drop-down list, select a domain.
 - b. In the Override Property drop-down list, select a property that specifies a domain.
Note: An override property is a designated global property with a domain name value used for domain name qualification and domain assignment.
 - c. If both options above are specified, the default domain is used only when the override property value at the node is blank. If the override property value is invalid, the default domain is not used and an error occurs.
 - d. In the Inclusion Filter drop-down list, select a filter to select qualified nodes.
3. Click **OK**.

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

Nodes in a version can also be assigned to a domain during the import process in bulk or at the time of adding a node.

Creating Shared Nodes

You create a shared node by adding another instance of a primary node to the same hierarchy.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A *shared node* is a node that has multiple instances in the same hierarchy. It is linked to a *primary node*. You create shared nodes by using either the new node or insert node method. Shared node names are appended with a unique identifier so that all node names remain unique for referential integrity purposes. Shared nodes share the same global property values with the primary node in the version but can have different local property values in the same hierarchy as the primary node. In the slide example, the “John Smith:Shared-001” node is the shared node and the John Smith node is the primary node.

You can also create shared nodes during an import process. This topic is discussed in a later lesson.

Note: The shared nodes feature is disabled by default; therefore, if you want to include share nodes, you must enable it in your application and in each hierarchy where shared nodes are to be allowed.

To enable shared nodes in applications:

1. Enable the SharedNodeMaintenanceEnabled system preference:
 - a. On the Home page, select the **Administer** task group.
 - b. Expand **System Preferences**.
 - c. Double-click the SharedNodeMaintenanceEnabled system preference to open it. The SharedNodeMaintenanceEnabled tab is displayed.
 - d. In the Value drop-down list, select **True**.
 - e. On the toolbar, click the Save button ().
2. In Data Relationship Management Console, restart your application.
3. For any hierarchy in which you want to allow shared nodes:
 - a. On the Home page, select the **Browse** task group.
 - b. On the Hierarchies tab, select a hierarchy.
 - c. On the Properties tab, select the **Enable Shared Nodes** property, select the check box, and click **Save**.

To add shared nodes:

1. Select the node where you want to insert a shared node.
2. Select **Nodes** and then **New**.
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 shared nodes:

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

Viewing and Editing Node Properties

Node properties:

- Store information about nodes
- Are grouped into property categories

The screenshot shows two separate Property grids from the Oracle Data Relationship Management interface. Both grids have tabs at the top: Properties, Relationships, Clipboard, and Shortcuts. The left grid is titled 'Properties' and has a 'System' category selected. It lists properties for a node labeled 'USA'. The right grid is also titled 'Properties' and has a 'Stats' category selected. It lists properties for a node labeled 'USA'.

Properties	
Property	Category
Label	System
Node ID	4
Name	USA
Description	United States
Leaf	False
Leaf Access	Add
Limb Access	Add
Display String	USA - United States
Node URL	

Properties	
Property	Category
	Stats
Added On	5/7/2013 12:39:38 AM
Added By	ADMIN
Last Changed On	5/7/2013 12:39:38 AM
Last Changed By	ADMIN
Node Changed	True
References	Markets
Level	2
Linked	False
Links	
Ancestor List	TotalMarkets
# Links	0
# Children	2
# Leaf Children	0
# Limb Children	2
# Descendants	5
# Leaf Descendants	3
# Limb Descendants	2
# Siblings	1
# Leaf Siblings	0
# Limb Siblings	1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view and edit properties for nodes:

1. Select a node.
2. On the toolbar, click the Node Properties button ().
The Properties tab is displayed.
3. In the Category drop-down list, select a property category (for example, System).
4. View read-only property values.
5. Edit property values:
 - a. Select a property.
The Value box for the property becomes editable.
 - b. In the Value box, enter or select a property value.
 - c. At the bottom of the Properties tab, click **Save**.

Node properties are grouped into property categories. The following table describes the built-in property categories and their properties:

Property Category	Properties
System	Properties related to the identifying characteristics of a node, such as ID, name, and description. The only change that you can make to this category is to assign the read-only flag for individual users. Users with read-only access cannot edit values but can view them. You cannot assign properties to this category.
Shared Info	Identification of primary nodes and shared nodes, list of related shared nodes, and identification of missing primary nodes. This category is displayed only when Shared Nodes is enabled through system preferences.
Stats	Properties that provide change-tracking and statistical information about a node, such as number of children and number of siblings

Implementation Considerations

- Determine a naming convention for nodes.
- Decide whether to use leaf nodes to enforce separate restrictions on dimension members.
- Decide whether to use shared nodes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consider the following when implementing nodes:

- Determine a naming convention for nodes based on the dimensions and systems being managed and how you want to resolve naming conflicts. Domains may be used to handle naming conflicts for commonly named nodes from different dimensions or systems that must reside in the same version(s).
- Decide whether to use leaf nodes to enforce separate restrictions on dimension members where transactional data is associated.
- Identify whether any hierarchies may require multiple instances of the same nodes under different parents. Decide whether these relationships can be managed in separate hierarchies or whether shared nodes must be enabled for specific hierarchies to handle the duplicate instances of the nodes.

Summary

In this lesson, you should have learned how to:

- Add, insert, delete, remove, annul, deactivate, reactivate, copy, move, and order nodes
- View orphan nodes
- Create domains and assign nodes to domains
- Create shared nodes
- View, edit, and copy properties for nodes
- Describe implementation considerations for nodes



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

Which of the following statements about nodes are correct?

- a. A primary node cannot exist more than once in the same hierarchy.
- b. Moving a node within a hierarchy keeps the original node under its parent and makes a copy under the new parent.
- c. A node can be linked to multiple hierarchies.
- d. Shared node names must be unique in a hierarchy.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c, d

Quiz

The Delete option applies only to hierarchy nodes with no descendants.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Defining Properties



ORACLE®

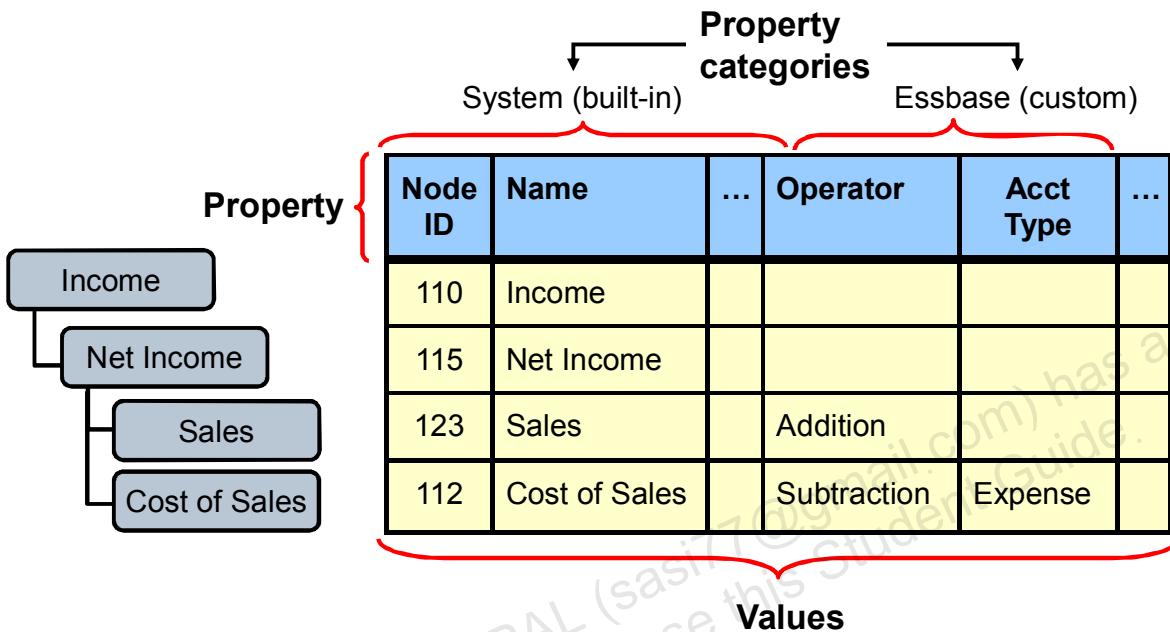
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe custom properties
- Configure defined properties
- Configure list values for properties
- Configure lookup values for properties
- Configure inheriting properties
- Create property categories
- Load metadata with the Migration Utility

Custom Properties



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application administrators can create property definitions to describe nodes based on specific characteristics of the external systems in which they belong and to categorize nodes within hierarchies according to business functions. Each node has the same set of properties; however, a node can have properties without values set.

Application administrators can group properties into property categories and assign user access to them. Each property must belong to at least one category.

In the slide example, the System and Essbase property categories are displayed. The System property category is a built-in category; however, the Essbase category is created by the application administrator. It consists of custom properties such as Operator and Acct Type.

Namespaces

- Namespaces define fully qualified property names in the application.
- System-defined properties use the **Core** namespace.
- User-defined properties use the **Custom** namespace.
- Other namespaces are reserved for use by Data Relationship Management application templates for other Oracle products.

The screenshot shows a configuration dialog for a property. The 'Namespace' dropdown is set to 'Custom'. The 'Name' field contains 'StateProvince'. The 'Label' field contains 'State Province'. The 'Description' field contains 'State or Province'. The 'Property Level' dropdown is set to 'Local Node'. The 'Property Type' dropdown is set to 'Defined'. The 'Data Type' dropdown is set to 'String'. The 'Default Value' field is empty. There are checkboxes for 'Inherited', 'Overridable' (which is checked), 'List', and 'Hidden'. A red box highlights the 'Name' field, and a red arrow points from it to the 'Fully Qualified Name' field, which contains 'Custom.StateProvince'.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management properties use namespaces. Namespaces prevent conflicts between properties with similar names from different origins when used together in the same application. Namespaces are used to segregate predefined properties from those that are created or loaded by application administrators at implementation time.

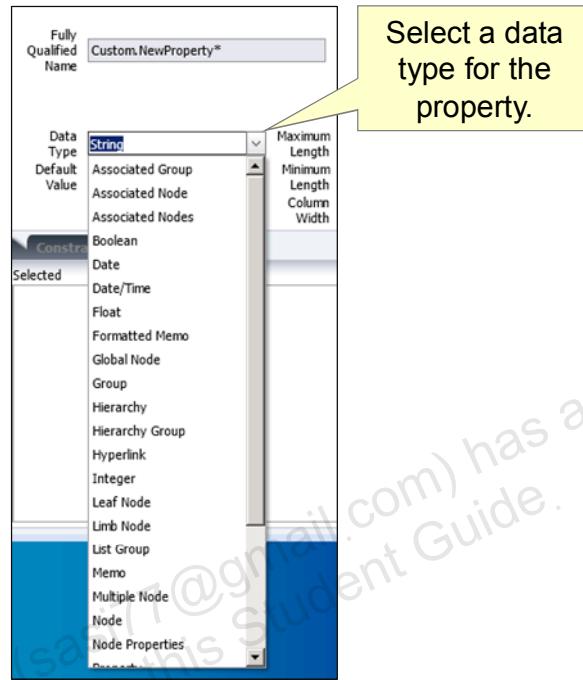
The following namespaces are available by default:

- Core** represents system-defined properties that control the behavior of application features.
- Custom** represents user-defined properties that meet your own requirements of the organization.

Other namespaces are reserved for application templates with integrated products. When you load an application template, namespaces used in the template are added to your application.

Data Types

Data types define the type of values for a property, such as date, integer, or string.



Select a data type for the property.

ORACLE

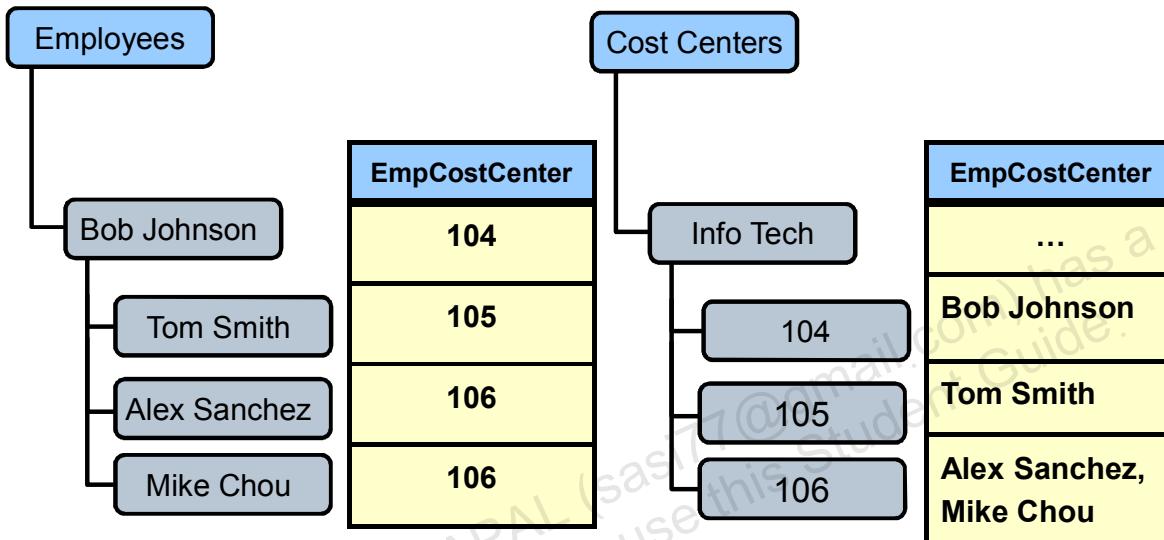
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data types define the type of values for a property. The following table describes commonly used data types:

Data Type	Description
Boolean	True or false
Date	Date
DateTime	Date and time
Float	Floating-point decimal value
Hier	Hierarchy
Integer	Integer value
ListGroup	List of items (One or more items in the list may be selected.)
Memo	Memo field
String	String value

Node Data Types

Create a property with an AscNodes data type to define a bi-directional association between nodes.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use properties to manage non-hierarchical relationships between nodes in the same or different hierarchies. Several data types are available that allow nodes to point to other nodes. Each data type manages a different relationship type. Values for these properties refer to nodes using their names, and in some cases, the hierarchy where the node is located.

An associated node data type, such as AscNodes, enables you to define a relationship between nodes using a property that can be referenced from any node that is in the association. This data type is useful when mapping nodes of different types across hierarchies.

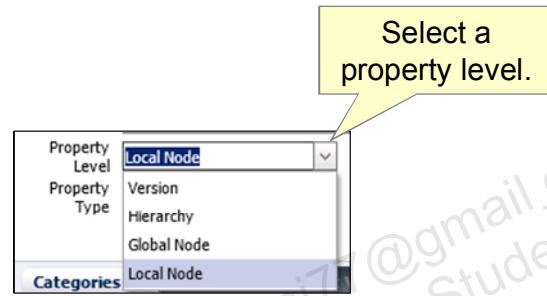
In the slide example, employee Tom Smith is associated with cost center 105. When you view the properties of Tom Smith, the property points to the cost center 105 node. When you view the properties of cost center 105, the property points to employee Tom Smith.

The following table describes node data types:

Node Data Type	Description
AscGroup	Associated node group A property of this type can point to multiple nodes. Those nodes automatically point back to this node and automatically point to each other as well.
AscNode	Associated node A property of this type can point to a single other node. That node automatically points back to this node.
AscNodes	Associated nodes A property of this type can point to multiple nodes. Those nodes automatically point back to this node but not to each other.
GlobalNode	Points to a global node When a value is assigned, it displays the node name only in the value field on the Properties tab.
LeafNode	Points to a leaf node When a value is assigned, it displays the name of the hierarchy and node.
LimbNode	Points to a limb node When a value is assigned, it displays the name of the hierarchy and node.
MultiNode	MultiNode
Node	Points to a hierarchy node When a value is assigned, it displays the name of the hierarchy and node.

Property Levels

- The *property level* defines the scope of the property.
- There are four property levels:
 - Version
 - Hierarchy
 - Global Node
 - Local Node



ORACLE

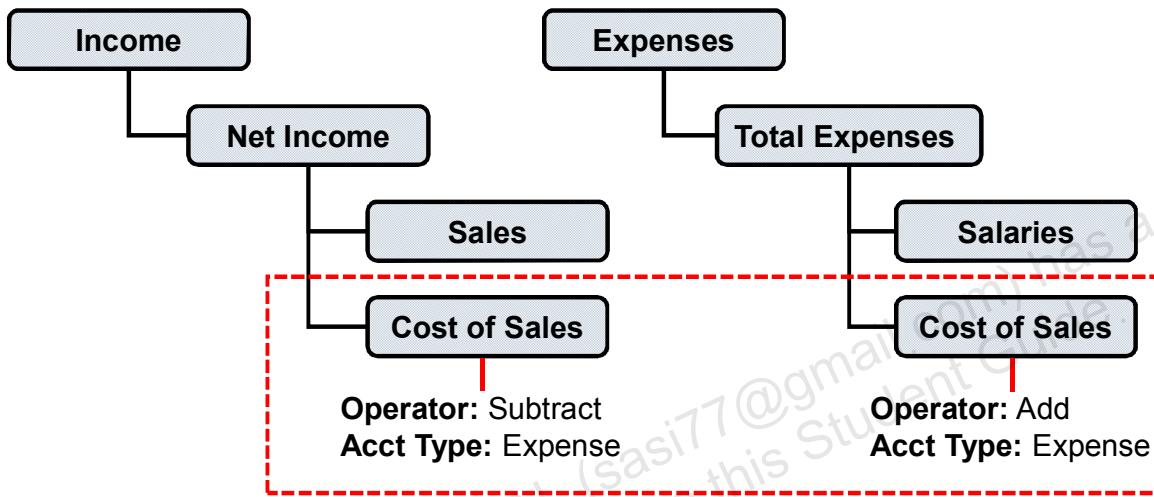
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can define properties at four levels. Most properties are defined at the global node level, which means that the property values are not based on the context and usage of the node (that is, the hierarchy where the node is located).

You can define properties at the version and hierarchy levels to meet organization or process needs. For example, you create a property for hierarchies that identifies the external systems for which the hierarchies are valid. Properties at the version and hierarchy levels are less common than properties at the node level.

Local and Global Properties

- Local properties can have different values in different hierarchies.
- Global properties have the same value in all hierarchies.



Operator is a **local** property. **Acct Type** is a **global** property.

ORACLE

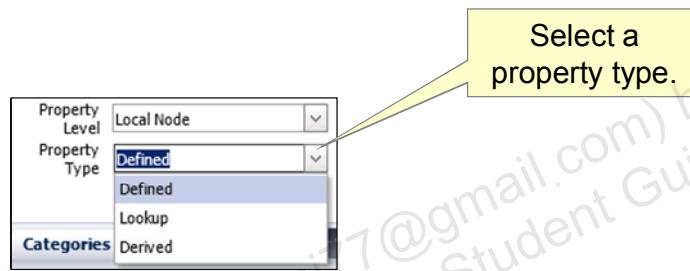
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A local property for a node can have different values in different hierarchies in which the node occurs. A global property for a node has the same value in all hierarchies in which the node occurs. When you update the value for a global property for a node in one hierarchy, that property value is updated in all other hierarchies in which the node occurs.

In the slide example, Operator is a local property and Acct Type is a global property. The Operator property value for the Cost of Sales node is Subtract in the Income hierarchy and Add in the Expenses hierarchy. The Acct Type property value for the Cost of Sales node is Expense in both the Income hierarchy and the Expenses hierarchy.

Property Types

- *Property types* define how values are assigned to a property.
- There are three property types:
 - Defined
 - Lookup
 - Derived



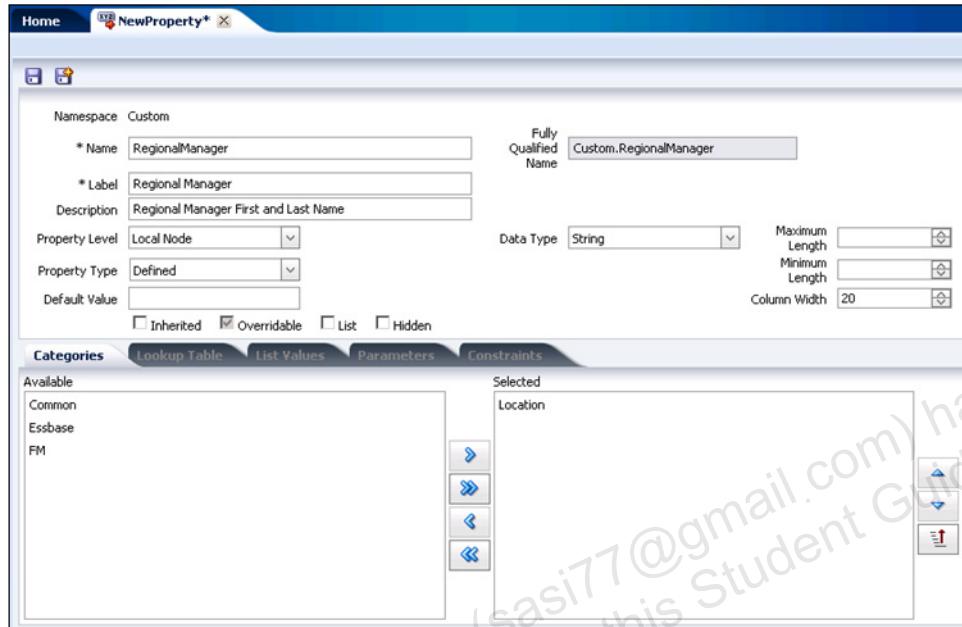
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Property types define how values are assigned to a property. The following table describes the available property types:

Property Type	How Property Values Are Determined
Defined	Users enter values manually or select them from a list defined when the property is configured. You can also define these property values with imports and action scripts.
Lookup	Property values are assigned to the property automatically based on the values of another property and a lookup table. For example, the Country property might be filled in automatically based on the value for the City property.
Derived	Property values are calculated by using a formula or script. Derived properties are read-only by default, but you can set them to allow override values if needed. Derived properties are covered in more detail in a later lesson.

Configuring Defined Properties



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring defined properties is the easiest and fastest method for creating properties. In the slide example, the defined property named RegionalManager has the String data type, is at the local node property level, and has a column width of 20. Setting up the property in this manner enables users to enter Regional Manager information for each node.

To configure defined properties:

1. On the Home tab, select the **Administer** task group.
 2. In the New drop-down list, select **Property Definition**.
The New Property tab is displayed.
 3. Enter a name and an optional description for the property.
 4. In the Label box, enter a label for the property.
- Tip:** The property label is displayed on the Properties tab when you browse versions, hierarchies, and nodes. The default property label is the property name.
5. In the Property Level drop-down list, select a property level: **Version**, **Hierarchy**, **Global Node**, or **Local Node**.
 6. In the Property Type drop-down list, select **Defined**.

7. In the Data Type drop-down list, select a data type.
8. In the Minimum Length and Maximum Length boxes, enter the minimum and maximum property values.

Note: The labels for these parameters depend on the data type selected. For String data types, they are Min Length and Max Length. For Integer data types, they are Min Value and Max Value.

9. In the Default Value box, enter a default value if required.

The default value is used if an end user does not enter a value for a property. It can also represent a constant prefix for the property values.

10. In the Column Width box, enter the width for fixed-width columns.

11. On the Categories tab, perform one or more of the following actions to assign the property to property categories:

- Select items in the Available list, and click the Select button () to move them to the Selected list.
- Click the Select All button () to move all items in the Available list to the Selected list.
- Select items in the Selected list, and click the Remove button () to move them to the Available list.
- Click the Remove All button () to move all items in the Selected list to the Available list.

12. On the toolbar, click the Save button ().

Configuring List Values

You can define a set of valid values for the property.

The screenshot shows the 'New Property' configuration screen. The 'Name' is set to 'Consolidation' and the 'Fully Qualified Name' is 'Custom.Consolidation'. The 'Label' is also 'Consolidation'. The 'Description' is 'Consolidation Options'. The 'Property Level' is 'Local Node' and 'Defined'. The 'Data Type' is 'String', 'Default Value' is 'Addition', and 'Column Width' is 20. The 'List' checkbox is checked, enabling the 'List Values' tab. The 'List Values' tab is selected, showing a list of items: Addition, Subtraction, Multiplication, Division, Percent, and Ignore. The 'Ignore' item is highlighted with a red box. On the right, there is a toolbar with various actions like Add, Remove, and Sort.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can set up a list of valid values for a property to make data entry for users more efficient and accurate. In the slide example, the property named Consolidation has six values: Addition, Subtraction, Multiplication, Division, Percent, and Ignore.

To configure list values:

1. On the New Property tab, select the **List** check box.
The List Values tab is enabled.
2. Select the **List Values** tab.
3. Click **Add**.
4. Enter a value, and click the Update button ().
5. Repeat step 4 until you complete the list values.

Configuring Lookup Values

The value of a lookup property is determined by the value of another property and a lookup table.

The screenshot shows the 'New Property' configuration screen. The 'Name' field is set to 'DefaultCurrency' and the 'Fully Qualified Name' is 'Custom.DefaultCurrency'. The 'Label' is 'Default Currency' and the 'Description' is 'Default Currency'. The 'Property Level' is 'Local Node' and the 'Property Type' is 'Lookup'. The 'Data Type' is 'String' and the 'Lookup Property' is 'Country'. The 'Maximum Length' is 20. Below this, there are tabs for 'Categories', 'Lookup Table' (which is selected), 'List Values', 'Parameters', and 'Constraints'. The 'Lookup Table' tab displays a table with columns 'Lookup Key' and 'Result Value'. The data in the table is:

Lookup Key	Result Value	Action
Canada	CND	
China	CNY	
France	EUR	
Italy	EUR	
South Africa	ZAR	
United Kingdom	EUR	
USA	USD	

An 'Add' button is located at the bottom right of the table.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

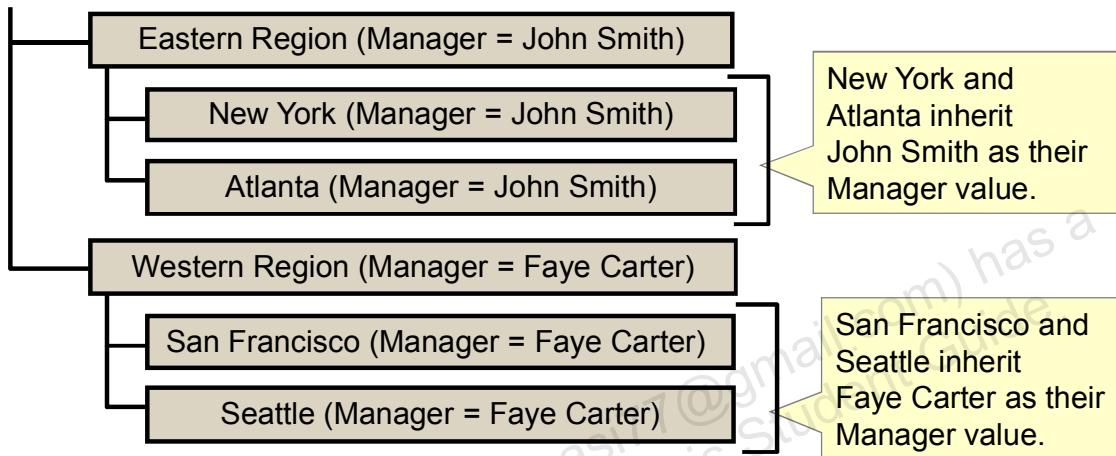
The value of a lookup property is determined by the value of another property and a lookup table. In the slide example, the DefaultCurrency property value is determined by the Country property and its lookup value. For example, if the Country property for a node is equal to USA (USA is referred to as the *lookup key*), the DefaultCurrency property value is equal to USD (USD is referred to as the *result value*) according to the lookup table.

To configure lookup values:

1. On the New Property tab, in the Property Type drop-down list, select **Lookup**.
2. In the Lookup Property drop-down list, select the property that contains the lookup key values.
3. Click the **Lookup Table** tab.
4. Click **Add**.
5. Enter a lookup key and a result value, and click the Update button ().
6. Repeat steps 4 and 5 until you enter all lookup keys.

Configuring Inheriting Properties

Descendant nodes inherit values for a property from an ancestor.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Inheritance allows high-level nodes to share property values with lower nodes in the hierarchy, so that you do not have to store and maintain redundant information. When you define a property as inheriting, the values of the property cascade down to its descendants. Proper use of inheritance enhances the integrity of property values and reduces maintenance requirements.

Note: You can configure inheritance only for properties with a Defined property type.

When determining the value of inheriting properties, apply the following rules:

1. If you directly enter a value at the current node, the entered value is used.
2. If you do not enter a value at the current node, the property value of the closest ancestor that has a stored value is used.
3. If no ancestor has an entered value, the default value is used.

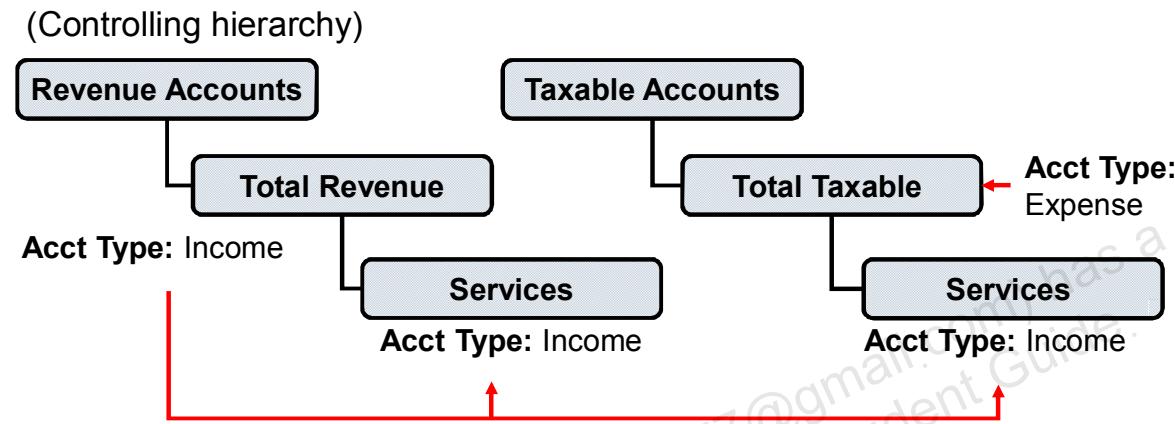
In the slide example, the cities under each region inherit the Manager property value. The Manager value for the western region is Faye Carter. The cities in the western region, San Francisco and Seattle, inherit Faye Carter as their Manager value.

To configure inheriting properties:

- On the New Property tab, select the **Inherited** check box.

Assigning Controlling Hierarchies to Global Inheriting Properties

You must select a controlling hierarchy for properties defined as both global and inheriting.



Acct Type is inherited from the controlling hierarchy (Revenue Accounts), but you can override it in other hierarchies if needed.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Because a node can have different ancestors in different hierarchies, determining the value of global inheriting properties requires additional considerations.

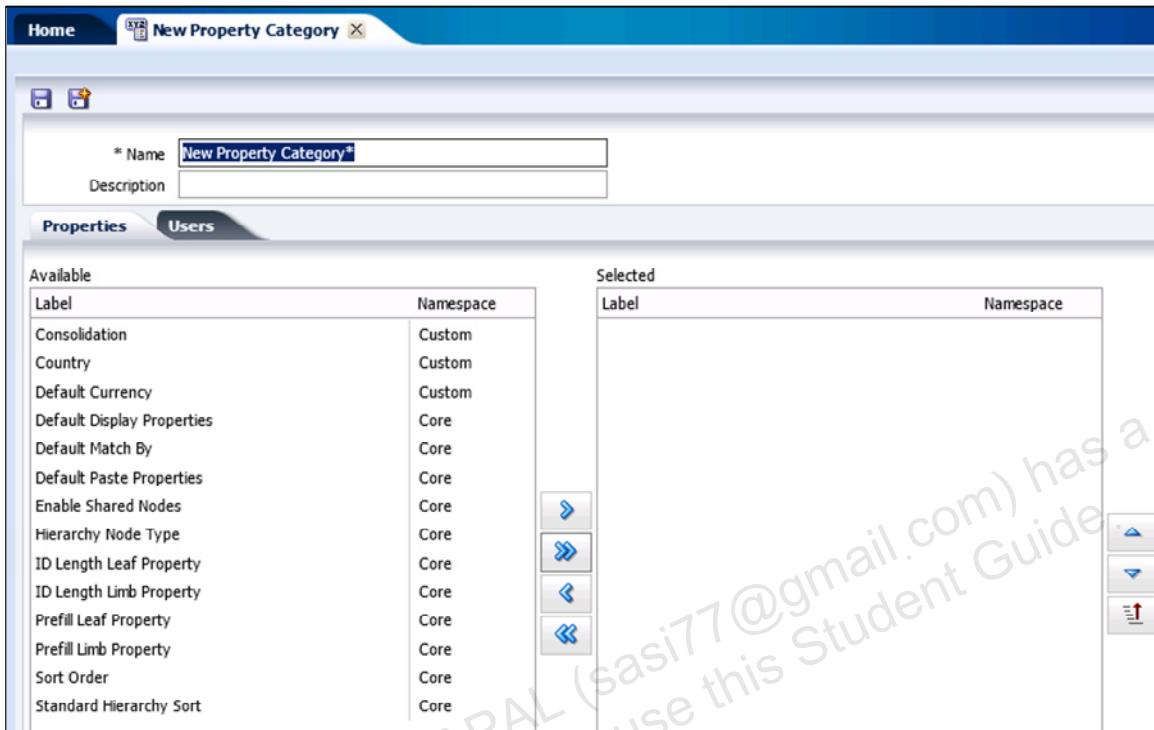
When you configure a property that is defined as both global and inheriting, you must specify a controlling hierarchy for the property. If a node exists in multiple hierarchies, the value inherited from the node ancestor (or ancestors) in the controlling hierarchy determines the value for that property in all hierarchies in which the node occurs.

In the slide example, the Acct Type property is defined as both global and inheriting. Revenue Accounts is specified as the controlling hierarchy for the Account Type property. In the Revenue Accounts hierarchy, the Services node inherits the value Income for the Acct Type property from its Total Revenue ancestor. Because Revenue Accounts is specified as the controlling hierarchy, the Services node in the Taxable Accounts hierarchy also inherits Income as the Acct Type property value.

To assign controlling hierarchies to global inheriting properties:

1. In the Browse task group, on the Hierarchies tab, right-click the hierarchy that controls the global inheriting property. Select **Assign Control** and then **Properties**.
The Set Controlled Properties dialog box is displayed.
2. Perform one or more of the following actions to assign global inheriting properties to the controlling hierarchy:
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
3. Click **OK**.

Creating Property Categories

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application administrators can create property categories to group related properties according to the data needs of their organization. For example, you group properties that belong to a particular external system (for example, Essbase, Planning, Oracle, or PeopleSoft), or a specific functional area within a company (for example, HR, Marketing). To control security privileges, an Access Manager role user can restrict user access to property categories.

A property does not need to be assigned to a category; however, the usability of the property becomes significantly reduced in some cases. Properties for versions, hierarchies, and nodes that need to be visible and editable in the property grid must belong to at least one category.

To create property categories:

1. In the Administer task group, select **New** and then **Property Category**.
The New Property Category tab is displayed.
2. Enter a name and optional description for the new property category.
3. On the Properties tab, perform one or more of the following actions to assign properties to the property category:
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
4. Select the **Users** tab, and perform one or more of the following actions to assign users to the property category:
 - Select users in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all users in the Available list to the Selected list.
 - Select users in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all users in the Selected list to the Available list.
5. On the toolbar, click the Save button ().

Loading Metadata with the Migration Utility

- The Migration Utility enables application administrators to bulk load metadata and security objects (including properties and property categories) into Data Relationship Management applications.
- You load metadata from an XML file into a Data Relationship Management application.
- A log file is created after loading and displays the following data details:
 - Audit
 - Information
 - Warning
 - Error message



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Data Relationship Management Migration Utility provides application administrators the ability to bulk load metadata and security objects (including properties and property categories) into Data Relationship Management applications. You can extract these objects from another Data Relationship Management application or access them from an integrated product in the form of an application template.

You can perform the following tasks in the Migration Utility:

- Extract metadata object types from a Data Relationship Management application to an XML file.
- Load metadata from an XML file into a Data Relationship Management application. You can load only files with the Data Relationship Management XML format. A log file is created after loading and displays the following severities of data: audit, information, warning, and error message.
- Compare metadata differences between two sources and create an XML file with the differences.
- View metadata in an XML file.
- Generate an HTML report from the load results.

To load metadata with the Migration Utility:

1. Select **Start**, then **Programs**, then **Oracle EPM System**, then **Data Relationship Management**, and then **Migration Utility**.
The main menu of the Migration Utility is displayed.
2. Click **Load**.
The Load Metadata window is displayed.
3. Click **Browse**, select the XML file that you want to load, and click **Upload**.
The Uploaded File Information window is displayed.
4. Review the information, and click **Next**.
The Login Connection window is displayed.
5. Enter Data Relationship Management connection information, and click **Log In**.
The Select Metadata Objects window is displayed.
6. Perform one or more of the following actions, and click **Next**.
 - Click the plus sign in the hierarchy tree to view objects.
 - Select the check box for an object type to select the object type and all its objects, or select the check box for the objects that you want to load.
 - Select an object name to display the object type definition in a new window.The Review Load Metadata Object Dependencies window is displayed.
7. In the Filter drop-down list, select filter options one at a time, and view the resulting list of objects that match the filter criteria:
 - **All Objects**
 - **Objects with Dependencies**
 - **Objects with Errors**
 - **Objects with Unresolved Errors**
8. Click **Next**.
The Review Load Metadata Object Dependencies window is displayed.
9. **Optional:** Select **Continue Load after Error** for the load to continue even if errors are encountered.
10. At the bottom of the window, click **Run Load**, and review the load results.
11. **Optional:** Perform one or more of the following actions to work with the log file:
 - To change the view of the log file, select the severity of detail to display: **audit**, **information**, **warning**, or **error**.
 - To save the log file, click **Download**.
 - To sort log items by any column, click the column header links.

Summary

In this lesson, you should have learned how to:

- Describe custom properties
- Configure defined properties
- Configure list values for properties
- Configure lookup values for properties
- Configure inheriting properties
- Create property categories
- Load metadata with the Migration Utility



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

Which of the following statements about Data Relationship Management properties are correct?

- a. Properties store information about nodes.
- b. Properties provide statistical information about nodes.
- c. Only application administrators can create properties and group them into property categories.
- d. A property must be assigned to a property category.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

The most commonly used data types are String, Boolean, and Integer.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

At which of the following levels can you define a property?

- a. Version
- b. Hierarchy
- c. Property category
- d. Local node



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, d

Quiz

Identify the correct statements about properties.

- a. There are three property types: Defined, Lookup, and Derived.
- b. When you configure a property that is defined as both global and inheriting, you must specify a controlling hierarchy for the property.
- c. To prevent users from entering incorrect values for properties, you can configure list values for properties.
- d. The values of a property with the Lookup property type are determined by the values of another property and a lookup table.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Importing Data

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe how to import data
- Explain import files and database sources
- Open, copy, and delete imports
- Create and run imports

About Importing Data

- Importing data is an efficient way to populate Data Relationship Management applications.
- You can include data from external applications.
- Importing always creates new versions.
 - Other versions are not affected.
 - Imported data can be verified.

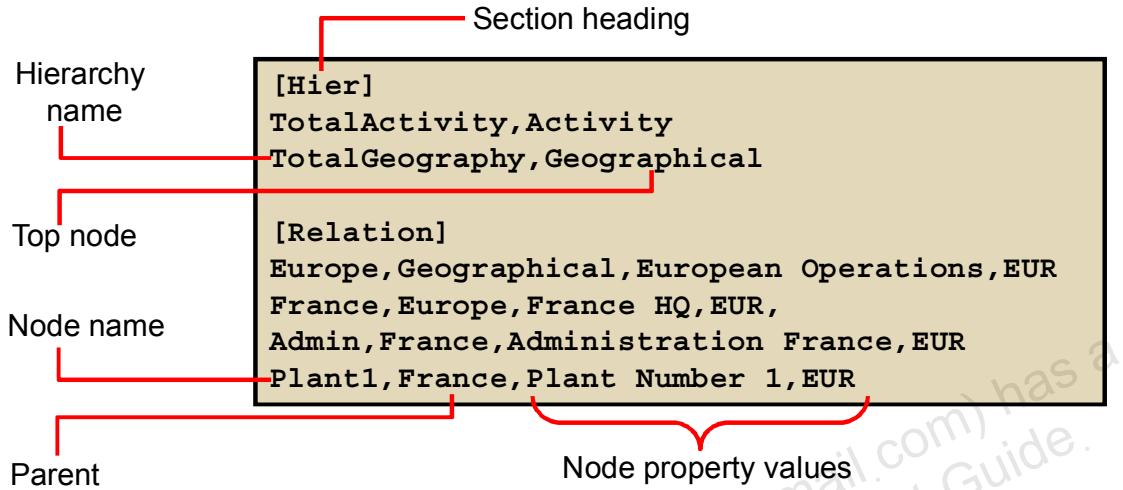
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Importing data is an efficient way to populate Data Relationship Management applications. It also provides a way to include data from external applications.

When you import data into an application, the import process creates a new version and loads the hierarchies, nodes, and property values into the new version. This approach enables you to verify that the data was imported correctly, and it ensures that other versions are not affected unexpectedly.

Importing from Files



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An *import file* consists of the data that you want to import into a Data Relationship Management application. A variety of formats is supported for import files, including delimited and fixed width.

Import files are organized into sections and columns:

- Sections designate the type of data being imported. An import file can contain several sections for importing hierarchy, node, relationship, and other types of information. The sections can appear in any order and must be identified by section headers. You can include multiple sections of the same type in a single file (for example, several hierarchies for the same Essbase dimension). The import process combines the sections before importing them.
- The data in each section is organized into columns. An import column is mapped to a corresponding property in Data Relationship Management, such as Name, Description, or any other non-read-only property. Columns can be fixed width or they can be separated by a delimiter, such as a comma or a colon. Columns can be in any order and do not require labels.

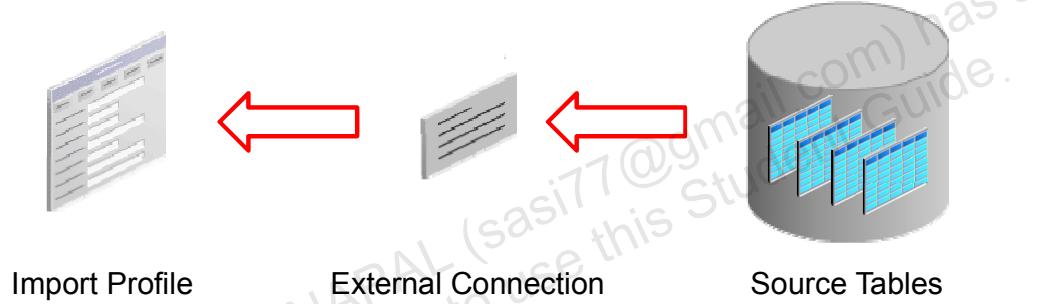
The following table describes the valid sections and required columns for each section:

Import Section	Required Columns	Description
Version	Version Name	Contains version property values
Hierarchy	Hierarchy Name, Top Node	Contains structural information about each hierarchy and top node to be imported. Can also contain columns for hierarchy property values.
Node	Node Name	Contains structural information about each node to be imported. Can also contain columns for node property values. Nodes that are not related to hierarchies will be imported into the new version as orphans.
Relation	Parent Node Name, Child Node Name	Contains parent-child information about each relationship to be imported. Can also contain columns for node property values for the child in each pair.
Hierarchy Node	Hierarchy Name, Node Name	Contains node property values that are specific to a given hierarchy

Importing from Database Tables

You can:

- Leverage common external connections to supported relational databases
- Load versions, hierarchies, nodes, and property values from source tables
- Map import sections to source data from tables
- Use a query to filter source data

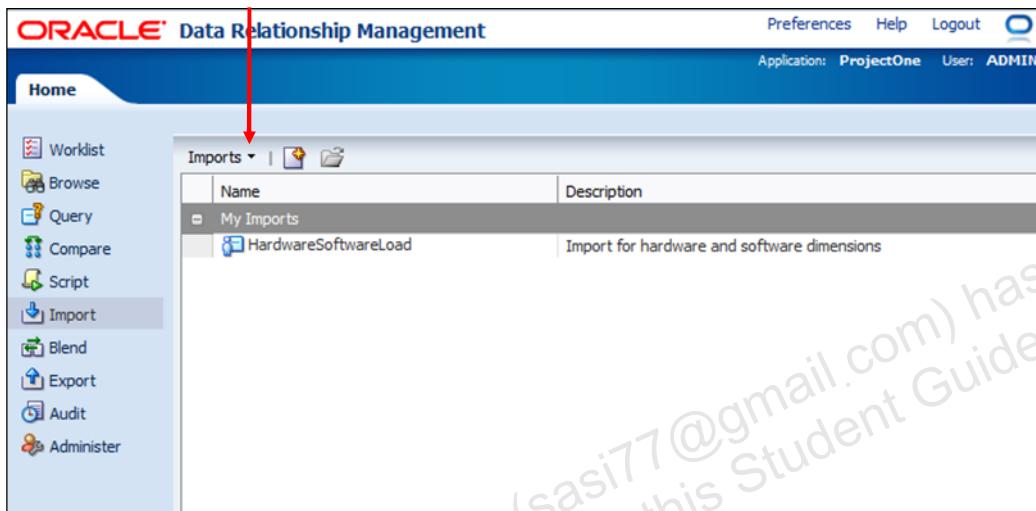


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can import data from database tables into Data Relationship Management by using external connections to supported relational databases (for example, Oracle and Microsoft SQL Server relational databases). An import profile can source data from tables in a single database. Each section of an import profile can be mapped to a single database table. You can also configure a query filter to import a subset of data from the source database tables.

Opening, Copying, and Deleting Imports

Open, Copy, and Delete options



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

You can open, copy, and delete imports in the Import task group. *Imports* define the source, style, columns, filters, and target for import operations. Be careful when deleting imports because deletions cannot be undone.

To open imports:

1. On the Home page, select the **Import** task group.
2. In the list of imports, select an import.
3. In the Imports drop-down list, select **Open**.

The import is displayed on another tab with five subtabs (Source, Style, Columns, Filters, and Target). The Source subtab is displayed by default.

To copy imports:

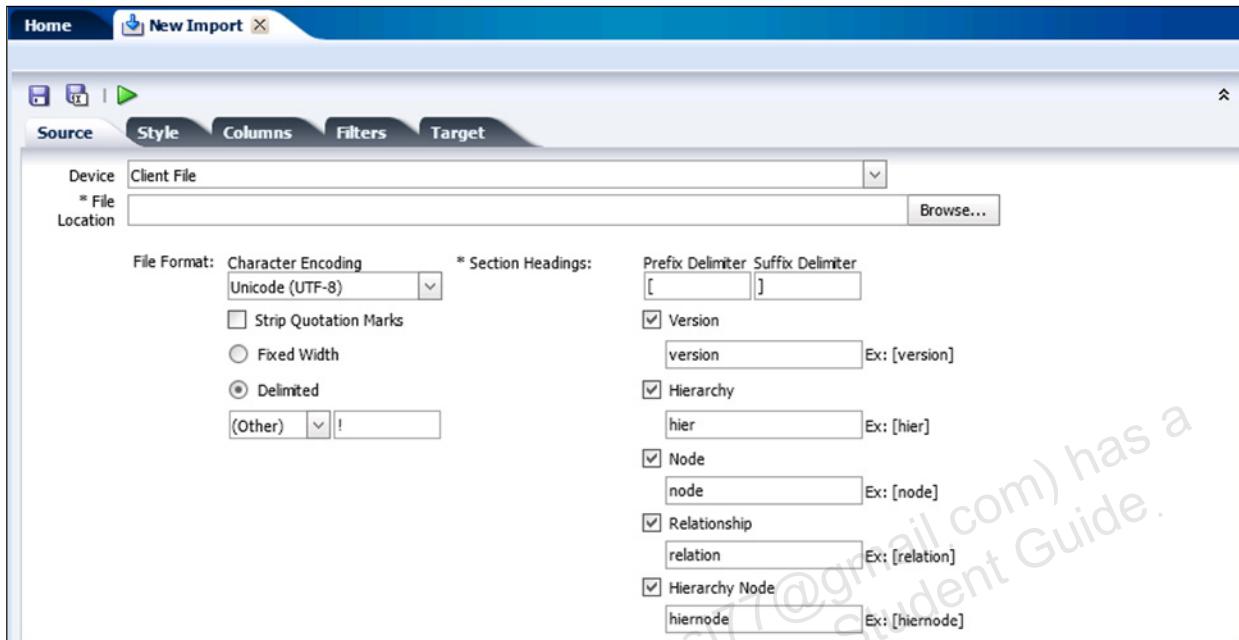
1. In the list of imports in the Import task group, select an import.
2. In the Imports drop-down list, select **Copy**.
The Copy Import dialog box is displayed.
3. Enter a name and optional description for the import.
4. In the Object Access Level drop-down list, select **User**, **Standard**, or **System**.
5. Click **OK**.

To delete imports:

1. In the list of imports in the Import task group, select an import.
2. In the Imports drop-down list, select **Delete**.
The Confirm Delete dialog box is displayed.
3. Select **Delete the Import**.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Creating Imports



Import Wizard

ORACLE

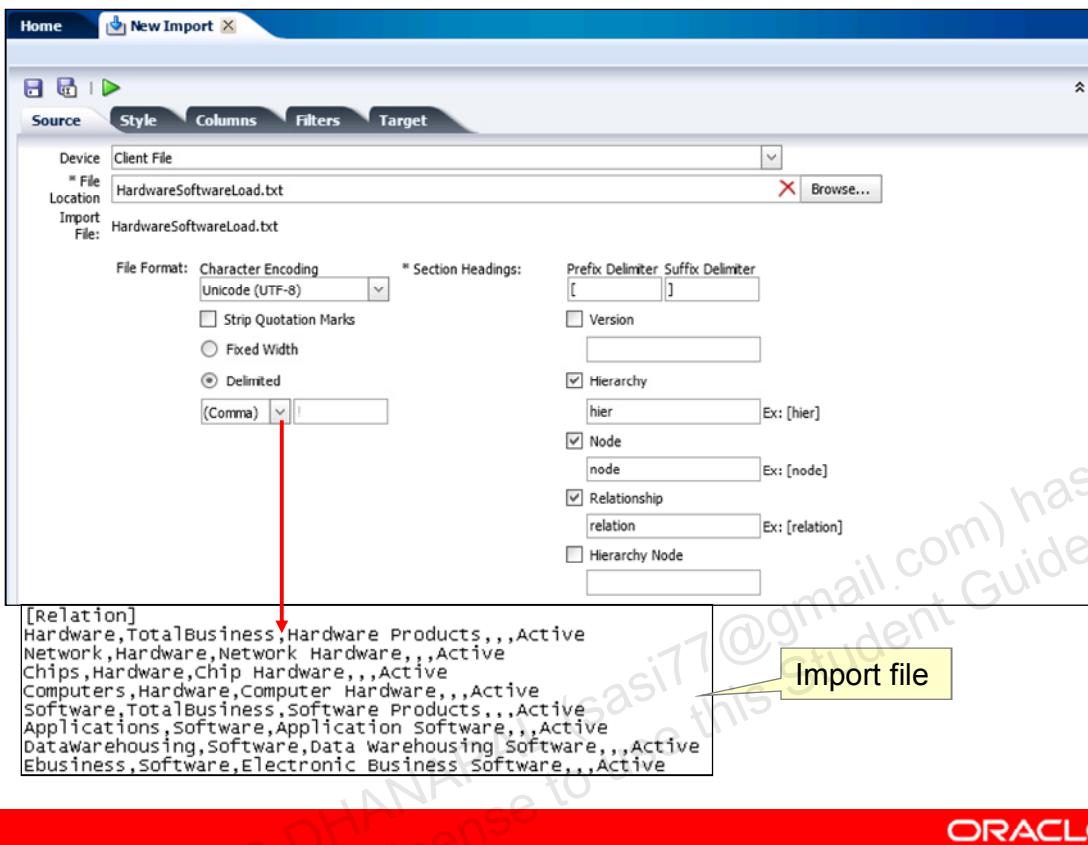
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You create imports in the import wizard, which consists of the Source, Style, Columns, Filters, and Target tabs. You start the import wizard in the Import task group.

To create imports:

1. On the toolbar in the Import task group, click the New Import button ().
The import wizard is started, and the New Import tab is displayed.
2. Configure the Source, Style, Columns, Filters, and Target tabs.
The configuration procedures are presented in the next topics in this lesson.
3. Save the import.

Configuring the Source: Import Files



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Source tab in the import wizard, you select an import source and file formats, and you configure section headings.

To configure the import file source:

1. In the import wizard, click the **Source** tab.
2. In the Device drop-down list, select one of the following file sources:
 - Select **Client File**, click **Browse**, and select the import file.
 - Select **Server File**, and then select the connection and a file name.
3. Select the following file format options:
 - In the Character Encoding drop-down list, select a codepage (for example, UTF-8).
 - **Optional:** Select **Strip Quoted Strings** to remove quotation marks during the import.
Note: Single quotation marks ('') and double quotation marks (") are removed. If both types of quotation marks are used at the same time, only the outer set of quotation marks is stripped.
 - Select **Fixed Width** or **Delimited**. For Delimited, enter the delimiter character. For Fixed Width, enter fixed-width column values under Columns Options on the Columns tab.

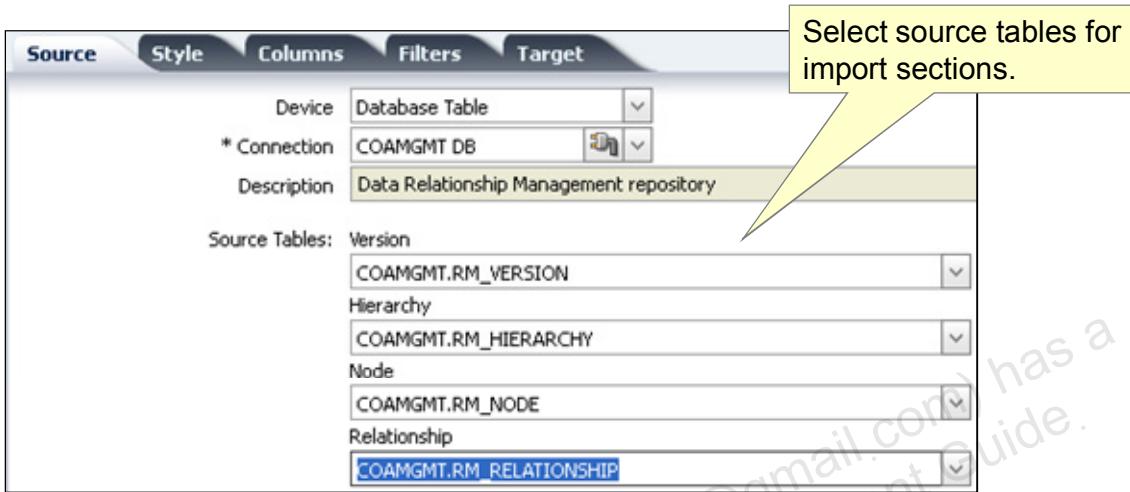
4. Configure the following section heading options:

- In the Prefix Delimiter and Suffix Delimiter boxes, enter characters to be displayed at the beginning and end of a section heading (for example, /, /**, or []). The suffix delimiter 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. Clear sections that are not in the import source.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Configuring the Source: Database Tables



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To configure the database table source:

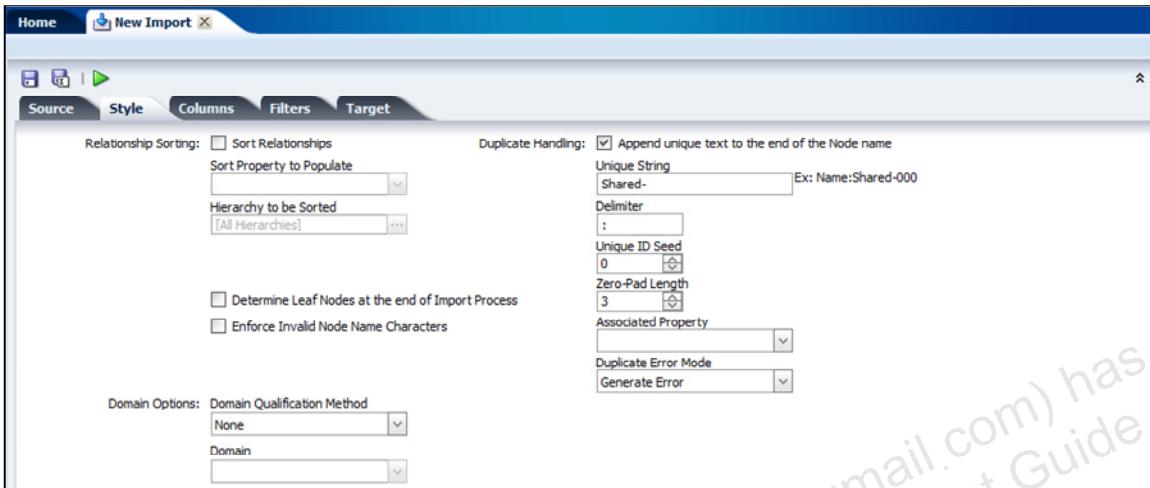
1. In the import wizard, click the **Source** tab.
2. In the Device drop-down list, select **Database Table**.
3. From Connection, select the external connection to the database.
4. Click the Test Connection button () to test the connection.
5. For each import section, select the database table from which to import.

Note: The list of available database tables to choose from is controlled by the list of tables configured for the external connection.

6. **Optional:** Select Clear Imported Records After Processing to clear database records that were loaded from the source connection when the import successfully completes.

Note: To use this option the database user configured for the external connection must have the DELETE privilege for the database tables.

Configuring the Style



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Style tab in the import wizard, you can perform the following tasks:

- Enable the import to sort nodes based on the order in which they are displayed in the Relation section of the source file.
- Configure childless nodes as leaf nodes.
- Enforce invalid node name characters.
- Append unique names to duplicate nodes (for example, <node name>:Shared-000) to enable them to be imported rather than rejected. A *duplicate node* is any node that appears under multiple parents within the same hierarchy.
- Qualify node names with a prefix or suffix in order to assign the nodes to a domain after import.

Leaf Nodes Option

The “Determine Leaf Nodes at the end of Import Process” option requires a second pass of the imported data and may require additional time to process. If you do not select this option, all nodes are imported as limb nodes unless the Leaf property is specified as a column in the Node or Relation section of the imported data.

Enforce Invalid Characters Option

The Enforce Invalid Node Name Characters option stops nodes from being imported if the name of the node defined in the source file includes invalid characters configured in the InvName system preference for the application. Otherwise, nodes with invalid characters are allowed to be imported for further analysis.

Domain Options

The Domain Qualification method enables the import process to apply a prefix or suffix to node names at the time of creation to allow the nodes to be assigned as members of a domain after import. The actual prefix/suffix to be applied is configured by domain. A single domain to be used for all imported nodes can be specified for the import profile or the domain for each node can be specified as a column in the import source to allow nodes to be qualified for different domains in the same import process.

Duplicate Handling Options

You must configure duplicate handling options if you want duplicate nodes that are identified during the import process to be replaced with unique node names. As you specify options, an example displays the resulting value. When the import process is completed, you can query duplicate nodes by using the specified naming guidelines to resolve underlying data issues.

If you want to import shared nodes, you must use the duplicate handling options on this page. Make sure that the prefix or suffix matches the one that is defined in the system preferences for identifying shared nodes.

The unique text that you can append to the end of node names for duplicate nodes consists of the parameters described in the following table. If you do not append text to the end of node names, the unique portion of the name is displayed as a prefix at the beginning (for example, 001NodeA, 002NodeA).

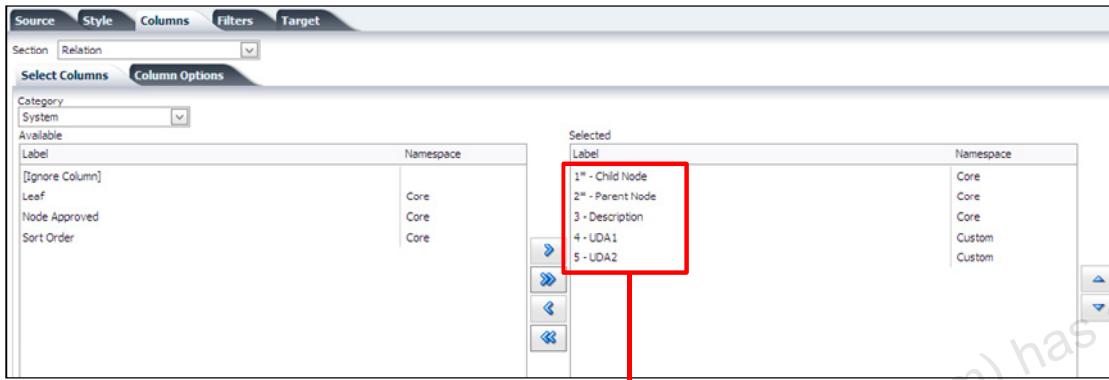
Parameter	Description
Unique String	A string to be used to form part of the new unique node name Example: Shared-
Delimiter	A character to be used to separate the original node name from the unique portion of the name Example: a colon (:)
Unique ID Seed	A starting number for IDs Example: If NodeA occurs 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 to illustrate the use of unique ID seed.
Zero Pad Length	The minimum number of digits required in the unique ID portion of the duplicate name 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.

Parameter	Description
Associate Property	The property that enables duplicate nodes to be grouped so that they can be easily queried after the import is completed. 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	The mode to determine the severity with which the import processor treats a duplicate. You can choose to add a warning or an error message to the log file for each duplicate. Processing stops if the number of errors exceeds the Max Errors value, which you specify on the Target tab.

To configure the style:

1. In the import wizard, click the **Style** tab.
2. **Optional:** Select **Sort Relationships**, and then select the node property in which to store the sort order, and the hierarchies to which to apply the sort.
3. **Optional:** Select **Determine Leaf Nodes at the end of Import Process**.
4. **Optional:** Select **Enforce Invalid Node Name Characters**.
5. **Optional:** Next to Duplicate Handling, select **Append unique text to the end of the Node name**, and configure the unique string, delimiter, unique ID seed, zero-pad length, associated property, and duplicate error mode (**Generate Error** or **Generate Warning**).
6. **Optional:** In the Domain Options section, specify how domain names are qualified in the import by selecting an option:
 - **None:** Domain names are not used
 - **Select a Domain:** Qualifies all nodes in the domain
Note: If you select this option, select a domain name from the drop-down list.
 - **Select Import Column:** Qualifies nodes based on the selected columns
Note: Qualifying the node names will append the qualifier text of the specified domain; it will not assign the node to a domain.

Configuring the Columns: Import Files



5100,Manufacturing,Manufacturing-Plant1,USD,Manufacturing
 (A row from the import file in the Relation section)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Columns tab in the import wizard, you map properties to columns in the input file and you map columns to hierarchies. Create separate mappings for each section that is specified in the file. Consider the following guidelines:

- Each property maps to a column in the import file.
- The number of properties in the Selected list must match the number of columns in the import file for the selected import section.
- The order of the properties in the Selected list must correspond to the order of the columns in the file. For example, if data for the Description property is in the third column of the import file, the Description property must be displayed as the third item in the Selected Properties list.
- A special property called Ignore Column is available as a placeholder for any column in the input source that should not be processed during the import.
- For hierarchy-specific (or local) node properties, you can use the Column Options tab to specify the hierarchies for which the mapping is valid (for example, if you want to populate the Currency property only for nodes in the Geographical hierarchy but not from nodes in the Functional hierarchy). By default, all hierarchies are selected. This feature applies only to local node properties.

In the slide example, the Relation section is configured with five properties (Child Node, Parent Node, Description, UDA1, and UDA2) to match the number of values in the rows under [Relation] in the import file.

To configure the columns:

1. In the import wizard, click the **Columns** tab.
2. In the Section drop-down list, select **Version, Hierarchy, Node, Relation, or Hierarchy Node**. The available file sections correspond to the sections that you selected on the Source tab.
3. Perform one or more of the following actions to configure columns for the import:
 - Select columns in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all columns in the Available list to the Selected list.
 - Select columns in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all columns in the Selected list to the Available list.

Note: Use the Category drop-down list to select a property category to ease navigation.

4. **Optional:** Select **[Ignore Column]** in the Available list, and click the Select button () to move it to the Selected list to act as a placeholder for any column in the input source that should not be processed during the import.
5. **Optional:** In the Selected list, select a column, and click the Move Up button () or the Move Down button () to position it. The first column in the list is the first column in the source, the second column in the list is the second column in the source, and so on.
6. Click the **Column Options** tab, and define options for each import column:
 - a. Double-click a column name row.
 - b. Perform one or more of the following actions:
 - 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. 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. You can define the default value for each import column width by using the Column Width parameter of a property definition.
 - If you selected Select Import Column for Domain options on the Style tab, the node and relationship sections on the Columns tab include an additional subtab named Domain Options. On this subtab, you can select the import columns that identify the domain for the node, parent, and child used for name qualification purposes.

Configuring the Columns: Database Table

The screenshot shows the Oracle Data Relationship Management interface with a 'New Import' project selected. The 'Columns' tab is active. A yellow callout box points to the 'Action' column of the 'Database Column Options' table, containing icons for edit, update, and help. The table data is as follows:

Column	Source Field	Field Type	Length	Order	Action
Name	C_ABBREV	String	255		
Description	C_DESCR	String	255		
Leaf	B_LEAF	Integer		-1	

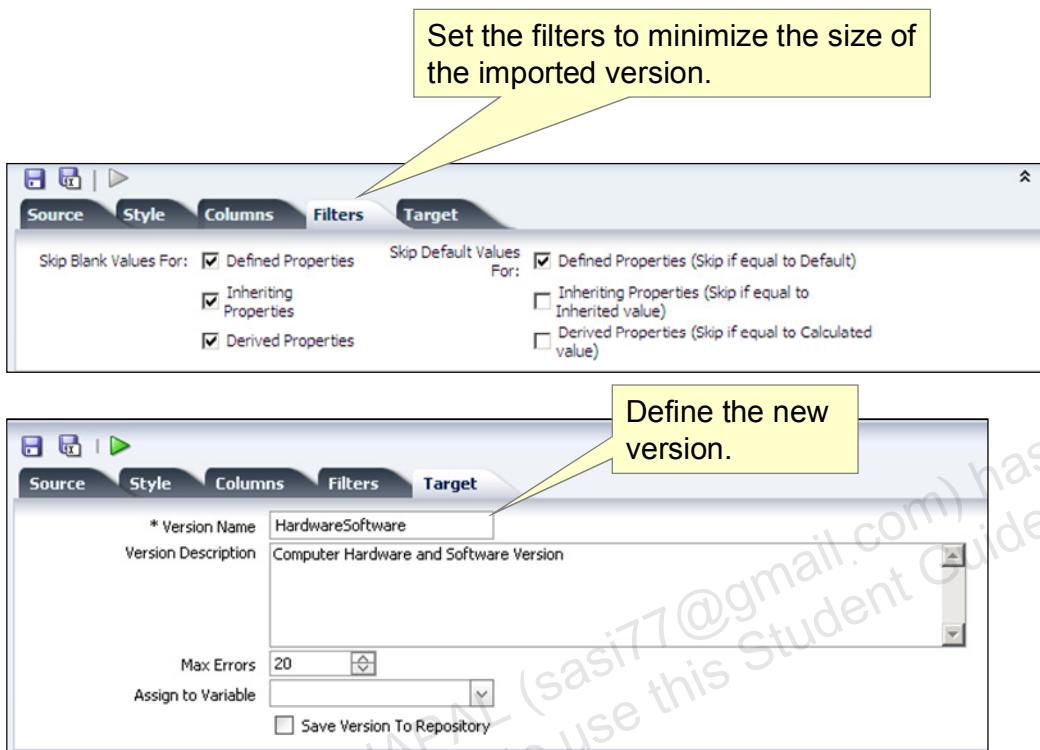
At the bottom right of the interface is the ORACLE logo.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you are importing from database tables, the Database Options tab is displayed. For each import section, the following columns are displayed:

- **Column:** Displays the name of the Data Relationship Management property being populated
- **Source Field:** Click the Edit button () in the Action column to select a field from the source database table for the section. Click the Update button () to save the changes.
Note: A source field may be mapped to more than one Data Relationship Management property.
- **Field Type:** Displays the data type of the field from the source database table
- **Length:** Displays the length of the field from the source database table (if applicable)
- **Order:** Click the Edit button () in the Action column and select a field from the source database table used to order the data before import. Click the Update button () to save the changes.

Configuring the Filters and Target



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

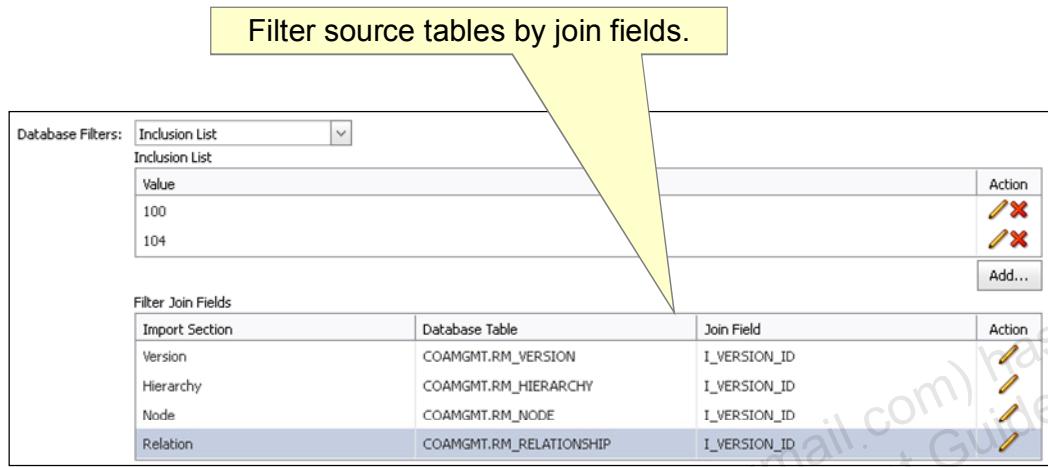
To configure the filters:

1. In the import wizard, click the **Filters** tab.
2. **Optional:** Select **Defined Properties**, **Inheriting Properties**, and **Derived Properties** to skip their blank values.
3. **Optional:** Select the following items to skip their default values:
 - **Defined Properties** (skip if equal to Default)
 - **Inheriting Properties** (skip if equal to Inherited value)
 - **Derived Properties** (skip if equal to Calculated value)

To configure the target:

1. In the import wizard, click the **Target** tab.
2. Enter a name and an optional description for the new version.
3. In the Max Errors drop-down list, select the maximum number of errors that can occur during the processing before the Import stops. The default number of errors is 20.
4. In the Assign To Variable drop-down list, select the version variable to which the version created by the import process will be assigned.
5. Select Save Version To Repository to automatically save the new version to the repository.

Filtering Source Tables



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you are importing data from database tables, the following database filters are available:

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

Saving Imports

Specify the following information:

- Name
- Description
- Object access level
 - User
 - Standard
 - System
 - Custom object access group



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When saving imports, you provide a name and description for your import, and select one of the following object access levels:

- **User** enables only you to run and edit the import.
- **Standard** enables all users to run the import, but only users with the Data Manager role can edit it.
- **System** enables only users with the Application Administrator role to run and edit the import.
- Custom object access group.

Note: Object access levels are the same for queries, compares, blenders, and exports.

To save imports:

1. On the toolbar in the import wizard, click the Save As button ().
The Save Import As dialog box is displayed.
2. Enter a name and an optional description for the import.
3. In the Object Access Level drop-down list, select **User**, **Standard**, or **System** (depending on your user role).
4. Click **OK**.

Running Imports

- An import must be open before you can run it.
- After an import runs, view summary information and detailed messages in the import results.
- Data is not automatically saved after an import unless Save Version To Repository is selected on the Target tab.

You can download import results as CSV, PDF, RTF, or XLS.

Type	Action	Hierarchy	Node	Property	Before	After	Message
Error	Duplicate Node Found	Geography	4100		4100	4100:Shared-000	Replaced Duplicate Node '4100' with new Node '4100:Shared-000'

Import results

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An import must be open before you can run it. After running an import, you can view summary information and detailed messages in the import results. The import results identify the completion status of the process, the statistics for errors, and the statistics for hierarchies, nodes, and orphans that were created. Messages for warnings or errors that were encountered during the import are displayed below the summary information.

You can download import results to an external CSV, PDF, RTF, or XLS file.

Unless you selected the Save Version To Repository option on the Target tab, data is not permanently saved after an import; you must save the new version to make it permanent.

To run imports:

1. If necessary, open an import in the Import task group.
2. If necessary, on the Source tab in the import wizard, click **Browse**, select the import file, and click **Open**.
3. On the toolbar in the import wizard, click the Run button ().
4. View summary information and messages in the import results.
5. **Optional:** Download the results.
 - a. Above the import results, in the Download drop-down list, select **Download as CSV**, **Download as PDF**, **Download as RTF**, or **Download as XLS**.
The File Download dialog box is displayed.
 - b. Click **Save**.
The Save As dialog box is displayed.
 - c. Browse to a location, enter a file name, and click **Save**.
6. In the Browse task group, save the new version.

Summary

In this lesson, you should have learned how to:

- Describe how to import data
- Explain import files and database sources
- Open, copy, and delete imports
- Create and run imports

Quiz

When you import data into an application, you always create a new version.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Which of the following statements about import files are correct?

- a. Import files contain data that is imported into Data Relationship Management applications.
- b. Import files are organized into sections and columns.
- c. Sections in import files can appear in any order and must be identified by section headers.
- d. Columns in import files are mapped to read-only properties in Data Relationship Management applications.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

Which of the following guidelines for mapping properties to columns in import files are correct?

- a. Each property maps to a column in the import source.
- b. The number of properties in the Selected list must match the number of columns in the import source.
- c. The order of the properties in the Selected list must correspond to the order of the columns in the file.
- d. A special property called Ignore Column is available as a placeholder for any column in the input source that should not be processed during the import.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Quiz

If you want to import shared nodes, you must use the duplicate handling options and make sure that the prefix or suffix matches the one that is defined in the system preferences for identifying shared nodes.

- a. True
- b. False

 ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Querying and Comparing Data



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Search for nodes
- Create queries
- Create visual, structure, rename, and property compares
- Work with query and compare results

Searching for Nodes

Searching for nodes in a version with names that start with 100:

The screenshot shows the 'Find Nodes' tab selected in the top navigation bar. A search input field contains '100*' with a red box around it. A yellow callout box labeled 'Search by name.' points to this field. Below the search bar is a table with columns 'Select', 'Name', and 'Description'. The 'Name' column shows a hierarchy tree under the 'Product' category, with nodes like '100', '100-10', '100-20', and '100-30'. The 'Description' column lists 'Speakers', 'Stereo Speaker System', 'Surround Sound Speakers', and 'Digital Speakers'. There are green circular icons with '+' signs next to some rows.

Searching for the description “Atlanta Office” in the Entity hierarchy:

The screenshot shows the 'Entity - Accessories' tab selected. A search input field contains 'Atlanta Office' with a red box around it. A yellow callout box labeled 'Search by description and locate results in the tree.' points to this field. Below the search bar is a table with columns 'Name' and 'Description'. The 'Name' column shows a hierarchy tree under the 'TotalEntities' category, with nodes like 'Asia', 'USA', 'East', 'Atlanta', 'New York', and 'West'. The 'Description' column lists 'Asia', 'United States', 'East US Region', 'Atlanta Office', 'New York Office', and 'West US Region'. The 'Atlanta Office' node is highlighted in blue. The Oracle logo is visible at the bottom right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

In versions and hierarchies, by default you can search for nodes by name or by description. You can also specify additional properties to enable users to search with, by configuring the `FindByProperties` system preference.

Searching enables you to quickly locate nodes for viewing and updating purposes. In the search text, you can use the asterisk (*) and percent sign (%) as wildcards for multiple characters and the underscore (_) as a wildcard for single characters in a specific position. You can also use the backslash (\) as an escape character for exact match searches.

A version-level search finds nodes in any hierarchy in the version that matches the search criteria. For hierarchy searches, you can choose to view resulting nodes in a list or you can have the nodes highlighted in the hierarchy. Only descendants of the selected node in the hierarchy are searched. You can select a node other than the top node of the hierarchy to search only a branch.

The slide shows two examples. In the first example, the user searches for nodes in a version with names that start with 100. The user enters 100* to make sure all nodes are returned in the results. Search results for a version are grouped by hierarchy. In the second example, the user searches for nodes with the description “Atlanta Office” in the Entity hierarchy. The resulting Atlanta node with the matching description is highlighted in the hierarchy tree.

To search for nodes in versions:

1. In the Browse task group, select a version.
2. Select the **Find Nodes** tab.
3. On the toolbar, click the Find By button (), select **Find By**, and then select **Name** or **Description**.
4. In the text box, enter search text, and click **Go**.
Nodes that match the search criteria are listed.
5. **Optional:** In the Go column for a listed node, click Go To Node button () to view the node in its hierarchy.

To search for nodes in hierarchies:

1. On the Hierarchies tab in the Browse task group, double-click a hierarchy to open it.
The hierarchy tab is displayed.
2. On the toolbar, click the Find By button (), select **Find By** and then **Name** or **Description**.
3. On the toolbar, click the Find By button (), select **Show Results** and then **Locate in Tree** or **Show as List**.
4. In the text box, enter search text, and click **Go**.
Nodes that match the search criteria are listed on the Relationships tab or they are highlighted in the hierarchy tree, depending on how what you did in step 3.
5. **Optional:** Perform one of the following actions:
 - If search results are returned in a list, click the Go To Node button () in the Go column on the Relationships tab to view the node in its hierarchy.
 - If search results are returned by highlighting the node in the tree, click the Find Next button () to view the next copy of the node in the hierarchy.

If domains are enabled for the version, then you can further narrow the search for nodes in versions and hierarchies by specifying a particular domain.

To select a domain for searches:

- On the toolbar, click the Find By button (), select **Find By**, select **Find In Domain**, and then select a domain.

Creating Queries

- With the query wizard in the Query task group, you can query for nodes in a version or a hierarchy based on criteria that are more complex than those provided by the search feature.



- You can create two types of queries: global and local.
- You can save and reuse queries for recurring analyses.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With the query wizard in the Query task group, you can create queries for nodes in a version or a hierarchy based on one or more properties.

You can create two types of queries:

- 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. You can run them 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.

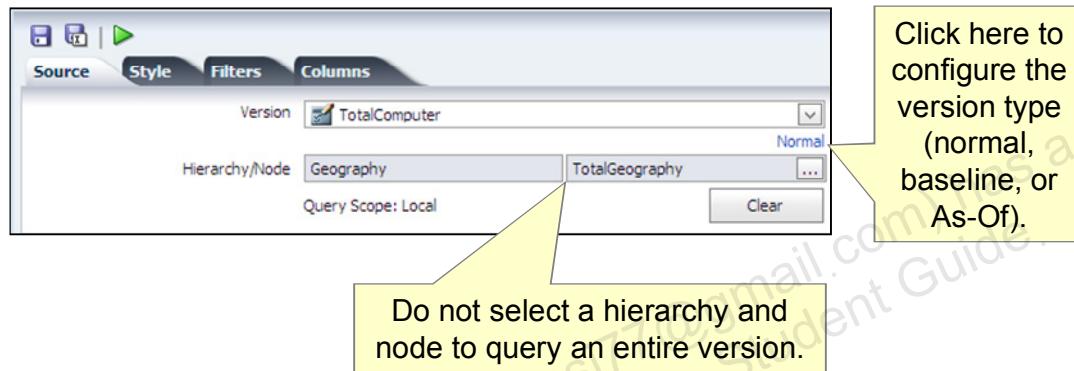
To start new queries:

- On the Home tab, select the **Query** task group.
- On the toolbar, click the New Query button ().

The query wizard is started, and the Source subtab is displayed by default. The query wizard is organized into four tabs (Source, Style, Filters, and Columns) to guide you when defining the parameters and criteria for a query. You can navigate between the tabs at any point during the process.

Configuring the Source

- Define the query scope.
- You can query normal, baseline, and As-Of versions.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Source tab in the query wizard, you define the query scope. If you select only a version (and not a hierarchy and top node), the scope is global. If you select a version, hierarchy, and top node, the scope is local. All nodes below and including the top node are queried.

By default, the version list contains only normal versions. However, you can select a baseline version if needed. You can also create an As-Of version from here.

To configure the source:

1. In the query wizard, click the **Source** tab.
2. In the Version drop-down list, select a version to query or a version variable.
Note: A version variable can be saved with a query, whereas the selection of a regular version is not saved.
3. **Optional:** If you want to select a baseline version or create an As-Of version, select **Normal** and configure the Version Parameters dialog box as appropriate.
4. Perform one of the following actions to create the query:
 - To create a local query: In the Hierarchy/Node box, click the ellipsis button ([...]). In the Select Node dialog box, select a hierarchy and a top node, and click **OK**.
 - To create a global query: Leave the Hierarchy/Node box empty.

Configuring the Style

- Configure how query results are displayed.
- You can list nodes, mark nodes, or do both.

The screenshot shows two views of the query wizard's style configuration:

- Listed Nodes:** Shows a table with columns "Select", "Name", and "Go". The "Name" column lists countries: South Africa, China, France, Italy, United Kingdom, and USA. Each item has a checkbox in the "Select" column and a green circular icon with a plus sign in the "Go" column.
- Marked and Expanded Nodes:** Shows a hierarchical tree view. Root categories include TotalGeography, Africa, Asia, Europe, and North America. Under Africa, South Africa is selected (indicated by a checked checkbox). Under Europe, France is selected. Under France, 1400 and 5200 are listed. Under Italy, United Kingdom, and USA, all items are selected. To the right, a "Description" column provides details for each node.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Style tab in the query wizard, you determine how query results are displayed. You have the following options:

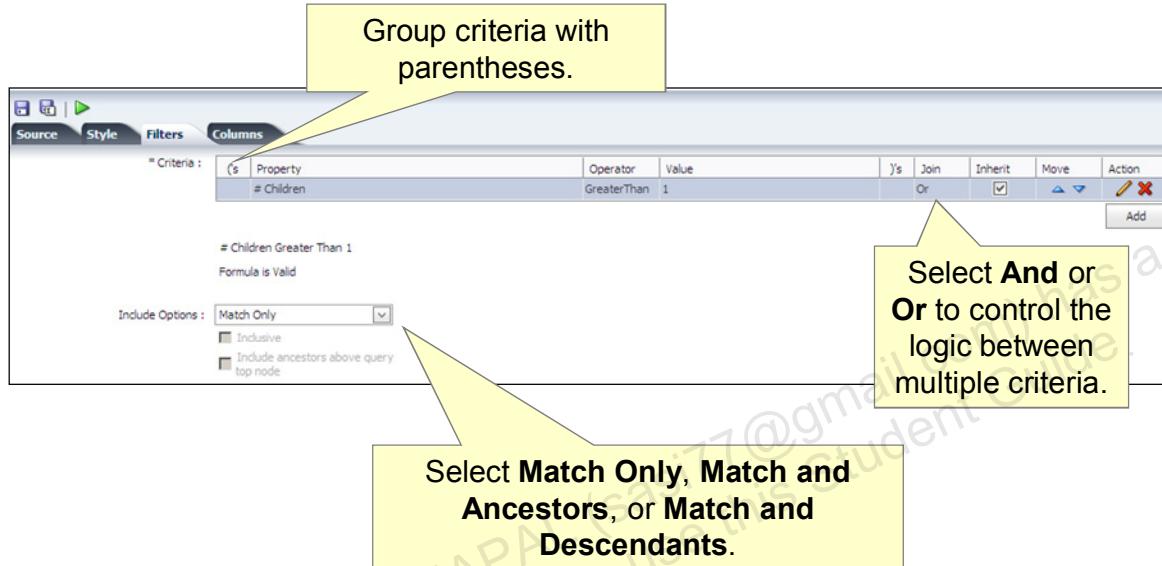
- List** nodes that match the query criteria. Global queries return results only in list format. By default, list is selected.
- Mark Nodes** that match the criteria with a check mark in the hierarchy.
- Both** list and mark nodes that match the criteria.

To configure the style:

- In the query wizard, click the **Style** tab.
- In the Result Display drop-down list, select **List**, **Mark Nodes**, or **Both**.
- Optional:** If you selected Mark Nodes or Both for a local query, select **Expand to Marked Nodes** so that the hierarchy with marked nodes is automatically opened and expanded.

Configuring the Filters

Define the query criteria and specify which nodes are included in query results.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

On the Filters tab in the query wizard, you define the query criteria by considering the following:

- A number of operators are supported, including Like, In, and Length Equal.
- You can create query criteria similar to WHERE clauses in conventional SQL statements, and run queries against sets of nodes.
- You can create complex property queries by grouping multiple criteria with AND and OR operators and parentheses.
- You can include a property's inherited values or you can match only on defined values.

You also specify which nodes are included in the query results. The following options are available:

- **Match Only** returns only the nodes that match the criteria.
- **Match and Ancestors** returns all nodes that match the criteria, and their ancestors. If needed, you can also include any ancestor nodes above the query top node.
- **Match and Descendants** returns all nodes that match the criteria and their descendants.

To configure the filters:

1. In the query wizard, click the **Filters** tab.
2. Click **Add** to insert a query row.
3. Configure the criteria boxes as needed: Enter parentheses, select a property and query operator, enter a value, select a join operator (**And** or **Or**), select **Inherit**, and click the Update button () to save the row.

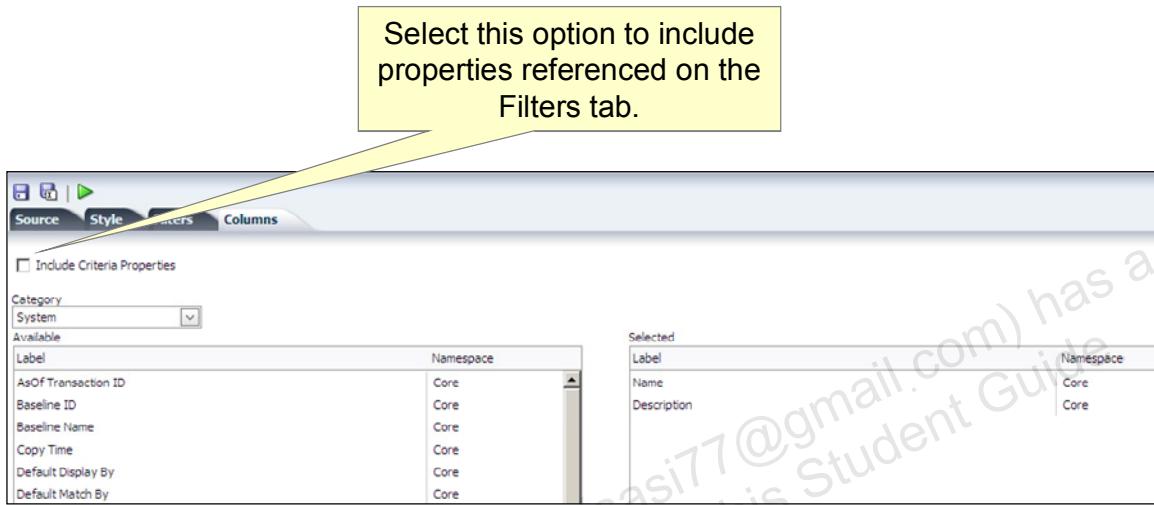
The formula is stated below the criteria boxes. “Formula is Valid” is also stated if the formula passes the syntax check.

Note: Refer to the *Oracle Data Relationship Management User's Guide Release 11.1.2.3* for the operator descriptions.

4. **Optional:** Repeat steps 2 and 3 to enter another filter criteria.
5. **Optional:** Perform one or more of the following actions:
 - Click the Edit button () to modify a row.
 - Click the Delete button () to delete a row.
 - Click the Move Down button () and the Move Up button () to move a row.
6. In the Include Options drop-down list, select **Match Only**, **Match and Ancestors**, or **Match and Descendants**.
If you select “Match and Ancestors” or “Match and Descendants,” the Inclusive option is selected by default and includes the node matching the criteria in the result.
7. **Optional:** For global queries, if you select Match and Ancestors, select **Include ancestors above query top node** to include ancestor nodes that are above the top node selected for the query.

Configuring the Columns

Select properties to display as columns in the query results.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Columns tab in the query wizard, you select properties to display as columns in the query results. You can choose to automatically include in the query results properties that are referenced in the filter criteria.

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

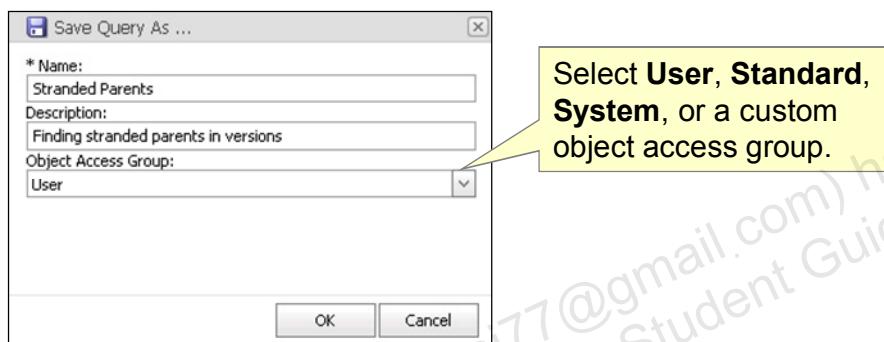
To configure the columns:

1. In the query wizard, click the **Columns** tab.
2. **Optional:** Select **Include Criteria Properties**. This option enables including properties referenced as query criteria as columns in query results returned in a list format.
3. In the Category drop-down list, select a property category.

4. Perform one or more of the following actions to configure the properties to be displayed as columns:
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
5. **Optional:** In the Selected list, select a property, and click the Move Up button () or the Move Down button () to reorder it.
The first property in the list is the first column, the second property is the second column, and so on.
6. **Optional:** Click the Alphabetize button () to order the selected columns alphabetically.

Running and Saving Queries

- You run and save queries from the query wizard.
- You can run queries without saving them and save queries without running them.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After you configure your query, you are ready to run and save the query in the query wizard. You can run queries without saving them and save queries without running them. When saving queries, you set the object access level the same way you set it with imports.

Note: Saving compares is similar to saving queries.

To run queries:

1. On the toolbar in the query wizard, click the Run button ().
The query results are displayed in a list or as marked nodes in the tree.
2. **Optional:** Click the Return to Query Wizard button () to return to the Source tab in the query wizard.

To save queries:

1. On the toolbar in the query wizard, click the Save As button ().
The Save Query As dialog box is displayed.
2. Enter a name for your query and an optional description.
3. In the Object Access Group drop-down list, select **User, Standard, System**, or a custom object access group.
4. Click **OK**.

Creating Compares

- Compares enable you to identify similarities or differences of nodes and property values between hierarchies.
- You can compare two hierarchies within the same version, the same hierarchy in two different versions, or different hierarchies in different versions.
- You compare hierarchies in the Compare task group.
- You can perform the following comparisons:
 - Visual compare
 - Structure compare
 - Rename compare
 - Property compare

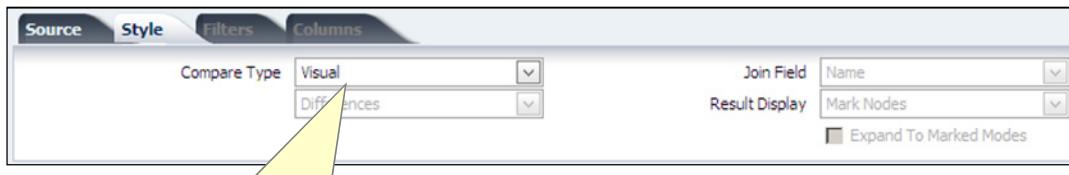


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the Compare task group, you can perform the following comparisons:

- **Visual compare** displays the hierarchy trees side-by-side without running a comparison process.
- **Structure compare** identifies nodes that are similar or different between two versions of the same hierarchy or two points of different hierarchies. This compare returns nodes from each hierarchy that are similar or different based on the type of comparison being performed. You can display the results as two lists, two marked trees, or both.
- **Rename compare** is the same as the structure compare, but it treats renamed nodes as being the same for comparison purposes.
- **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.

Performing Visual Compares



On the Style tab, select **Visual** for the Compare Type. All other options become unavailable.

Example



HierarchyA Nodes in Version 1 HierarchyA Nodes in Version 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

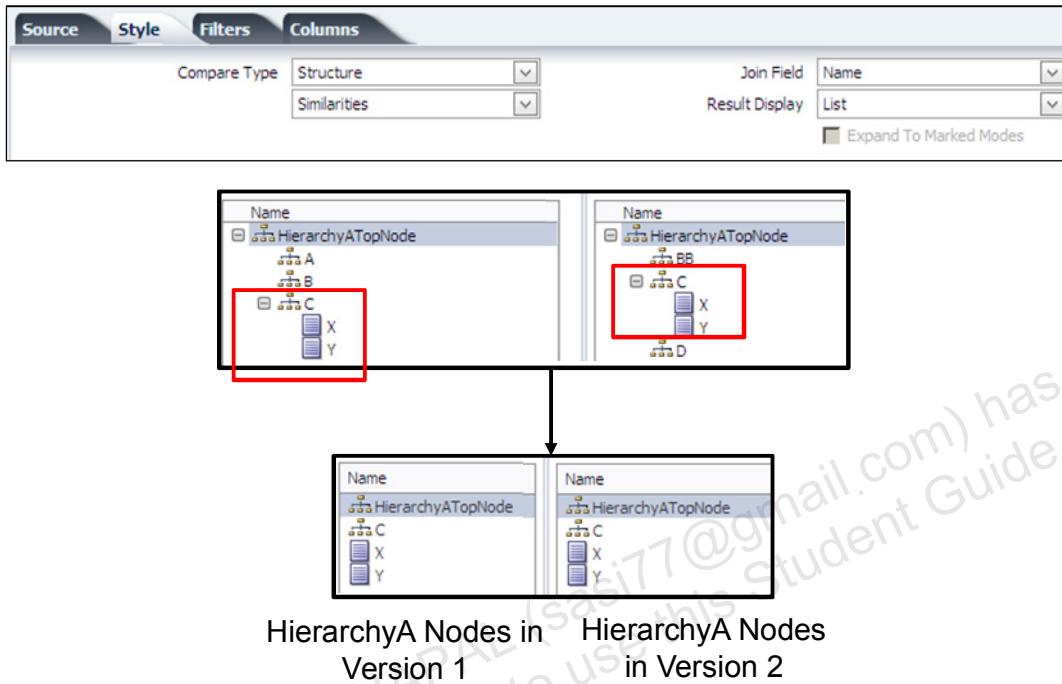
In the slide example, the visual compare enables you to view HierarchyA nodes in version 1 next to HierarchyA nodes in version 2.

To create visual compares:

1. On the Home page, select the **Compare** task group.
2. On the toolbar, click the New Compare button ().
The New Compare tab is displayed.
3. On the left side of the tab, select the first version, a hierarchy, and a compare top node.
4. On the right side of the tab, select the second version, a hierarchy, and a compare top node.
5. Select the **Style** tab.
6. In the Compare Type drop-down list, select **Visual**.
7. On the toolbar, click the Run button () to run the compare.
The results are displayed.

Creating Structure Compares

Example: Structure compare that finds similarities



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you create a structure compare, you configure the Source, Style, Filters, and Columns tabs in the Compare task group. Consider the following:

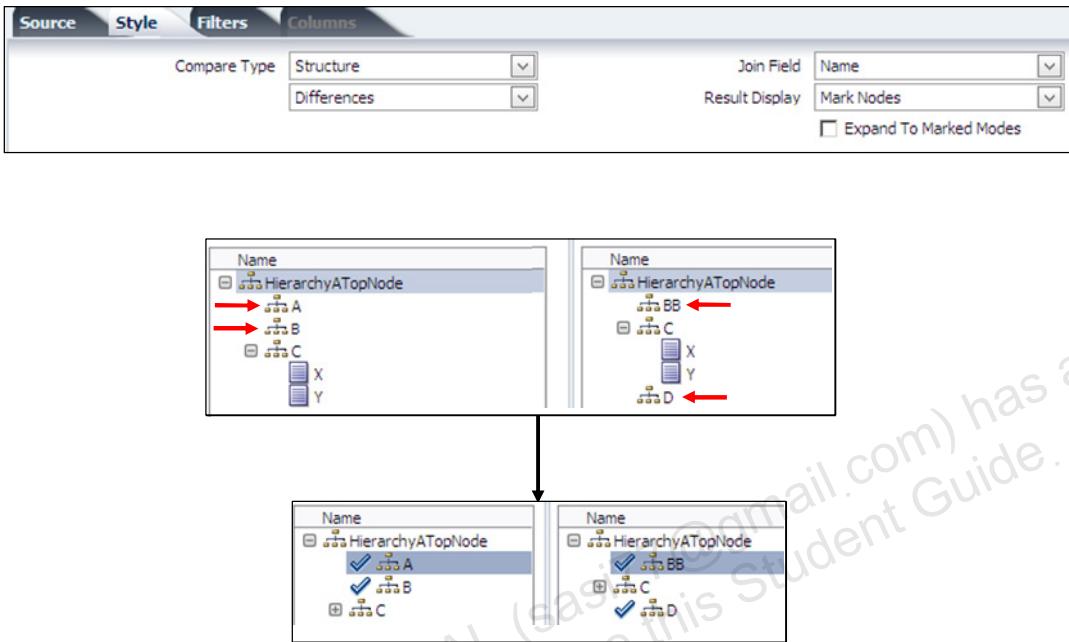
- On the Style tab, you select a join property. By default, a structure compare compares nodes from each hierarchy that has the same name. You can also join nodes based on other properties. For example, if a property named SortCode has a unique key in the specified hierarchies, you can use this property to identify differences between nodes having the same SortCode in each hierarchy.
- On the Filters tab, you select or create a query to filter the comparison results. You can also filter leaf, limb, or both node types.
- On the Columns tab, you select properties to display as columns in the results. Column selections are runtime parameters and are not saved with the compare. You can automatically include properties being compared or used for filtering.

In the slide example, a structure compare finds similarities between HierarchyA nodes in versions 1 and 2. It shows that nodes C, X, and Y are in both versions. The results are displayed in a list.

To create structure compares:

1. On the Home page, select the **Compare** task group.
2. On the toolbar, click the New Compare button ().
The New Compare tab is displayed, and the Source subtab is displayed by default.
3. Select the versions and hierarchies to compare.
4. Select and configure the **Style** tab:
 - a. In the Compare Type drop-down list, select **Structure**.
 - b. Below the Compare Type drop-down list, select **Similarities or Differences**.
 - c. In the Join Field drop-down list, select a property by which to map nodes.
 - d. In the Result Display drop-down list, select **List, Mark Nodes**, or **Both**.
5. **Optional:** Select and configure the **Filters** tab:
 - a. Select or create a query, and select node-matching options.
 - b. In the Limb/Leaf Filter drop-down list, select **Leaf**, **Limb**, or **Both**.
6. **Optional:** Click and configure the **Columns** tab:
 - a. Add properties from the Available list to the Selected list.
 - b. Order the properties as needed.
7. On the toolbar, click the Run button () to run the compare.
The results are displayed.
8. Save the compare.

Example: Structure Compare to Find Differences



ORACLE

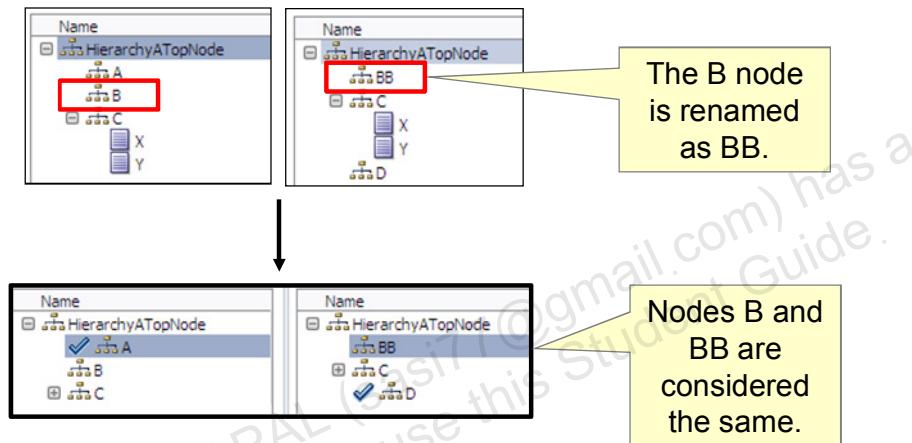
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the slide example, a structure compare finds differences between HierarchyA nodes in versions 1 and 2. It shows that nodes A and B are only in version 1, and nodes BB and D are only in version 2. The results are displayed as marked nodes.

Creating Rename Compares

Examples of rename compares:

- When showing differences, nodes B and BB are not marked.
- When showing similarities, nodes B and BB are marked.



ORACLE

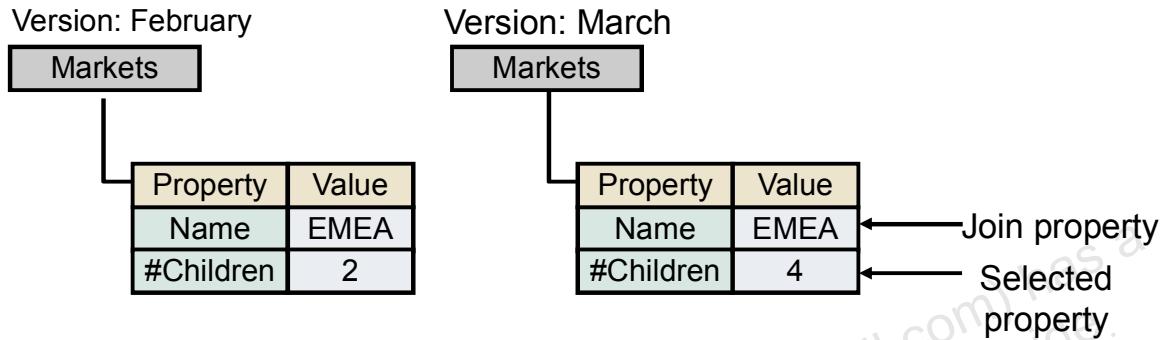
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the slide example, a rename compare finds similarities between HierarchyA nodes in versions 1 and 2. Node B in version 1 is renamed as node BB in version 2. In the rename compare with the similarities options selected, nodes B and BB are regarded as the same and are, therefore, listed. The rename compare joins on the Node ID to allow nodes that have been renamed (but have the same Node ID) to be treated as the same node. In a rename compare with the differences option selected, those two nodes are regarded as the same and are, therefore, not listed.

Note: The steps for creating rename compares are identical to creating structure compares, with the following exception: You select the Rename compare type on the Style tab.

Creating Property Compares

The property compare analyzes nodes for property value differences.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the slide example, the #Children property for the EMEA node is compared in the February and March versions of the Markets hierarchy. In the February version, the #Children property is equal to 2. In the March version, the #Children property is equal to 4. In a property compare that joins on Name, the resulting node list consists of EMEA nodes with their #Children values.

To create property compares:

1. On the Home page, select the Compare task group.
2. On the toolbar, click the New Compare button ().

The New Compare tab is displayed, and the Source subtab is displayed by default.

3. Select the versions and hierarchies to compare.

4. Click and configure the **Style** tab:
 - a. In the Compare Type drop-down list, select **Property**.
Below the Compare Type drop-down list, Differences is automatically selected and you cannot change it.
 - b. In the Join Field drop-down list, select a property that is used to match nodes whose properties will be compared.
 - c. In the Result Display drop-down list, select **List, Mark Nodes**, or **Both**.
5. Click and configure the **Filters** tab:
 - a. Select properties whose values you want to compare, and add them to the Selected list.
Note: Select property categories to facilitate the selection process.
 - b. **Optional:** Select or create a query, select node-matching options, and filter leaf and limb nodes.
6. Click and configure the **Columns** tab:
 - a. Select properties that you want to display in the query results, and add them to the Selected list.
 - b. **Optional:** Select **Include Compare/Filter Properties**.
7. On the toolbar, click the Run button ().
The results are displayed.
8. Save the compare.

Working with Query and Compare Results

- You can configure query and compare results to be displayed in a list, in a marked tree, or in both.
- Global query results are displayed only in a list.
- Local query results are displayed as a single list and tree.
- Compare results are displayed as two lists and trees (side by side).



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Nodes that meet query or compare criteria are displayed in a list on the List tab. The Tree tab displays the structure of the hierarchy with nodes that meet query or compare criteria marked in the tree.

To work with nodes in lists:

Perform one or more of the following actions:

- In the list, select a node to view its properties on the Properties tab. For local query and compare results, you can edit the properties of the selected node.
- Click the Go To Node button () to the right of a node to navigate to a node in the tree.
- To insert a node into another hierarchy or copy its properties, select a node in the list, and then select **Take/Copy Properties** from the menu to add the node to the Clipboard. You can add multiple nodes from the list to the Clipboard.

To work with nodes in trees:

Perform one or more of the following actions:

- On the Tree tab, click the “Expand tree to specified level” button () to expand the tree to display all marked nodes.
- Click the “Move to Next Marked Node” button () to navigate from one marked node to the next.
- For compare results, right-click a node in one tree, and select **Synchronize** to locate the selected node in the other tree.
- Drag nodes from one tree to another to resolve differences between the hierarchies found by a compare.

To modify query or compare criteria and rerun operations:

- Click the “Return to Query Wizard” or “Return to Compare Wizard” button () to close the query or compare results and return to the wizard.

Summary

In this lesson, you should have learned how to:

- Search for nodes
- Create queries
- Create visual, structure, rename, and property compares
- Work with query and compare results

Quiz

To find similar nodes between two hierarchies, which one of the following tasks do you perform?

- a. On the Find Nodes tab in the Browse task group, define a search criteria to find similar nodes.
- b. Create a rename comparison in the Compare task group.
- c. Create a query in the Query task group.
- d. Create a structure comparison in the Compare task group.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz

Rename comparisons treat renamed nodes as being different for comparison purposes.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

When comparing properties, you can select a property that joins nodes whose properties you want to analyze.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Updating Data with Action Scripts

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe action scripts
- Create flat files for action scripts
- Load action scripts from flat files
- Run action scripts and review results
- Download action script results
- Automate changes in versions with action scripts from transaction logs
- Model nodes by using action scripts



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Action Scripts

- Action scripts enable you to process bulk sets of changes to an existing version of data.
- Each record within an action script represents a separate action.
- Action scripts are particularly useful when you need to perform the same set of actions for multiple versions, hierarchies, or nodes.
- Sources for action scripts include flat files, transaction logs, and nodes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Action scripts enable you to process bulk sets of incremental changes in an automated fashion. Each record within a script represents a separate action to be performed and processed separately from other actions. You can group actions of different types in the same script. Action scripts are particularly useful when you need to perform the same set of actions for multiple versions, hierarchies, or nodes.

You can load action scripts from flat files, generate them from transaction logs, or create them from node models. In the Script task group, you can load and run only one action script at a time.

Creating Flat Files for Action Scripts

Action	Param1	Param2	Param3	Param4	Param5	Param6	Param7
AddHier	Ver	Hier	Desc	TopNode	Desc	ResultDesc	
DeleteHier	Ver	Hier	ResultDesc				
Add	Ver	Hier	Name	Parent	Leaf	ResultDesc	Domain
Insert	Ver	Name	Parent	ResultDesc			
AddInsert	Ver	Hier	Name	Parent	Leaf	ResultDesc	Domain
ChangeProp	Ver	Hier	Name	PropLabel	Value	ResultDesc	
RemoveProp	Ver	Hier	Name	PropLabel	ResultDesc		
LockProp	Ver	Hier	Name	PropLabel	Lock	ResultDesc	
PropCompare	Ver	Hier	Name	PropLabel	Value	ResultDesc	

Ver = Version name

Hier = Hierarchy name

Name = Node name

Leaf: 0 = Limb, 1 = Leaf



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create action script files to process incremental changes that originate from an external source. Action script files have a tabular format that consists of rows and columns. Each row defines an action to be performed in Data Relationship Management. Each row is processed as a separate transaction and is performed sequentially. A status message is returned for each row. Details are displayed in the Return Value column.

In the slide example, nine different actions and their parameters are shown. To create a text file based on this example, you list each row from the table, separating column data with a delimiter, such as a comma. To include a comment, use the following format:

Rem: Remark text

Required Format for Flat Files

- The first column in the file must contain the name of the action to be performed. If this column does not contain a recognized action, the row is ignored and is not loaded. A Remark action is available to enable you to add comments to the file.
- The remaining columns contain parameters that define each action. Typically, for most actions, the first three columns of parameters contain the names of the Data Relationship Management version, hierarchy, and node upon which the action is to be performed. The last four columns of parameters show other parameters relevant to each action.

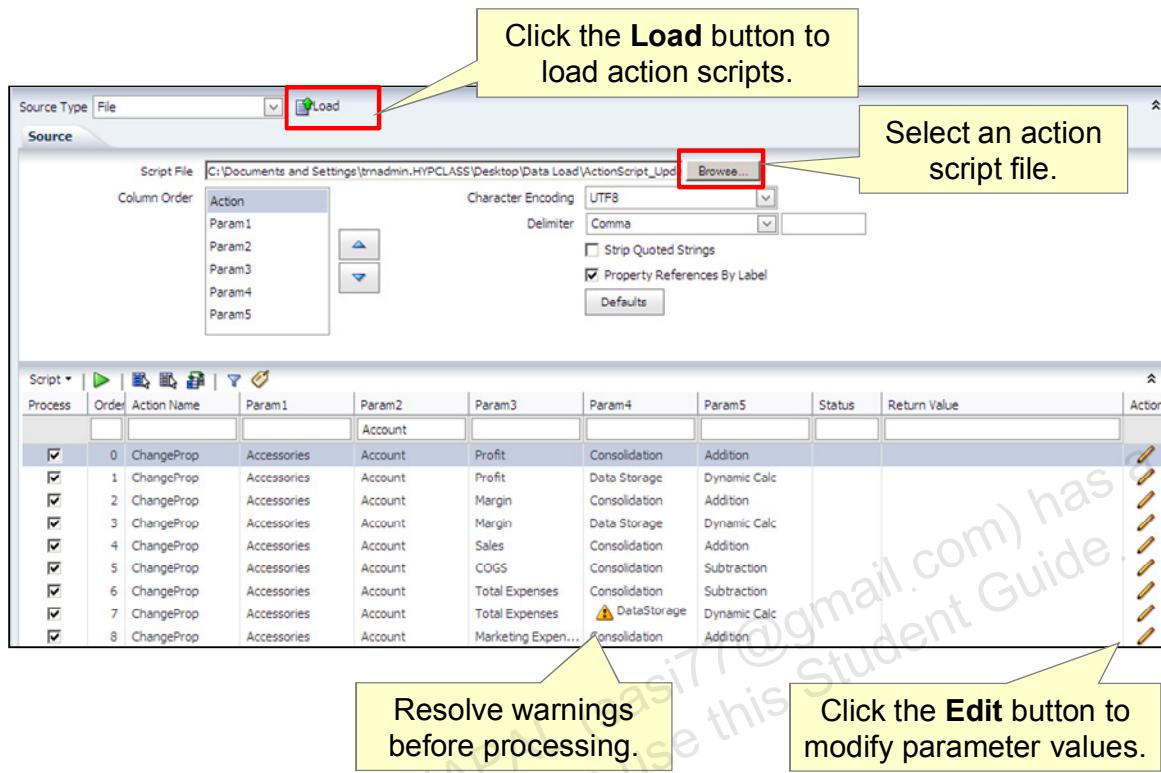
- Actions that refer to properties can identify properties by their label or by name. If the property label does not uniquely identify a property, the system will require you to select a single property to resolve the situation.
- After the action script is processed, column Status is populated by Data Relationship Management with the status. If a particular action fails, the reason for a failure is displayed below the action.

Commonly Used Actions

The following table describes the most commonly used actions:

Action	Purpose
AddHier	Adds a hierarchy to a version
DeleteHier	Deletes a hierarchy from a version
Add	Adds the specified node to the specified hierarchy. Use this action only for a node that does not already exist in the specified version.
Insert	Inserts the specified node into the specified hierarchy. Use this action only for nodes that already exist in another hierarchy in the specified version.
AddInsert	Adds the specified node if it does not already exist in the specified version; otherwise, the node is inserted.
ChangeProp	Updates the value of the given property of the specified node
RemoveProp	Removes (clears) the value of the given property of the specified node. The property itself is not deleted because many other nodes may use this property.
LockProp	Locks or unlocks the given property of the specified node
ChangeProp	Updates the value of the given property of the specified node
PropCompare	Returns True if the value of the given property of the specified node is equal to the value listed in the script; otherwise, returns False

Loading Action Scripts from Flat Files



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

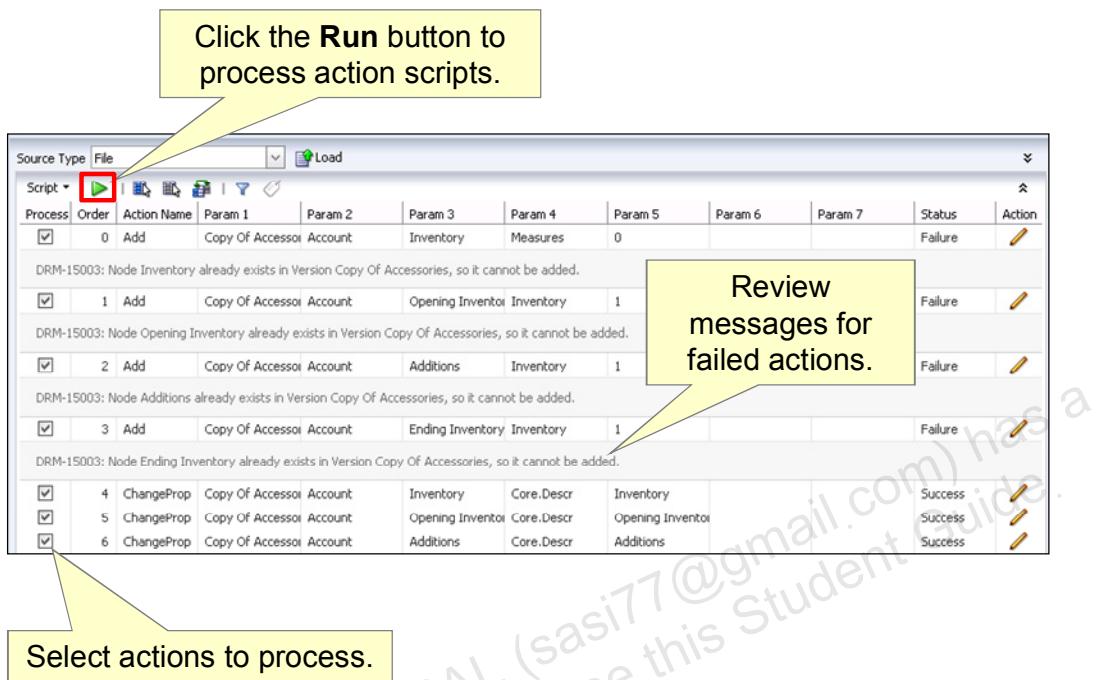
In the Script task group, you load the flat file that contains the action scripts. After loading, edit rows that display warning symbols to ensure proper processing.

To load action scripts from flat files:

1. On the Home page, select the **Script** task group.
2. In the Source Type drop-down list, select **File**.
3. Next to the Script File box, click **Browse**.
The “Choose File to Upload” dialog box is displayed.
4. Browse and select the flat file to use for the action script, and click **Open**.
5. **Optional:** In the Column Order list, click the Move Up button () or the Move Down button () to customize the order of the six required columns from the flat file.
6. In the Character Encoding drop-down list, select a character encoding option (for example, Unicode (UTF8)).
7. In the Delimiter drop-down list, select the column delimiter that is used in the flat file (for example, Comma).
8. **Optional:** Select **Strip Quoted Strings** to remove quotation marks from values in the source file.

9. **Optional:** Clear **Property References By Label** if the property names are fully qualified in the script file.
10. **Optional:** Click **Defaults** to reset all options to the default settings.
11. Click **Load**.
Actions and parameter values are displayed.
12. Perform one or more of the following actions:
 - On a row, click the Edit button () to modify parameters for that row, and then click the Update button () to save changes.
 - In the Script drop-down list, select **Substitute Versions** to change a version. Select the version to be replaced and the version to replace it with, and click **OK**.
 - Double-click a column header to sort rows alphabetically.
 - On the toolbar, click the Toggle Filter button () to filter one or more columns. Enter filter values in the empty boxes above the scripts.

Running Action Scripts and Reviewing Results



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can run any number of actions within a script at one time. When processing is completed, you can review information in the following locations:

- The Status column displays the status of each action (Unprocessed, Failure, or Success).
- The Return Value column displays details about actions with failed statuses.

To run action scripts:

1. After you load an action script, select actions in one of the following ways:
 - In the Process column, select the check box for each action that you want to run.
 - In the Script drop-down list, choose **Select All** to select all actions.
2. On the toolbar, click the Run button () to process the selected actions.
 - A warning message (failed rows) or a confirmation message (successful script processing) is displayed.
3. Click OK.
 - The results of each action are displayed.
4. Review the Status and Result Value columns.
5. **Optional:** Edit and rerun any failed actions. You can filter on the Status column for failed actions and select only those actions that you want to rerun.

Downloading Action Script Results

You can download action script results in the following formats:

- Loadable action script file
- Comma-separated or tab-separated file
- PDF, RTF, or XLS report



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After processing, you can download the results to a file in the following formats:

- Loadable action script file
- Comma-separated or tab-separated file
- PDF, RTF, or XLS report

To download results:

1. **Optional:** Filter or order the rows of the script to control which rows are downloaded:
 - On the toolbar, click the Toggle Filter button (), enter a filter criteria, and press **Enter**.
 - Click a column header to sort the rows based on that column.
2. In the Script drop-down list, select **Download**, and then select a download option (for example, Download as Comma-separated).

The File Download dialog box is displayed.

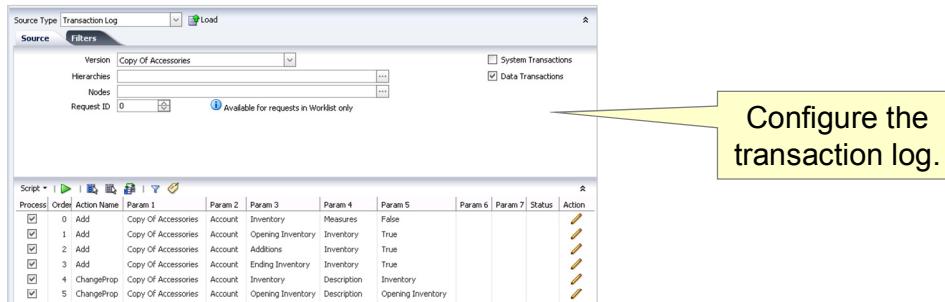
3. Click **Save**.

The Save As dialog box is displayed.

4. Browse to a location, enter a file name, and click **Save**.

Automating Version Changes with Action Scripts from Transaction Logs

1. Load an action script from a transaction log



2. Substitute version



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

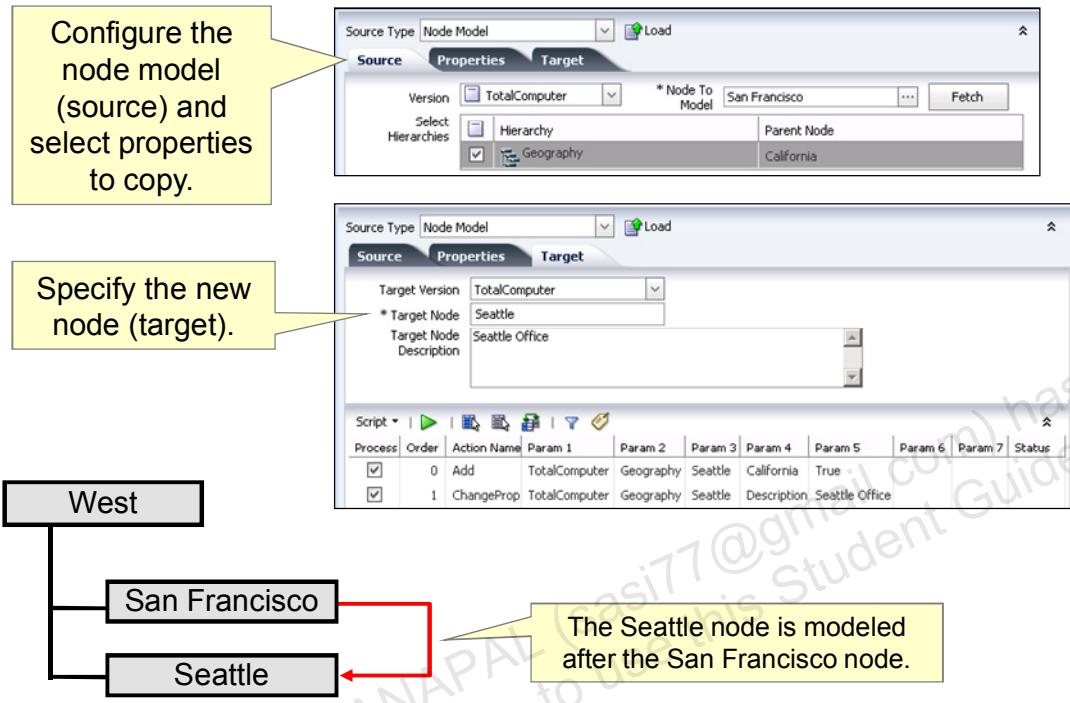
Action scripts created from the transaction log are often used to automate changes made manually in one version to another version. This is useful when multiple versions of the same hierarchies are available for edit at the same time and users are only making certain changes in one of the versions but the changes need to be synchronized to the other version as well.

To load an action script from a transaction log:

1. On the Home page, select the **Script** task group.
2. In the Source Type drop-down list, select **Transaction Log**.
3. On the Source tab, perform the following actions:
 - a. Select the source version, hierarchies, and nodes.
 - b. Select whether to return system transactions, data transactions, or both.
4. On the Filters tab, set the following options:
 - a. **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)
- b. **Actions:** Click to specify transaction log action types to filter.
 - c. **Properties:** Click to specify a list of properties to filter.
 - d. **Users:** Click to specify users to filter.
 - e. **Include Child Nodes:** Select to include child nodes for nodes returned by the query.
 - f. **Include Shared Nodes:** Select to include shared nodes for nodes returned by the query.
 - g. **From Date:** Select and enter a start date.
 - h. **To Date:** Select and enter an end date.
 - i. **Filter to Current Session:** Select to filter transactions to those performed during the current session.
 - j. **From Transaction:** Select and enter a start transaction ID.
 - k. **To Transaction:** Select and enter an end transaction ID.
 - l. **Max Records:** Enter a value to limit the number of transactions displayed.
5. **Click Load.**
Actions and parameter values are displayed.
- To substitute version:**
1. In the Script drop-down list, select **Substitute Versions** to change a version.
The Script Version Substitution dialog box is displayed.
 2. In the Substitute drop-down list, select the version to replace with, and click **OK**.
The substitute version is listed in the Param1 column.

Modeling Nodes by Using Action Scripts



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the Script task group, you can create an action script to define a new node from the model of an existing node. A node model includes the hierarchies and properties for a node in a version. Creating a script from a node model is useful when the new node must be located in the same hierarchies and have the same properties in those hierarchies as the existing node. You can specify which hierarchies and properties are included in the action script.

After loading the action script for processing, you can edit the parameters of the script to handle any differences in relationships or property values between the new node and the existing node.

To model nodes:

1. In a hierarchy tree, select a node to model.
2. Click the **Shortcuts** tab.
3. Expand **Script**, and select **Model After**.
The Source tab in the Script task group displays information from the model node.
4. Click the **Properties** tab, and configure the properties to be copied to the new node.
 - **Optional:** Select a subset of hierarchies from the node model to be included in the script.
 - Select global properties from the existing node and local properties for each hierarchy.

Note: You must select global and local properties.
5. Click the **Target** tab, and perform the following actions:
 - a. Select a target version.
 - b. Enter a name and an optional description for the new node.
6. Click **Load** and review the scripts that will create the node.
7. **Optional:** Modify the scripts (for example, change a parameter value).
8. On the toolbar, click the Run button (▶).
A confirmation dialog box is displayed.
9. Click **OK**.
The results of each script action are displayed in the Status column.
10. Verify that the node has been created by viewing it in its hierarchy.

Summary

In this lesson, you should have learned how to:

- Describe action scripts
- Create flat files for action scripts
- Load action scripts from flat files
- Run action scripts and review results
- Download action script results
- Automate changes in versions with action scripts from transaction logs
- Model nodes by using action scripts



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

In the Script task group, you can make bulk changes to versions and hierarchies by performing node and property manipulations.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Modeling nodes enables you to perform which of the following tasks?

- a. Add a node to the hierarchy from where you are modeling.
- b. Insert nodes into other hierarchies under the same parent as the model node.
- c. Copy global properties of the model node to the new node.
- d. Copy local properties of linked model nodes to the new node.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Quiz

To create a flat file for an action script, which of the following tasks do you perform?

- a. Create columns of data, separating data with a delimiter (such as a comma).
- b. Make the first column contain the name of the action that is to be performed.
- c. Make all columns, other than the first column, contain the parameters that define each action.
- d. Include an exit action at the end of the file.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

All actions that are loaded into the action script must be run.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Blending Versions

8

ORACLE®

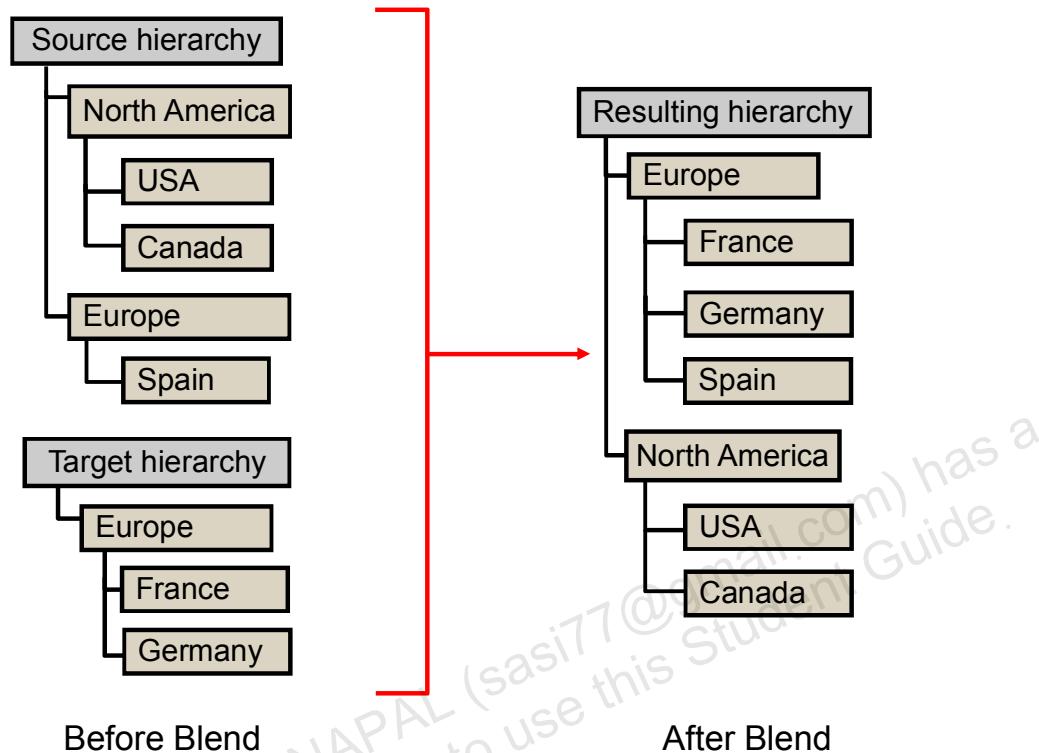
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe blending versions
- Open, copy, and delete blenders
- Create blenders
- Run blenders

Blending Versions



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Data Relationship Management, you can combine data from two versions into a single version by using a blender. A blender compares hierarchies between versions and propagates differences as changes to the target version of the blending process. You can select hierarchies to be blended and the types of actions to be performed. You can blend hierarchies directly into the target version or you can create a new version that contains the combined results.

With blenders, you can perform the following tasks:

- Create new hierarchies.
- Map top nodes from a source hierarchy to top nodes in a target hierarchy.
- Insert, move, or remove nodes in existing hierarchies.
- Activate and promote nodes (leaf to limb).
- Update global and local properties.
- Update validation and access assignments.

Use blenders to combine data from imported versions or from manually created versions. Users with the Application Administrator, Data Manager, or Data Creator role can create and manage blenders.

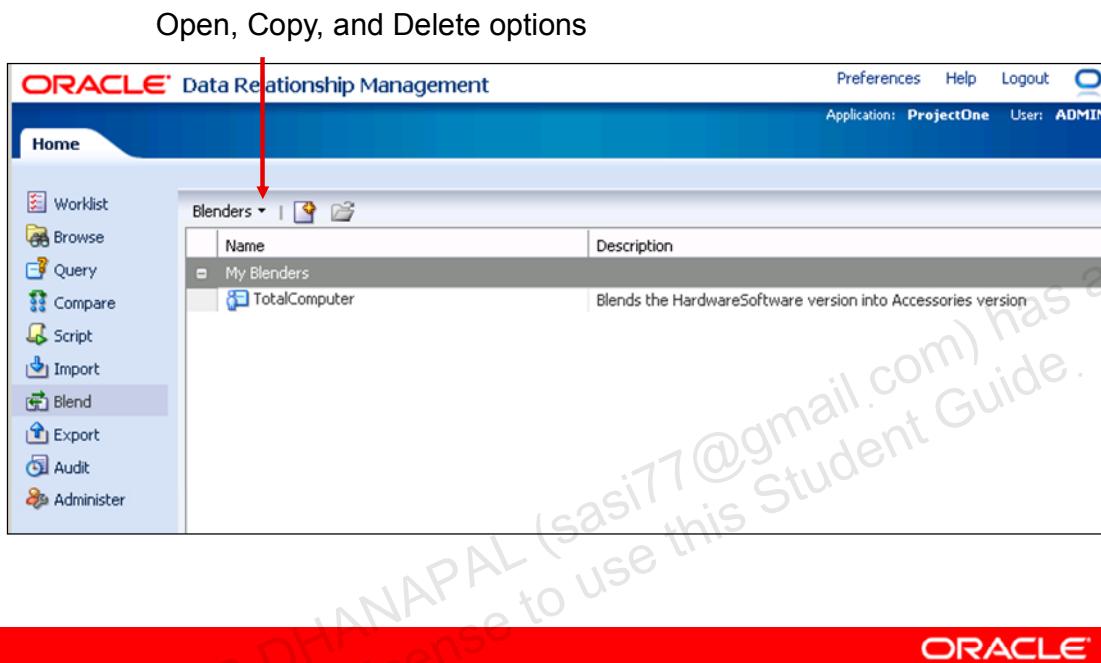
In the slide example, the source hierarchy is blended into the target hierarchy.

- The source hierarchy consists of two limb nodes: North America and Europe.
 - The North America node has two leaf nodes: USA and Canada.
 - The Europe node has one leaf node: Spain.
- The target hierarchy consists of one limb node, Europe, which has two leaf nodes: France and Germany.
- The resulting hierarchy consists of two limb nodes: Europe and North America.
 - The Europe node consists of three leaf nodes: France, Germany, and Spain.
 - The North America node consists of two leaf nodes: USA and Canada.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Opening, Copying, and Deleting Blenders

You open, copy, and delete blenders in the Blend task group.



To access the Blend task group:

- On the Home tab, select the **Blend** task group.

To open blenders:

- In the Blend task group, select a blender.
- In the Blenders drop-down list, select **Open**.

The blender is displayed on a new tab with five subtabs (Source, Style, Filters, Properties, and Target). The Source subtab is displayed by default.

To copy blenders:

- In the Blend task group, select a blender.
- In the Blenders drop-down list, select **Copy**.
The Copy Blender dialog box is displayed.
- Enter a name and an optional description for the blender.
- In the Object Access Group drop-down list, select **User, Standard, System**, or a custom object access group.
- Click **OK**.

To delete blenders:

1. In the Blend task group, select a blender.
2. In the Blenders drop-down list, select **Delete**.
The Confirm Delete dialog box is displayed.
3. Select **Delete the Blender**.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Creating Blenders

In the blender wizard, you configure the following tabs:

- Source
- Style
- Filters
- Properties
- Target



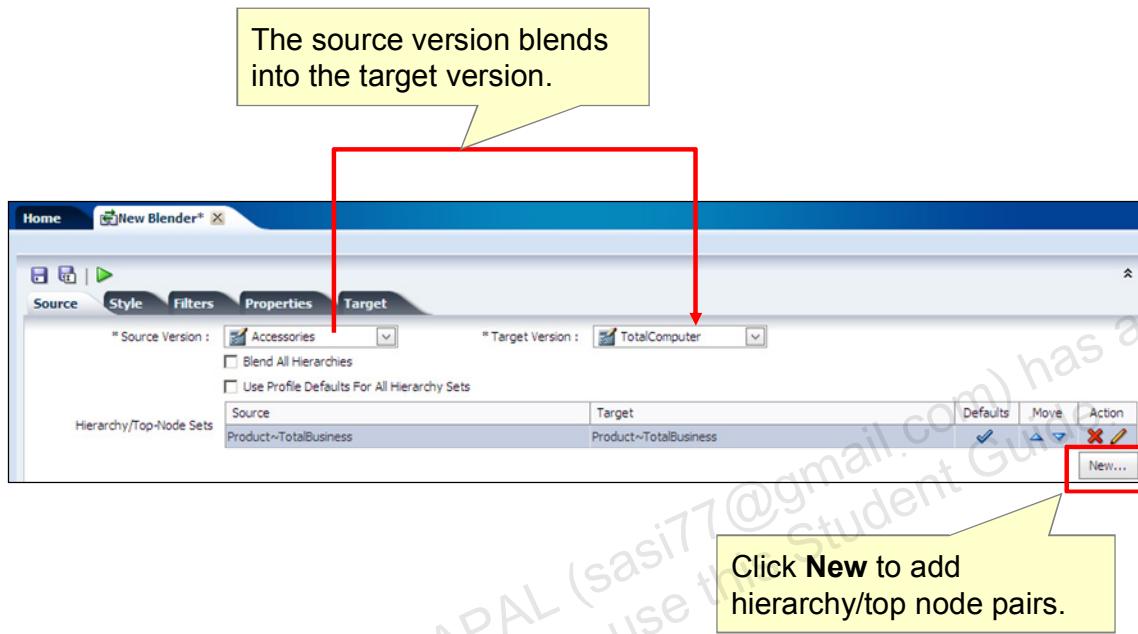
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the blender wizard to define the parameters and criteria for a blender. The wizard consists of the following five tabs to guide input: Source, Style, Filters, Properties, and Target. Although its tabs are ordered to gather input in a sequence, you can navigate between the tabs at any time.

To create blenders:

1. On the Home page, select the **Blend** task group.
 2. On the toolbar, click the New Blender button ().
The blender wizard is displayed.
 3. Configure the tabs.
 4. Save and run the blender.
- See configuration, save, and run procedures at the end of this lesson.

Configuring Blender Versions and Hierarchies



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Source tab in the blender wizard, you select source and target versions for the blender. You can blend all hierarchies in the selected versions, or you can select hierarchy/top node pairs to blend. These mappings from the source to the target indicate which hierarchy top nodes in the target are to receive values from the source. Depending on the blender settings, you can create top nodes in the target or you can map top nodes in the target to existing top nodes.

Default values are set for the blender profile. You can use the profile default values for all hierarchy/top node sets or only for specific sets. If you do not use the profile default values, you can select different values for each set.

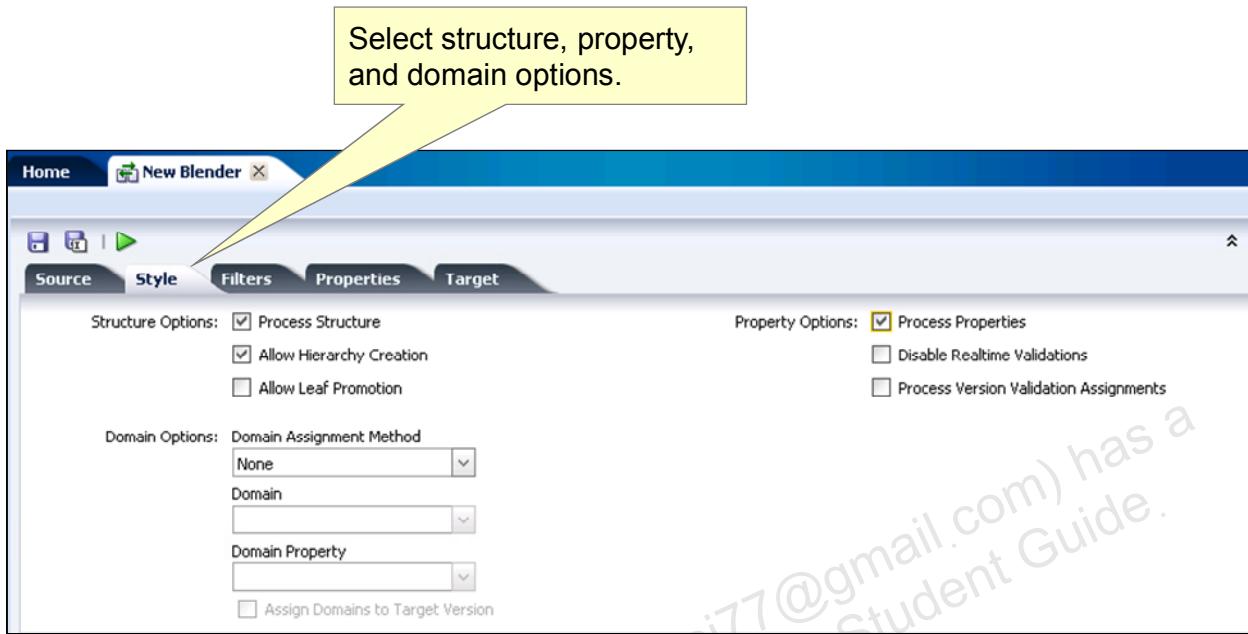
The order of hierarchy/top node sets controls the order in which the hierarchies are blended. The results of the blend operation may be different based on the order in which changes are propagated to the target version.

To configure the source:

1. In the blender wizard, click the **Source** tab.
2. In the Source Version drop-down list, select the version that contains the data that you want to blend into another version.
3. In the Target Version drop-down list, select the version into which you want the source data to be blended.

4. **Optional:** Select **Blend All Hierarchies**.
5. **Optional:** Select **Use Profile Default For All Hierarchy Sets**.
6. **Optional:** Define hierarchy/top node sets:
 - a. Clear **Blend All Hierarchies**.
 - b. Click **New**.
The Add Blender Hier-Set dialog box is displayed.
 - c. **Optional: Select Use Profile Defaults.**
Note: When you select Use Profile Defaults For All Hierarchy Sets, the Use Profile Defaults setting in the Add Blender Hier-Set dialog box is automatically selected for you and you cannot clear it. Clear Use Profile Defaults For All Hierarchy Sets if you want to clear the Use Profile Defaults setting.
 - d. On the Source tab, select a hierarchy from the source version and a top node (any limb node). You can search for a node, if needed.
 - e. On the Target tab, perform one of the following actions, and click **OK**:
 - Select **Specified**, and then select **Use Same As Source**, or enter a hierarchy and top node name.
 - Choose **Selected**, and select a hierarchy and top node.
The hierarchy/top node set is listed.
 - f. Repeat steps b through e to add more hierarchy/top node sets.
 - g. **Optional:** Select a hierarchy/top node pair, and click the Delete button () to remove it, or click the Edit button () to modify it.
 - h. **Optional:** Select a hierarchy/top node pair, and click the Move Down button () or the Move Up button () to reorder it.

Configuring Blender Processing Options



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Style tab in the blender wizard, you select structure, property, and domain options for a blender.

You can select the following structure options:

- **Process Structure** controls the defaults for the structural actions selected on the Filters tab. Enabling this option sets all structural actions on the Filters tab to True.
- **Allow Hierarchy Creation** creates hierarchies in the target version that exist only in the source version. If you do not select this option and hierarchies exist in the source version but not in the target version, an error occurs.
- **Allow Leaf Promotion** promotes a leaf node in the target version to a limb to insert children from the source version under it. If a node in the source version has children, but its corresponding node in the target version is a leaf node, this option enables the blender to promote the node to limb status. The child nodes from the source version are then added to the limb node in the target version.

You can select the following property options:

- **Process Properties** controls whether property values are blended based on the options selected on the Properties tab.

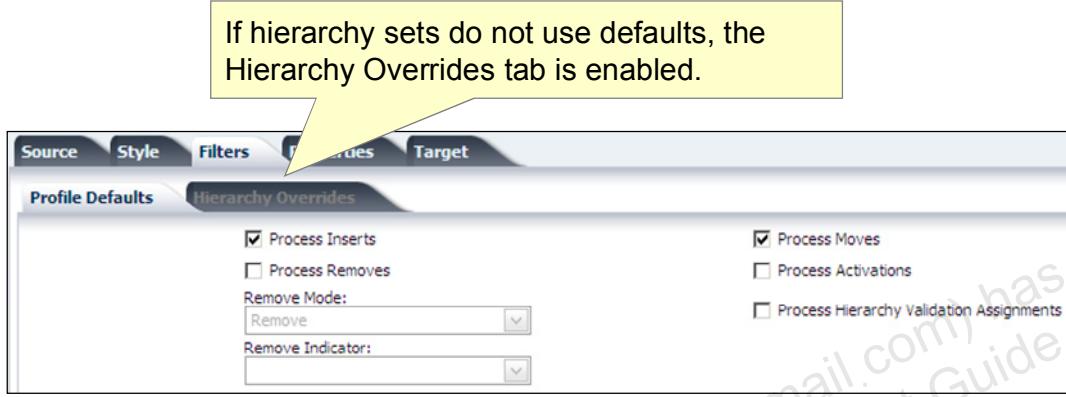
- **Disable Realtime Validations** turns off the real-time validations on the target version prior to blending. This can improve performance of the blending process. Batch validations can be run after the blend process to verify the results.
- **Process Version Validation Assignments** processes version-level validation assignments from the source version to the target version.

Note: Validations are covered in a later lesson.

You can select the following Domain options:

- **Domain Assigned Method** specifies how nodes are assigned to domains. You can select one of the following options:
 - **None** if nodes should not be assigned to any domain.
 - **Select a Domain** from the Domain drop-down list if you want to specify a single domain to which all nodes being blended are assigned if the nodes are qualified for the domain.
 - **Select a Domain Property** from the Domain Property drop-down list if you want to use the selected property to identify the domain to which each node being blended is assigned.
- **Optional:** Select **Assign Domains to Target Version** to assign a new domain used in the source version to the target version.

Configuring Blender Filters



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Filters tab in the blender wizard, you can specify individual types of changes to be propagated from the source to target version. You can select options for profile defaults (on the Profile Defaults tab) and hierarchy overrides (on the Hierarchy Overrides tab). Configure profile defaults if you selected Use Profile Defaults For All Hierarchy Sets on the Source tab. If any hierarchy sets do not use defaults, then the Hierarchy Overrides tab is enabled and you can provide hierarchy overrides for each hierarchy set.

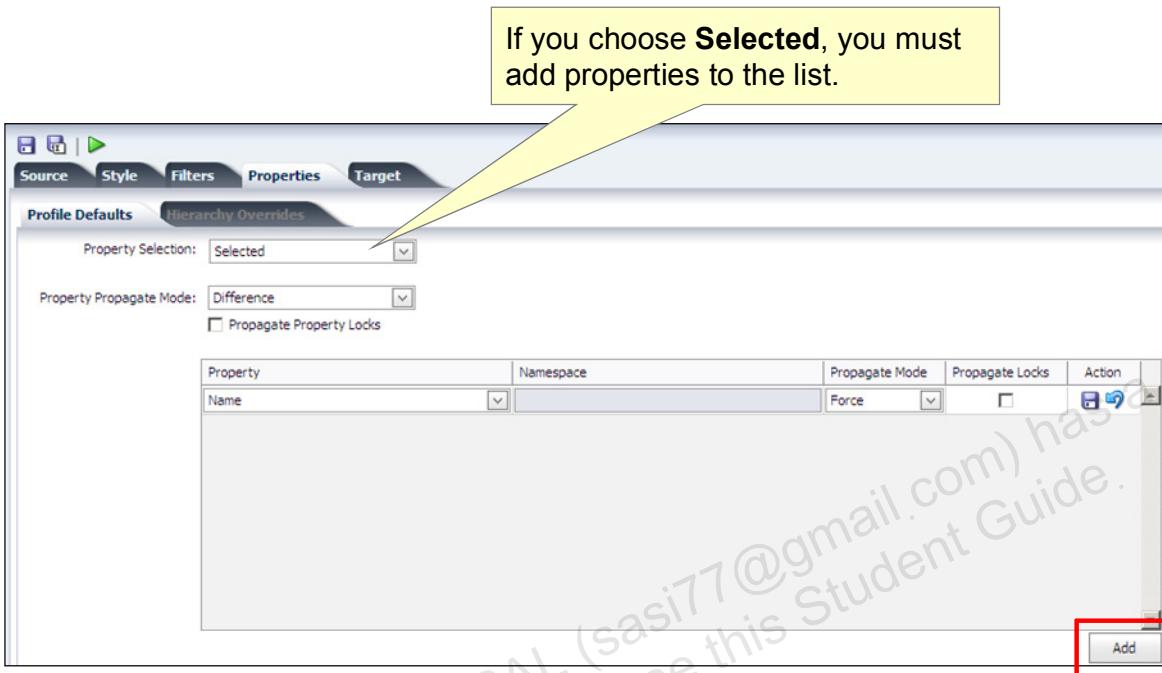
You can select the following filter options:

- **Process Inserts** inserts any nodes that exist only in the source hierarchy into the target hierarchy.
- **Process Removes** removes nodes that exist only in the target hierarchy. If you select this option, you must select one of the following options for the remove mode:
 - Select **Remove** to remove the node during blending.
 - Select **Mark** to mark the node for later removal, and select a remove indicator property that will contain the value showing that the node is flagged to be removed. You must create this local Boolean property beforehand.
- **Process Moves** moves nodes that have a different parent in the source version to that parent in the target version.

- **Process Activations** processes node activations (inactivate and reactivate) from the target version to the source version.
- **Process Hierarchy Validation Assignments** processes the hierarchy-level validation assignments from the source version to the target version.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Configuring Blender Properties



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Properties tab in the blender wizard, you select an option for blending properties, select a mode to determine how the property is populated in the target, and choose whether to propagate property locks (if you want inheritance locks in the source version to be blended to the target version).

You can select the following options for blending properties:

- **All Excluding Val/Access** blends all properties, excluding properties used for node-level validations and for node access group assignments.
- **None** does not blend any properties.
- **Selected** blends individually selected properties.

The following property propagation modes are available:

- **Force** overwrites the target property value with the source property value for every node in the target, regardless of whether that property is defined for that node, is inherited, or is derived.
- **Difference** overwrites the target property value with the source property value for any node where the source and target property values are different, regardless of how that property is populated in the target version.

- **Defined** overwrites the target property value with the source property value for any node where the property is populated in the source version.

Note: You can blend derived property values only for derived properties that can be overridden.

To configure the properties:

1. In the blender wizard, click the **Properties** tab.
2. On the Profile Defaults tab, in the Property Selection drop-down list, select an option for blending properties.
If you choose Selected, a list is displayed so that you can add specific properties.
3. In the Property Propagate Mode drop-down list, select a mode.

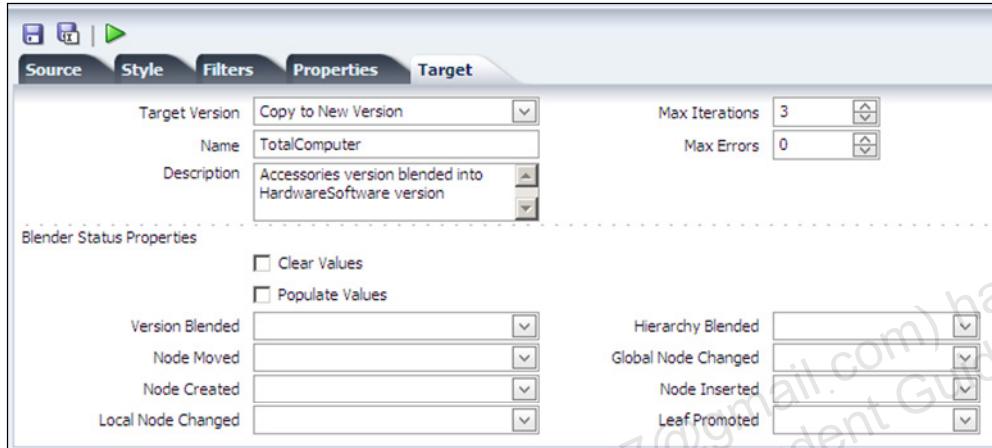
Note: If you did not select **Use Profile Defaults For All Hierarchy Sets** on the Source tab, click the **Hierarchies Overrides** tab, and select modes on it.

4. **Optional:** Select **Propagate Property Locks** if you want inheritance locks in the source to be blended into the target.
5. If you chose Selected in step 2, add properties to the list:
 - a. At the bottom of the Profile Defaults tab, click **Add**.
A property selection row is added to the list.
 - b. In the Property drop-down list, select the default (the first available property alphabetically) or select another property.
 - c. Select a propagate mode.
 - d. **Optional:** Enable propagate locks.
 - e. Repeat steps a through d to add more properties.

When you select a property, its name is removed from the Property drop-down list, and you cannot select it again for blending. The default property becomes the next available property alphabetically.

Note: If you did not previously select the first property in the list, it becomes the default value for the next property selection row.

Configuring Blender Target Options



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Target tab in the blender wizard, you configure the following items:

- **Target Version** is the existing target version or a new version.
- **Max Iterations** is the maximum number of attempts to reconcile the source and target versions. This setting prevents the blender from running indefinitely in case of irresolvable conflicts. The recommended value is 3. Multiple iterations may be necessary to propagate unsuccessful changes due to the order in which hierarchies were processed.
- **Max Errors** is the maximum number of errors that can occur during processing before the blender stops.
- **Blender Status Properties** can be populated during the blending process to identify the nodes that were affected and the types of changes that were made. These properties can be used by queries to report on blender changes after the blend process has finished. You can select the following options for blender status properties:
 - **Clear Values** resets all Blender Status Property values to False during processing.
 - **Populate Values** sets each Blender Status Property to True during processing.
 - **Version Blended** is the property that is set to TRUE if the version is blended (version node level, Boolean-defined property).

- **Node Moved** is the property that is set to TRUE when the node is moved by the blend (local node level, Boolean-defined property).
- **Node Created** is the property that is set to TRUE when the node is created by the blend (global node level, Boolean-defined property).
- **Local Node Changed** is the property that is set to TRUE when the node's local properties are changed by the blend (local node level, Boolean-defined property).
- **Hierarchy Blended** is the property that is set to TRUE when the hierarchy is blended (hierarchy node level, Boolean-defined property).
- **Global Node Changed** is the property that is set to TRUE when the node's global properties are changed by the blend (global node level, Boolean-defined property).
- **Node Inserted** is the property that is set to TRUE when the node is inserted by the blend (local node level, Boolean-defined property).
- **Leaf Promoted** is the property that is set to TRUE when the node is promoted from a leaf to a limb by the blend (global node level, Boolean-defined property).

To configure blender target options:

1. In the blender wizard, click the **Target** tab.
2. In the Target Version drop-down list, select the target and do one of the following:
 - Select **Use Selected Target**, and skip to step 4.
 - Select **Copy to New Version**, and proceed to step 3.
3. Enter a name and description for the new version.
4. In the Max Iterations and Max Errors boxes, enter values.
5. **Optional:** Perform one or both of the following actions:
 - Select **Clear Values** or **Populate Values**.
 - Select properties for one or more blender statuses.

Saving Blenders

Specify the following information:

- Name
- Description
- Object access group
 - User
 - Standard
 - System
 - Custom object access group

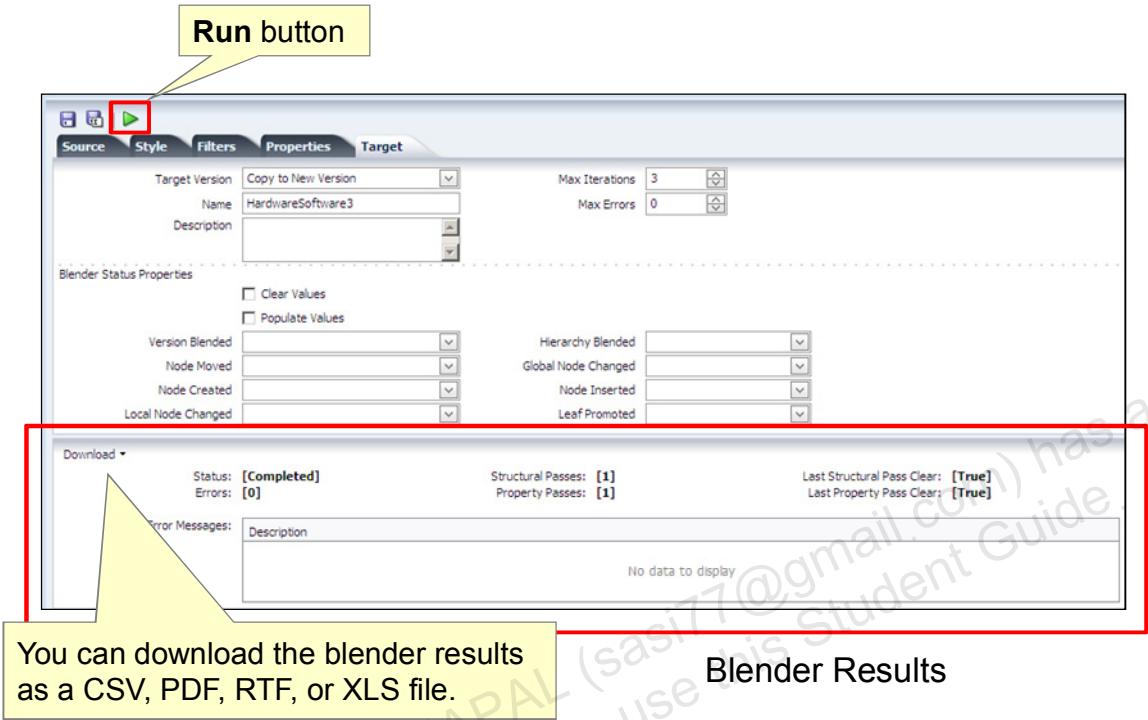
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To save blenders:

1. On the toolbar, click the Save As button ().
The Save As dialog box is displayed.
2. Enter a name and an optional description for the blender.
3. In the Object Access Group drop-down list, select **User**, **Standard**, **System**, or a custom object access group (depending on your user role).
4. Click **OK**.

Running Blenders



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After a blender runs, you are presented with an error count, a completion status, the number of structural and property passes, an indication of whether the last structural and property pass cleared (True or False), and a list of error messages that were generated during the processing.

Note: If you are blending into a new version data is not permanently saved after you run a blender. You must save the new version to make it permanent.

To run blenders:

1. If necessary, in the Blend task group, open a blender, and configure the source and target.
2. On the toolbar, click the Run button ().
The blender runs and processing details are displayed.
3. **Optional:** Save the new version.
 - a. Click the **Home** tab and then the **Browse** task group.
 - b. In the version list, right-click the new version, and select **Save**.
The version is saved, and a check mark is displayed in the Saved column.

Summary

In this lesson, you should have learned how to:

- Describe blending versions
- Open, copy, and delete blenders
- Create blenders
- Run blenders

Quiz

In Data Relationship Management, you can blend elements of two or more versions at one time.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

When blending, you must merge source and target versions into a new version.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

When creating blenders, which one of the following options enables you to create hierarchies in the target version that exist only in the source version?

- a. Process Inserts
- b. Allow Leaf Promotion
- c. Process Structure
- d. Allow Hierarchy Creation



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz

When configuring blenders, you map hierarchy structures in the source version to hierarchy structures in the target version by defining hierarchy/top-node pairs.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Creating Derived Properties

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe formulas and scripts for derived properties
- Explain formula syntax and script constructs
- Create derived properties by using formulas and scripts
- Test formulas and scripts

Derived Properties

Derived properties are calculated by a deriver class:

- Formula (native formula language)

```
IF (Equals (String, PropValue (Core.Level), 1), TopNode,  
Concat (DefaultParent=, ParentPropValue (Core.Abbrev)))
```

- Script (JavaScript)

```
if (node.Level == 1)  
    return ("TopNode");  
else  
    return ("DefaultParent=" + node.ParentNodeAbbrev);
```



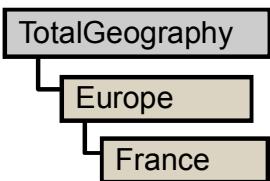
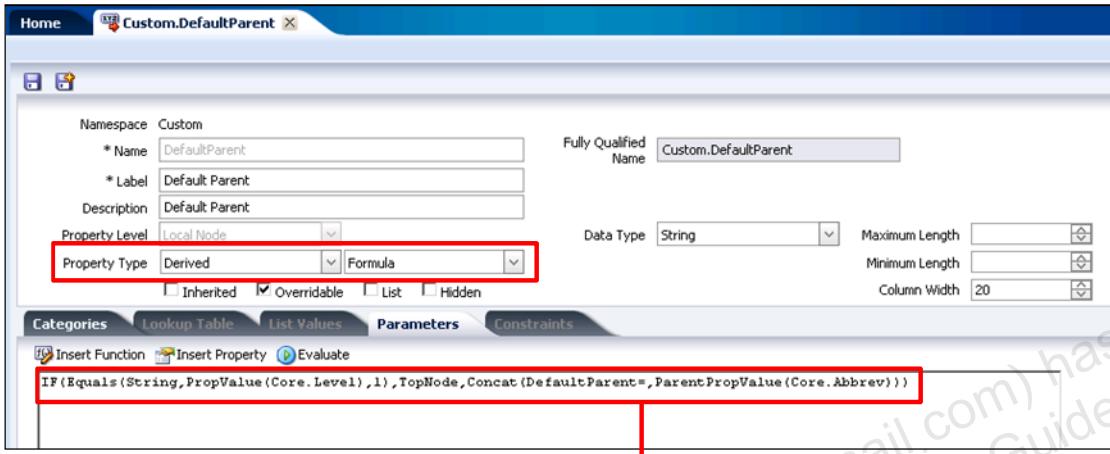
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Derived property type enables you to create properties whose values are calculated by one of the following deriver classes:

- The **Formula** deriver class enables you to define complex logic for derived properties and validations by using a native formula language in Data Relationship Management. Formulas are composed of functions and string literals and must follow specific syntax rules.
- The **Script** deriver class enables you to develop business logic for derived properties and validations by using JavaScript. Scripts provide a more robust and better performing alternative to formulas, using a standard scripting language. Scripts allow for better organization and less complexity of logic through the use of multiple statements, variables, and in-line comments. Scripts also provide support for advanced concepts like looping and regular expressions.

Derived Properties Using Formulas

Property values are derived by a formula.



TotalGeography DefaultParent=TopNode

Europe DefaultParent=TotalGeography

France DefaultParent=Europe

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you create a derived property based on the Formula deriver class, you define a formula to derive the property values. Formulas consist of string literals, functions, or both, as well as their parameters. The parameters can contain further string literals and other functions.

In the slide example, the parameters and formula are displayed for the DefaultParent property. The formula consists of the IF function.

Formula Syntax

- Function names are case-insensitive and are followed by parentheses.
- Parameters can be nested functions, property names, or literals.

```
UpperCase(PropValue(Core.ParentLongName))
```

Nested function Property name

- Literals can be string, numeric, or Boolean, and do not need to be enclosed in quotation marks.

```
Equals(String,PropValue(Core.ParentDescr),UnitedKingdom)
```

Property name String literal



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Formulas are automatically checked for syntax before they are saved. The following conditions are verified:

- Whether function names and property names are correct
Note: You must include the namespace in a property name (for example, Core.Descr).
- Whether an equal number of opening and closing parentheses are present
- Whether the correct number of parameters are present. The parameter count validation verifies that the actual number of parameters is equal to or greater than the expected number of parameters. Thus, too many parameters do not generate an error, but too few parameters do.

Note: The syntax validation does not evaluate the formula; therefore, errors may occur if you enter invalid constants.

Any value that is not a valid function name followed by parentheses is considered a literal. It can be a string, integer, decimal, or Boolean literal. The following considerations apply for literals:

- In a string literal, spaces are treated like any other character. Therefore, you should not use extra spaces in formulas unless they are necessary to derive the appropriate results.
- Quotation marks ("") are not needed around literals, including string literals.
- Valid Boolean literals are 1, 0, T, F, True, and False.

Functions

- `Concat(Item1, Item2, ... ItemN)`

```
Concat(Country Code:, PropValue(Custom.Code))
```

- `NodeIsLeaf()`

```
NodeIsLeaf()
```

- `PropValue(PropertyName)`

```
PropValue(Core.Descr)
```

- `SubString(String, Index, Count)`

```
SubString(Colas, 1, 2)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Function names are not case-sensitive and should be immediately followed by parentheses, regardless of whether any parameters are required.

Function parameters must be of the expected type and number. Use commas to separate parameters. Parameter values can be nested functions, property names, or string literals. If parameters are of an incorrect type, an error is reported. In the case of too few parameters, a “List index out of bounds” error is reported. In the case of too many parameters, the additional parameters are ignored.

In certain functions where parameter values may need to contain special characters (for example, comma, space, and tab), use a special [keyword] notation. For example, use `FlipList(NodeList, [comma])` to perform the FlipList function on the comma-delimited list named NodeList.

The following functions can take [comma], [space], or [tab] as the Delimiter parameter: `ArrayCount`, `ArrayIndex`, `ArrayItem`, `FlipList`, `Intersection`, `ListContains`, `PadList`, `RangeListContains`, and `IsRangeListSubset`.

The ReplaceStr function, which requires parameters for the old and new pattern, can take [comma], [space], [tab], [crlf], [cr], [lf], [openparen], or [closeparen] in addition to normal text strings.

The following list describes commonly used functions:

- **Concat** concatenates two or more specified strings into one and returns the result.
Syntax:
`Concat (Item1, Item2, ... ItemN:String) :String`
- **NodeIsLeaf** returns TRUE if the current node is a leaf; otherwise, it returns FALSE.
Syntax:
`NodeIsLeaf () : Boolean`
- **PropValue** returns the value of the specified property for the current node as a string.
Syntax:
`PropValue (PropAbbrev: String) : String`
where **PropAbbrev** is the name of the property to use.
- **SubString** returns a portion of the specified string, starting at the specified index and containing the specified number of characters.
Syntax:
`SubString (String: String, Index: Integer, Count: Integer) : String`
where
 - **SubString** is the string value on which to perform the function.
 - **Index** is an integer representing the index position to start searching for the substring. Zero indicates the first character position in the string.
 - **Count** is a number representing the number of characters to search, beginning from the starting index.

Note: For a complete list of functions, see the *Oracle Data Relationship Management Administrator's Guide, Release 11.1.2.3*.

Logical Operators

- Equals(ParamType, Param1, Param2)

```
Equals(String, PropValue(Custom.NodeType), Account)  
Equals(Integer, PropValue(Core.Children), 1)
```

- Or(Expression1, Expression2, ... ExpressionN)

```
Or(NodeIsLeaf(), Equals(String, PropValue(Custom.NodeType), AccLeaf))
```

- And(Expression1, Expression2, ... ExpressionN)

```
And(NodeIsLeaf(), Equals(String, PropValue(Custom.NodeType), Account))
```

- If(Expression, TrueResult, FalseResult)

```
If(NodeIsLeaf(), Account Leaf, Account Limb)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Logical operators enable you to test for different conditions. The following list describes some of the most commonly used logical operators.

- **Equals** returns TRUE if two values of a specified data type match exactly.

Syntax:

```
Equals(ParamType:String, Param1:String, Param2:String) : Boolean
```

- **Or** returns TRUE if any of the specified Boolean expressions are TRUE.

Syntax:

```
Or(Expression1, Expression2, ... ExpressionN: Boolean) : Boolean
```

- **And** returns TRUE if both of two expressions evaluate to TRUE.

Syntax:

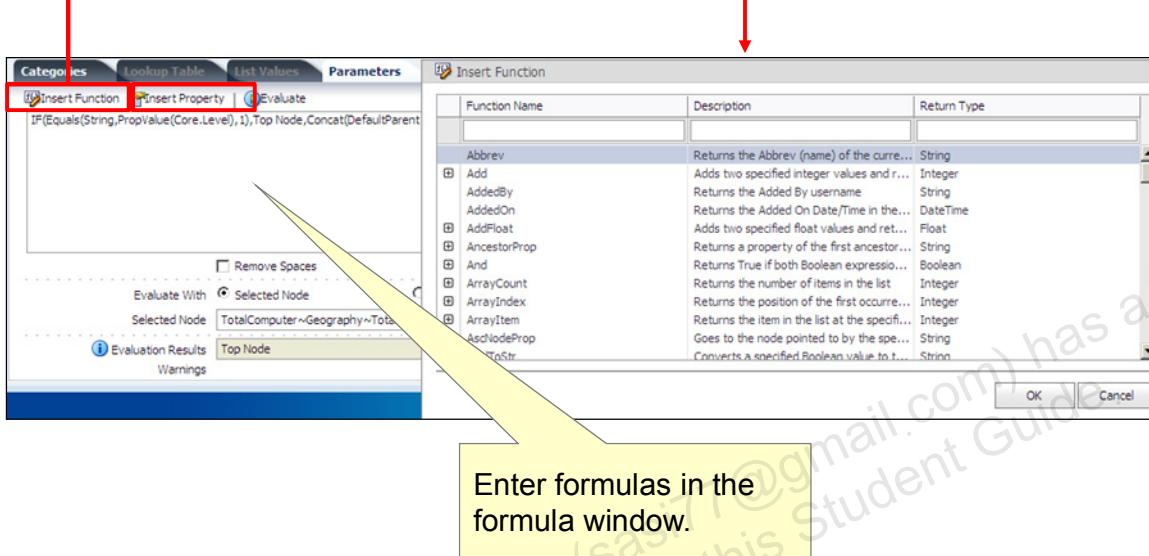
```
And(Expression1, Expression2, ... ExpressionN: Boolean) : Boolean
```

- **If** returns the first of two values if the test expression evaluates to TRUE; otherwise, it returns the second value.

Syntax:

```
If(Expression:Boolean, TrueResult:String, FalseResult:String) : String
```

Creating Derived Properties Using Formulas



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When creating a derived property based on a formula, you enter the formula on the Parameters tab. Formulas are evaluated from left to right, with functions and string literals being evaluated when they are encountered. By this method, nested functions are evaluated before additional parameters that appear to the right of the nested function. You can explicitly nest functions in a formula, or you can implicitly nest them by setting them to retrieve the value of another formula property.

To create derived properties based on the Formula deriver class:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Property Definition**.
3. Enter a name, a label, and an optional description for the property.
4. In the Property Level drop-down list, select **Global Node** or **Local Node**.
5. In the Data Type drop-down list, select a data type.
6. In the Property Type drop-down list, select **Derived**.
The Deriver Class drop-down list is automatically set to **Formula**.
7. **Optional:** In the Minimum Length, Maximum Length, and Column Width boxes, enter values.

8. Save the property definition.

Note: You must save the property definition before you can enter its formula.

9. Click the **Parameters** tab.

10. Perform one or more of the following actions to create a formula:

- Click **Insert Function** to insert a function into the formula. Select a function from the list, and click **OK**. Configure the parameters for the formula.
- Click **Insert Property** to insert a property into the formula. Select a property from the list, and click **OK**. This method is useful to ensure the accuracy of property names. Be sure to include the namespace followed by a period before the property name.

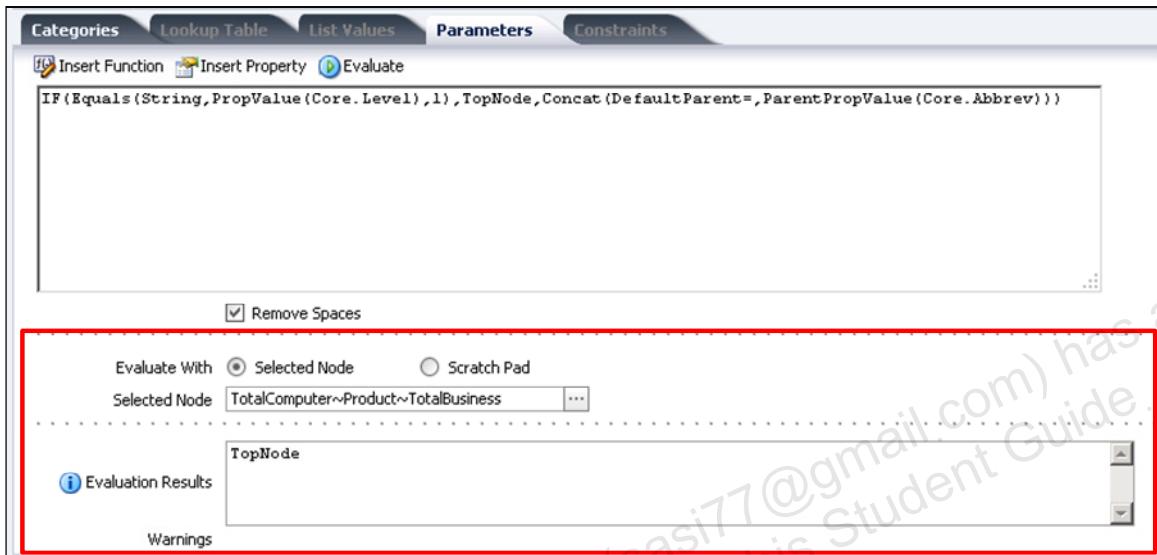
Note: Functions and properties are inserted where you place the cursor in the formula.

- Enter formula text.

11. **Optional:** Clear **Remove Spaces** if you want to include spaces in a literal value (for example, between the label and description in a concatenated string).

Testing Formulas

You can test formulas by using nodes or the scratch pad.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When testing formulas, use the following options to supply data:

1. Test by using a local node. The property values of the selected node are used for formula evaluation.
2. Test by using the scratch pad. The values you enter for properties and functions are used for formula evaluation.

When testing by node, you can evaluate the formula with existing data in the application rather than by entering test values. Use the scratch pad to evaluate the formula with values that you cannot retrieve from a node.

To test formulas by using nodes:

1. Next to Evaluate With, leave **Selected Node** selected.
2. In the Selected Node box, click the ellipsis button ([...]). The Select Node dialog box is displayed.
3. Select a version, hierarchy, and node, and click **OK**.
4. Click **Evaluate**.
5. In the Evaluation Results box, view the formula result or error.

To test formulas by using the scratch pad:

1. Next to Evaluate With, select **Scratch Pad**.
2. **Optional:** Perform the following actions:
 - a. **Optional:** Directly under the Property Name heading in the scratch pad, enter a property name, and press **Enter**. The scratch pad displays only the property name that you entered.
 - b. For a property or function referenced in the formula, click the Edit button (), enter a value in the Value box, and click the Update button ().
3. **Optional:** In the Copy From Node box, click the ellipsis button (). In the Select Node dialog box, select a version, hierarchy, and node, and click **OK**. Property values from the selected node are copied to the scratch pad. Modify the property values, if needed.
4. Click **Evaluate**.
5. In the Evaluation Results box, view the formula result or error.

Derived Properties Using Scripts

- Business logic using JavaScript
- Variables and loops
- Enhanced editing, formatting, and commenting
- Improved calculation performance for larger data sets

```

Property Type: Derived
Script
Inherited: 
Overridable: 
List: 
Hidden: 
Minimum Length: 
Column Width: 20

Categories | Lookup Table | List Values | Parameters | Constraints
Insert Example | Insert Property | Evaluate

if(node.Parent == null) //top node only
{
    var numDescendants = node.PropValue("Core.Descendants");
    var enumerator = node.GetDescendantEnumerator();
    var totals = {};
    //count the unique values across descendants
    while(enumerator.MoveNext()) {
        var value = enumerator.GetCurrent().PropValue("HFM.AccountType");
        if(value == null || value == "") continue; //don't count empty values
        if(typeof(totals[value]) == "undefined")
            totals[value] = 0;
        totals[value] = totals[value] + 1;
    }
}

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

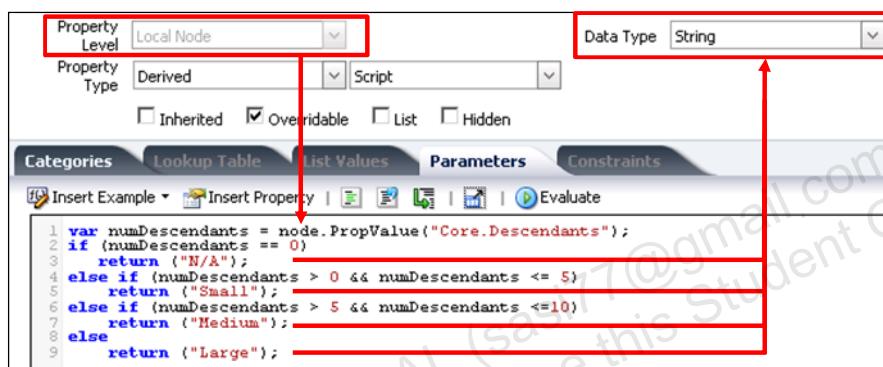
The Script deriver class enables you to create dynamic scripts for derived properties for versions, hierarchies, and nodes using JavaScript, a lightweight, interpreted, object-oriented programming language with first-class functions (JavaScript supports passing functions as arguments to other functions, returning them as the values from other functions, and assigning them to variables or storing them in data structures).

You create dynamic scripts in the script editor on the Parameters tab for the derived property definition.

Script Syntax

A script is a function that:

- Receives a parameter object defined by the property level of the derived property
- Returns a value appropriate to the data type of the derived property



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can think of a script as being wrapped in the JavaScript function syntax:

```
function myScript (parameter) { return (value); }
```

In this syntax, `parameter` is an object that depends on the Property Level of the derived property, as defined in the following table:

Property Level	Parameter	Object
Version	version	VersionObject
Hierarchy	hierarchy	HierarchyObject
Global Node	node	NodeObject
Local Node	node	LocalNodeObject

Note: Your script consists of the bold part. The function definition is provided for clarity.

The script must return a value that is appropriate to the data type of the derived property. If the returned value does not match, it will be coerced (for example, returning `null` for a Boolean property is treated as `false`). Not every property data type has a JavaScript representation. You are responsible for ensuring that you return a proper string representation of the property data type.

Property Enumeration Constants

- Use named constants to make code more readable and clear. For example,

```
if(nodeProp.PropOrigin == 2)
```

is equivalent to

```
if(nodeProp.PropOrigin == PropOrigin.Override)
```

- Named constants are available for DataType, PropType, PropLevel, and PropOrigin properties.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Certain properties are numbers that correspond to named constants. The following table lists named constants per property in ascending order, with values starting from zero.

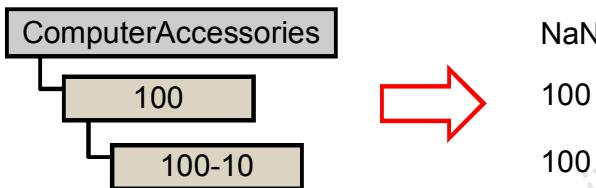
Property	Named Constants
DataType	Boolean, LeafNode, Date, Time, Float, Integer, Sort, Group, Node, LimbNode, String, Hier, Version, ListGroup, MultiNode, AscNode, AscNodes, AscGroup, Memo, FormatMemo, SortProp, Property, Query, StdQuery, GlobalNode, NodeProps, RangeList, DateTime, Hyperlink, HierarchyGroup
PropType	Invalid, System, Defined, Lookup, Derived, Stats, Validation, Verification, LimbAccessGroup, LeafAccessGroup, UserSpecific, RWDerived, SharedInfo
PropLevel	Node, Hier, Version
PropOrigin	Default, Inherited, Overridden, InheritedHier, InheritedVer, Derived, InheritedDomain, Unknown

JavaScript Data Type Objects and Functions

- Standard JavaScript data type objects and their methods are supported.
- JavaScript global functions are available.

```
var FMDesc = new String("English=");
return (FMDesc.concat(node.Descr));
```

```
var productFamilyCode = new String(parseInt(node.Abbrev));
```



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use standard JavaScript data type objects such as Number, String, Date, Array, Boolean, Math, or RegExp and their methods in your scripts. In the example shown in the slide, the concat () method is used to join the FMDesc String object with the property description. You must be familiar with JavaScript syntax and built-in objects, including what methods are available.

Note: JavaScript Document Object Model (DOM) objects are not supported in Data Relationship Management scripts.

You can also use JavaScript global functions in scripts. In the example shown in the slide, the parseInt () function parses the property definition name and returns an integer of the first number in the name. If the first character cannot be converted to a number, the function returns NaN.

Print and Format Functions

- Use the `print()` function to debug scripts.

The screenshot shows a script editor window with the following code:

```
1 var nodeLevel = node.Level;
2 print (nodeLevel);
3 if (nodeLevel == 1)
4   return ("TopNode");
5 else
6   return ("DefaultParent="+node.ParentNodeAbbrev);
```

Below the code, the "Evaluate With" dropdown is set to "Selected Node" and the "Selected Node" field contains "TotalComputer~Geography~North America". The "Evaluation Results" section shows the output: "DefaultParent=TotalGeography". At the bottom, there is a "Warnings" section with a yellow warning icon and the number "2". A red box highlights the code area, and another red box highlights the "Warnings" section.

- Use the `Format()` function to format output.

```
return Format("Time: {0:hh-MM-ss tt}", new Date());
```

Output: Time: 04-06-42 PM

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

`print()` Function

The `print()` function allows you to output debug information while creating scripts. The results are displayed in the Warnings section of the script editor. Although the `print` function produces only output in a testing context, the engine must still construct the arguments; therefore, comment out any `print` statements before saving a script for production use. In the example shown in the slide, the `print()` function outputs the node's level in the hierarchy.

`Format()` Function

The `Format()` function provides a much richer string formatting mechanism than standard JavaScript. The first parameter is a string that contains format specifiers surrounded with curly braces. Escape braces by doubling them, for example "`{}{}`" becomes "`{"}`" in the output. Format specifiers start at zero and increase incrementally. If you omit a specifier from a sequence, the equivalent parameter to the `Format` function is ignored. For example, "`{1}{}`" ignores the first value parameter to `Format` and uses the second.

Formatting specifiers are described in the chapter titled "Managing Dynamic Scripts" in the *Oracle Data Relationship Management Administrator's Guide Release 11.1.2.3*.

Data Relationship Management Objects

- SysObject
- PropDefObject
- VersionObject
- HierarchyObject
- NodeObject
- LocalNodeObject
- NodePropObject
- RangeListObject
- NodeEnumeratorObject
- ValidationObject



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

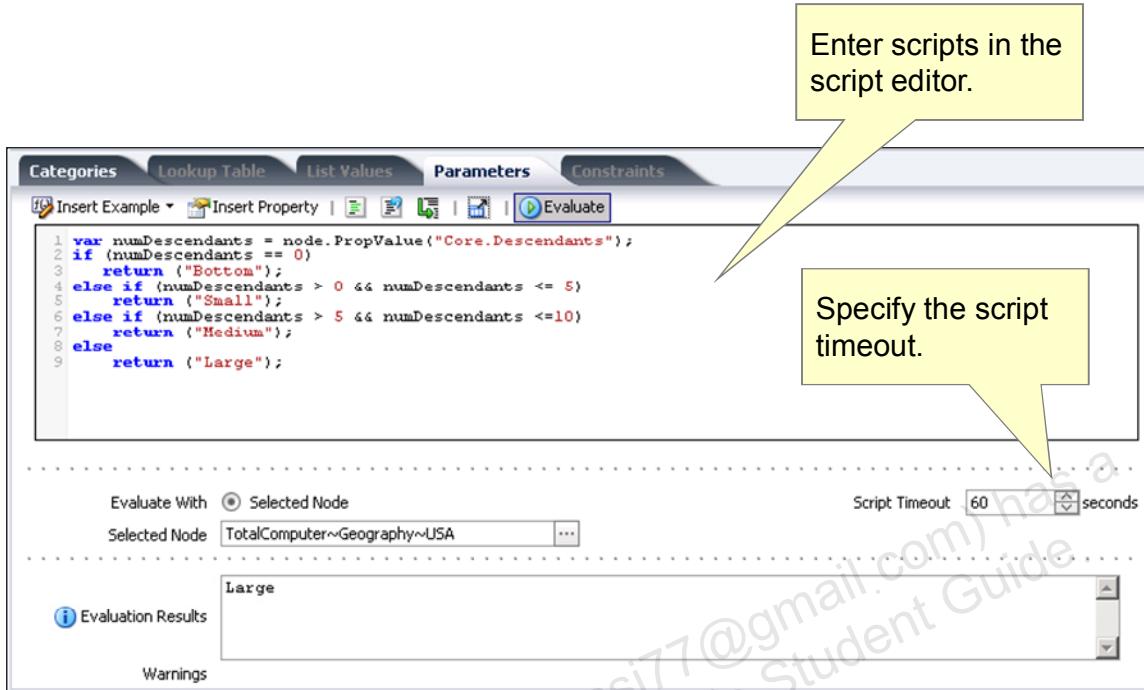
The following table lists general information about Data Relationship Management objects. For more details about their methods and properties, refer to the chapter titled “Managing Dynamic Scripts” in the *Oracle Data Relationship Management Administrator’s Guide Release 11.1.2.3*.

Object Name	Description
SysObject	Provides general functions and information about the Data Relationship Management application
PropDefObject	Provides property definition information
VersionObject	Provides version information
HierarchyObject	Provides hierarchy information
NodeObject	Provides global node information
LocalNodeObject	Provides local node information
NodePropObject	Provides node, version, or hierarchy property value information

Object Name	Description
RangeListObject	Represents a RangeList of values and can be used to inspect a RangeList property without having to manually parse strings. A new RangeListObject can also be constructed to return from a derived property of the appropriate data type.
NodeEnumeratorObject	Provides an efficient way to operate on a list of nodes. Instead of building the entire list all at once, the enumerator grabs only one node at a time as needed.
ValidationObject	Provides validation information

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Creating Scripts



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You create scripts in the script editor that is available on the Parameters tab for derived property definitions.

To create scripts:

1. Enter the script in the script editor.
2. **Optional:** Click **Insert Property** to insert a property into the script. Select a property from the list, and click **OK**. This method is useful to ensure the accuracy of property names.
Note: Functions and properties are inserted where you place the cursor in the script.
3. In the Script Timeout text box, specify the number of seconds until the script times out.
4. **Optional:** To evaluate the script with a selected node, click and select a node. The node's current property values are used in the script. Click **Evaluate**. The result is displayed in the Evaluation Results box.

Summary

In this lesson, you should have learned how to:

- Describe formulas and scripts for derived properties
- Explain formula syntax and script constructs
- Create derived properties by using formulas and scripts
- Test formulas and scripts

Quiz

For derived properties, you must save the property before you can enter the formula or script.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

When you create a formula for a derived property, which of the following can you do?

- a. Insert functions.
- b. Enter text.
- c. Define properties.
- d. Insert properties.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, d

Quiz

When you check formulas for syntax, which of the following conditions do you verify?

- a. Function names are correct.
- b. Property names are correct.
- c. An equal number of opening and closing parentheses are present.
- d. The correct number of parameters are present.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Quiz

You can test formulas and scripts for specific nodes.

- a. True
- b. False

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

10

Validating Data

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe validations, validation levels, and validation classes
- Create validations
- Assign validations to versions, hierarchies, and nodes
- Run validations
- Work with batch validation results
- Edit and delete validations



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Validations

- Enforce business rules on versions, hierarchies, nodes, and properties by using validations.
- Run validations in either real time or batch mode, or in both modes.
- Example validation: Ensure that all accounts starting with 100 are asset accounts.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Validations enable you to enforce business rules on versions, hierarchies, nodes, and properties. You can run validations in either real time or batch mode, or in both modes. Real-time validations are run at the time of modification and prevent changes from being saved if the action violates the rules being enforced. You explicitly run batch validations before or after edits are made to identify data conditions that are invalid and need to be addressed.

The following are typical requirements for which you can create validation rules:

- Adherence to standards for account numbers and account codes
 - **Example:** All accounts starting with 100 are asset accounts.
- Maximum depth of hierarchies
 - **Example:** Cost center hierarchies cannot exceed four levels.
- Adherence to limb and leaf requirements
 - **Example:** The Manager property must be defined for employee leaf nodes.

Validation Levels

- The validation level determines when and at what level validations run.
- The following levels are available:
 - Node
 - Hierarchy
 - Version
 - Global node
 - Merge
 - Move
 - Remove



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The validation level defines the scope of a business rule. For node real-time validations, the level can also include the type of action that must be performed in order for the validation to run. The following table lists available validation levels:

Validation Level	Meaning	Example
Node	Reviews node relationships and properties to ensure that criteria are met. This level is used most frequently.	Check that a description has a valid length.
Hierarchy	Reviews hierarchy properties to ensure that criteria are met.	Ensure that a hierarchy has no more than 10,000 nodes.
Version	Reviews version properties to ensure that criteria are met.	Ensure that a version contains no more than 100,000 nodes.
Global node	Validates every node in the version regardless of hierarchy, including orphans. Only properties defined as global are reviewed. It is assigned at the version level.	Ensure that all nodes have a name with more than eight characters.

Validation Level	Meaning	Example
Merge	Runs when an operation requiring a merge (for example, a delete or an inactivate) is performed. It is assigned at the version level.	Ensure that a leaf node is merged only with another leaf node.
Move	A validation triggered when an attempt is made to move a node. It is assigned at the hierarchy level.	Prevent moving cost center nodes within a hierarchy
Remove	Runs when an attempt is made to remove or delete a node from a hierarchy. You can use this level to prevent specified types of nodes from being deleted.	Prevent the deletion of cost center nodes from a hierarchy.

Note

- Hierarchy, Version, and Global Node validations can be run in batch mode only.
- Merge, Move, and Remove validations can be run in real-time mode only.

Validation Classes

- Enforce different types of business rules with validation classes.
- Create validations from a set of existing validation classes.
- Commonly used validation classes:
 - Date range check
 - Local property query validation
 - Global property query validation
 - Property equals value
 - Property length check
 - Required fields
 - Unique property value within a version
 - Formula
 - Script



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Validation classes enable you to enforce different types of business rules. You can use some validation classes generically, whereas you use other classes for specific purposes.

You can create validations from a set of existing validation classes. You can enforce many business rules on nodes with a validation class that uses a query for its logic. This method enables validations not only to leverage queries that are created for analysis purposes but also to manage data integrity. You can use other validation classes to enforce rules for versions and hierarchies or special cases for nodes.

Some validation classes are used for product testing purposes only and should not be used in a production environment.

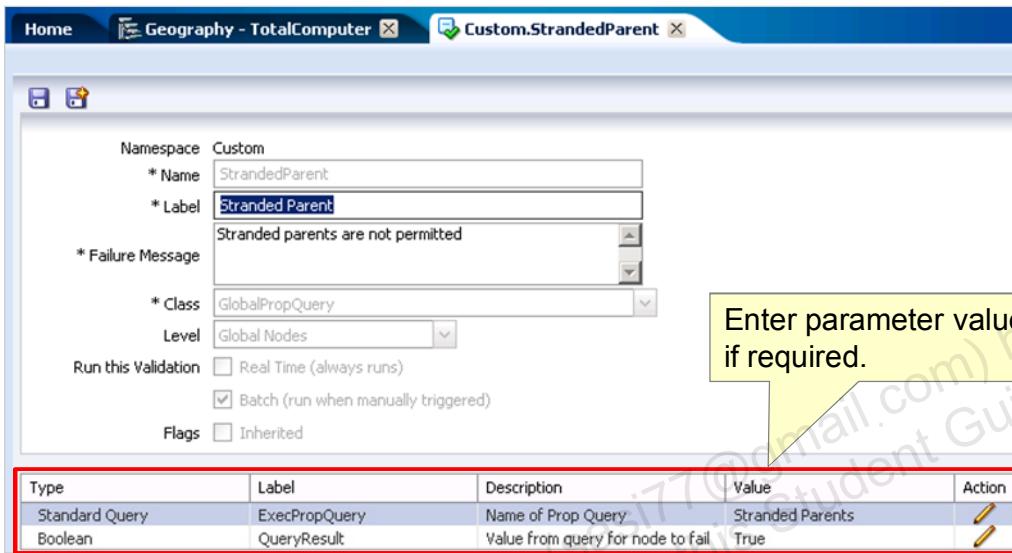
The following table describes the commonly used validation classes.

Validation Class	Level	Description	Parameters
Date range check	Node	Verifies that the From Date is earlier than or equal to the To Date	From Date Property, To Date Property
Local property query validation	Node	Verifies by using a predefined query and the expected result. You must set the object access level for the query to Standard.	Property query name, Failure value
Global property query validation	Global node	Verifies by using a predefined query and the expected result. You must set the object access level for the query to Standard.	Property query name, Failure value
Property equals value	Node	Fails for all nodes for which the specified property equals the specified value	Property, Value
Property length check	Node	Verifies that the specified property is at least minimum length and no more than maximum length	Property, Minimum Length, Maximum Length
Required fields	Node	For all nodes for which the specified property has a specified value, this class verifies that each property in the required list has a value: <ul style="list-style-type: none"> If the Reject Default Records flag is True, each property in the required list must have a value other than the default. If the Reject Default Records flag is False, default values are acceptable. 	Property, Value, Reject Default Records, Required Properties
Unique property value within a version	Global node	Verifies that the specified property has no duplicate values within a version. If the Include Defaults flag is False, nodes with the default value are not included.	Property, Include Defaults
Formula	Node	Verifies a node using business logic expressed in a formula. A formula result of False results in a validation failure.	Formula
Script	All levels	Verifies data using a dynamic script. A return value of True passes the validation. A return value of False results in failure of the validation.	Script

Note: For descriptions of all validation classes, see the *Oracle Data Relationship Management, Administrator's Guide Release 11.1.2.3*.

Creating Validations

Create validations in the Administer task group.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application administrators can create validations in the Administer task group. During creation, you select one of the following options to determine when the validation runs:

- **Real Time** runs when a change is saved. (This is available only for node-level validations.)
- **Batch** runs when explicitly requested by a user. (This is available for version, hierarchy, global node, and node-level validations.)
- **Inherited** runs for a node and its descendants. (This is available only for node-level batch validations.)

To create validations:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Validation**.
3. Enter a name and label for the validation.
4. Enter the message to display if the validation fails.
5. Select a validation class.

The validation levels that are populated depend on the selected class. For classes that can be run in real time at the node level, you can also select a level that includes a type of action to restrict when the validation runs.

6. Select a run option: **Real Time**, **Batch**, or **Inherited**.

Note: The selected validation class determines the availability of the run options.

7. If required, edit parameter values.

8. On the toolbar, click the Save button ().

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Validation Using Scripts

A validation script is a function that:

- Receives parameter objects defined by the validation level
- Returns an object that defines success or failure of the validation

The screenshot shows the Oracle Data Relationship Management interface. At the top, there is a code editor window containing a JavaScript validation script. The script checks if a node has a formula and returns an object with a success boolean and parameters array. A red box highlights the condition where 'Dynamic Calc' is found in the hierarchy. Below the code editor is a message box with a failure message: "Dynamic Calc requires a formula for node {0} in hierarchy {1}." A red arrow points from this message to the corresponding part of the code. Below the message box is an evaluation panel. It shows "Evaluate With Selected Node" set to "TotalComputer~Account~Margin Percent". The "Script Timeout" is set to 60. Under "Evaluation Results", there is a JSON object with "success: False" and "failureMessage: Dynamic Calc requires a formula for node Margin Percent in hierarchy Account.". A red arrow points from this evaluation result to the failure message in the message box.

```
1 var returnObject = new Object();
2 var MemberFormulaProp = node.PropValue("Essbase.Formula");
3 var DataStorageProp = node.PropValue("Essbase.DataStorage");
4
5 if (node.Leaf && MemberFormulaProp.length == 0 && DataStorageProp.indexOf("Dynamic Calc") >= 0) {
6   returnObject.success = Boolean(false);
7   returnObject.parameters = Array(node.Abbrev, node.HierAbbrev);
8 }
9 else
10   returnObject.success = Boolean(true);
11
12 return (returnObject);
```

* Failure Message Dynamic Calc requires a formula for node {0} in hierarchy {1}.

Evaluate With Selected Node Script Timeout 60
Selected Node TotalComputer~Account~Margin Percent

(i) Evaluation Results

```
{
  success: False
  failureMessage: Dynamic Calc requires a formula for node Margin Percent in hierarchy Account.
}
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Script validation class enables dynamic scripts to be used with validations. As with scripts for derived properties, you can think of a validation script as being wrapped in the following JavaScript function syntax:

```
function myScript(validation, parameter1, parameter2) {
    return (object); }
```

In this syntax, validation is a ValidationObject object, and parameter1 and parameter2 are objects that depend on the validation level as defined in the following table:

Validation Level	Parameters	Description
Version	version	VersionObject for the version being validated
Hierarchy	hierarchy	HierarchyObject for the hierarchy being validated
GlobalNode	node	NodeObject for the global node being validated
Node	node	LocalNodeObject for the node being validated

Validation Level	Parameter	Description
Remove	node	LocalNodeObject for the node being removed
Move	node	LocalNodeObject for the node being moved
	move	An object that contains information about the move: <ul style="list-style-type: none"> • OldParent: LocalNodeObject for the original parent • NewParent: LocalNodeObject for the destination parent • IsPost/IsPre: Indicates whether this script is running just before the move or just after the move has been completed. The script will usually be run twice: once before the move and once after the move. • Values: During the pre-move phase, simple key-value pairs can be stored in this object (for example, Values["key"] = "value"). During the post-move phase, these values are present, enabling you to store information about the pre-move state and compare it to the post-move state. All values are converted to String, Number, or Date objects. Complex objects are not currently supported.
Merge	node	LocalNodeObject for the node being deleted or inactivated.
	merge	An object that contains information about the merge: <ul style="list-style-type: none"> • Target: NodeObject for the target of the merge • IsInactivate: True if this is an deactivate operation • IsDelete: True if this is a delete operation

The validation script returns a JavaScript object that contains a property named `success`. If the script returns a Boolean value or a non-Boolean object (for example, Number or String), its value is converted to Boolean using standard JavaScript conversion rules and then assigned to the `success` property.

The script can return an optional JavaScript array of values in a property named `parameters`. The array values are substituted into the failure message of the validation by using string substitution.

Validation Enumeration Constants

- Use named constants to make code more readable and clear.
- Consider the following example:

```
if(validation.ValidationLevel == 0)
```

The preceding code is equivalent to this:

```
if(validation.ValidationLevel == ValidationLevel.Node)
```

- Named constants are available for ValidationLevel and ValidationType object properties.



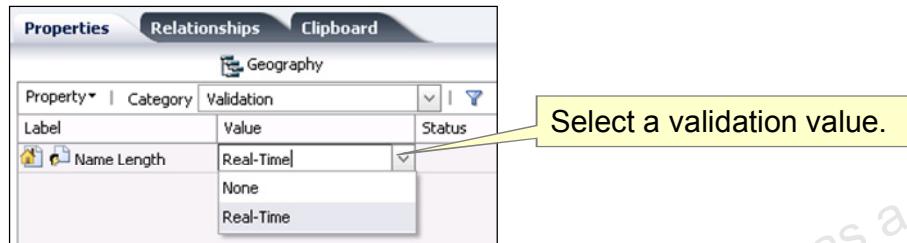
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the following validation enumeration constants in your scripts:

Property	Named Constants
ValidationLevel	Node, Hier, Version, GlobalNodes, Merge, Move, Remove
ValidationType	None, RealTime, Batch, Both

Assigning Validations

- Version and hierarchy owners as well as data managers can assign validations to versions, hierarchies, nodes, and node types.



- You can assign multiple validations at the same time.
- Assigned validations are inherited by all levels under the assigned level.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Version and hierarchy owners as well as data managers can assign validations to versions, hierarchies, nodes, and node types.

Note: Assigning validations to node types is covered in the lesson titled “Setting Up Node Types.”

You can assign multiple validations at the same time. Assigned validations are inherited by all levels under the assigned level, as in these examples:

- When assigned at the version level, validations are inherited by all hierarchies and nodes within the version.
- When assigned at the hierarchy level, validations are inherited by all nodes within the hierarchy.
- When assigned at the node level, validations are inherited by all levels under the node within the hierarchy if the Inherited option for the validation is set to True. You can unassign validations for selected descendants of the node, if needed.

To assign validations to versions:

- In the Browse task group, select a version.
- In the Versions drop-down list, select **Assign Validations**.
- On the Properties tab, select a validation and set its value to **None**, **Batch**, or **Real-Time**. Then click **Save**.

To assign validations to hierarchies:

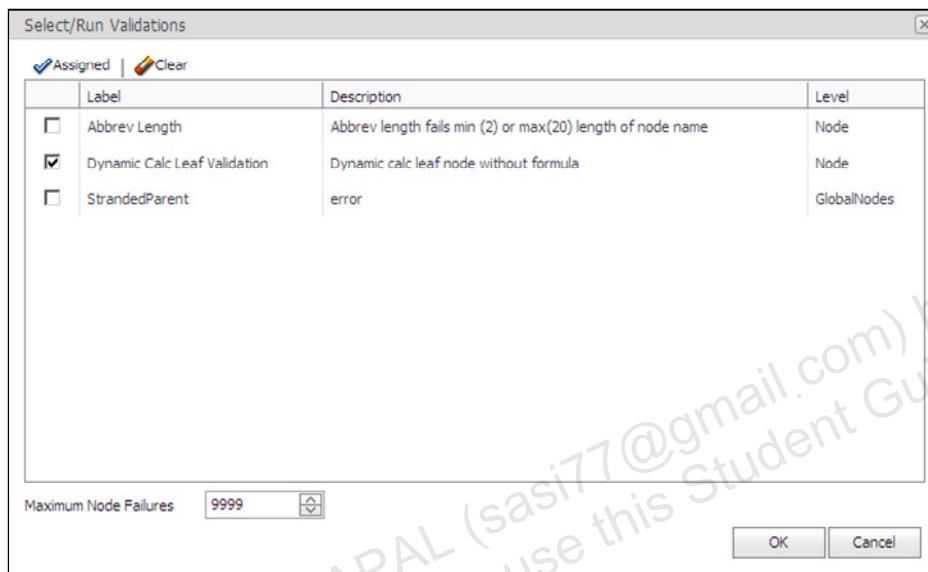
1. In the Browse task group, select a version.
2. On the Hierarchies tab, select a hierarchy.
3. In the Hierarchies drop-down list, select **Assign Validations**.
4. On the Properties tab, select a validation and set its value to **None**, **Batch**, or **Real-Time**. Then click **Save**.

To assign validations to nodes:

1. In the Browse task group, select a version.
2. On the Hierarchies tab, double-click a hierarchy to open it.
The hierarchy tab is displayed.
3. In the hierarchy tree, select a node.
4. In the Nodes drop-down list, select **Assign Validations**.
5. On the Properties tab, select a validation and set its value to **None**, **Batch**, or **Real-Time**. Then click **Save**.

Running Batch Validations

You can select specific batch validations to run, or you can run all assigned batch validations at one time.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can select specific batch validations to run, or you can run all assigned batch validations at one time.

To run selected batch validations:

1. Right-click a version, hierarchy, or node and select **Validate**. Then select **Selected**. The Select/Run Validations dialog box is displayed.
 2. Perform one of the following actions:
 - Select specific validations to run.
 - Click **Assigned** to automatically select validations that are currently assigned. You can select additional validations and deselect validations if needed.
 3. Click **OK**.
- The validations run. A message indicates that the validations were completed successfully, or the Validation Results tab displays failed validations.

To run assigned validations:

1. Right-click a version, hierarchy, or node.
2. Select **Validate** and then select **Assigned**.

The assigned validations run. A message indicates that the validations were completed successfully, or the Validation Results tab displays validation errors.

Working with Batch Validation Results

- Batch validations can return results for one or more validation levels, depending on where they are run.
- Results for version, hierarchy, and global node-level validations are displayed on the Validation Results tab in the Browse task group.
- Results for node-level validations are displayed in a list and marked tree.
- Hierarchies with validation errors are marked with a red exclamation point.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Batch validations can return results for one or more validation levels, depending on where they are run:

- Validations that are run for a version can return results for version, hierarchy, global node, and node-level validations.
- Validations that are run for a hierarchy can return results for hierarchy and node-level validations.
- Validations that are run for a node in a hierarchy return results only for node-level validations.

Results for version, hierarchy, and global node-level validations are displayed on the Validation Results tab in the Browse task group. Results for node-level validations are displayed in a list and marked tree.

To view and work with batch validation results:

1. Run a validation.
If the validation fails, a Validation Results tab is displayed in the Browse task group and shows version, hierarchy, and global node-level validation results.
2. Click the **Hierarchies** tab.
Hierarchies with validation errors are marked with a red exclamation point (!).

3. Open a hierarchy to view the node-level validation results for the hierarchy.
Node-level validation results are displayed in a list on another Validation Results tab and in a marked tree on a Tree tab.
4. On the Validation Results tab, perform one or more of the following actions:
 - Next to a node, click the plus sign (+) to view its validation errors.
 - Select the check box for a node, and view and edit its properties on the Properties tab.
 - In the Node drop-down list, select **Take/Copy** to add the node to the Clipboard.
 - To the right of a node, click the Go To Node button () to navigate to a node in the tree.
5. On the Tree tab, perform one or more of the following actions:
 - On the toolbar, click the Expand to Marked Nodes button () to expand the tree to display all marked nodes.
 - Click the “Move to Next Marked Node” button () to navigate from one marked node to the next.
 - Click the Clear Marked Nodes button () to clear all marked nodes in the tree.

To clear batch validations results:

1. Click the **Browse** tab.
2. Perform one of the following actions:
 - In the Versions drop-down list, select **Clear Validation Results**.
 - On the Hierarchies tab, select **Hierarchies** and then **Clear Validation Results**.

Editing and Deleting Validations

- You can edit and delete validations in the Administer task group.
- You cannot modify the class, level, or mode of operation parameters after you save a validation.
- When you delete a validation, all validation assignments to versions, hierarchies, and nodes are also deleted.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To edit validations:

1. On the Home page, select the **Administer** task group.
2. Under Metadata, expand **Validations**, and select a validation.
3. On the toolbar, click the Edit button ().
The validation tab is displayed.
4. Make changes to the validation, and click **Save**.

To delete validations:

1. On the Home page, select the **Administer** task group.
2. Under Metadata, expand **Validations**, and select a validation.
3. On the toolbar, click the Delete button ().
The Confirm Delete dialog box is displayed.
4. Select **Delete this Validation** to confirm the deletion.

Summary

In this lesson, you should have learned how to:

- Describe validations, validation levels, and validation classes
- Create validations
- Assign validations to versions, hierarchies, and nodes
- Run validations
- Work with batch validation results
- Edit and delete validations

Quiz

Which of the following is for running a validation for a selected node and its descendants?

- a. Batch
- b. Inherited
- c. None
- d. RealTime

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

Which of the following are available validation levels?

- a. Node
- b. Node Type
- c. Version
- d. Global node



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c, d

Quiz

To use a query in a validation, you must save it as a standard query.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Assigned validations are inherited by all levels under the assigned level.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

11

Setting Up Node Types

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain node types and their setup process
- Create, edit, and delete node types
- Create, edit, and delete glyphs and assign them to node types
- Determine the best methods for assigning node types to nodes
- Configure Hierarchy Node Type property values
- Troubleshoot node type assignments



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Node Types

- Classify nodes based on common characteristics
- Filter displayed and editable properties for nodes
- Filter validations to be run for nodes
- Can only be used for local nodes in hierarchies

Example:

	Node Type	Properties
Account dimension	Account	<ul style="list-style-type: none">• Account Type• Alias
Entity dimension	Entity	<ul style="list-style-type: none">• Country Code• Alias



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use node types to classify and manage nodes based on their characteristics. Node types enable you to filter the list of properties displayed differently for users and apply specific validations to a subset of nodes in one or more hierarchies.

In the slide example, the properties in the application include Alias, Account Type, and Country Code, and have the following requirements:

- Nodes used in the Account dimension in the target systems do not need the Country Code property.
- Nodes used in the Entity dimension in the target systems do not need the Account Type property.
- Nodes for both Account and Entity dimensions in the target systems need the Alias property.

To meet the requirements, you create the following node types:

- The Account node type includes the Account Type and Alias properties, but not the Country Code property. You assign the Account node type to hierarchies for the Account dimension.
- The Entity node type includes the Country Code and Alias properties, but not the Account Type property. You assign the Entity node type to hierarchies for the Entity dimension.

Often, more complex node types, which combine several groupings, are required. For example, the IsCalculated property for all account leaf nodes must be set to FALSE. To meet the requirement, you perform the following tasks:

- Create node types named Account-Limb and Account-Leaf.
- Create a validation to verify that the IsCalculated property is set to FALSE for all account leaf nodes.
- Assign the rule to the Account-Leaf node type.

Note: Node types and property categories are different. The node type assigned to a node determines which properties and validations are valid for the node. Property categories organize properties into groups for display and selection purposes.

Process for Setting Up Node Types

1. Create node types.
2. **Optional:** Create glyphs and assign them to node types.
3. Assign node types to nodes.
4. Configure Hierarchy Node Type property values.

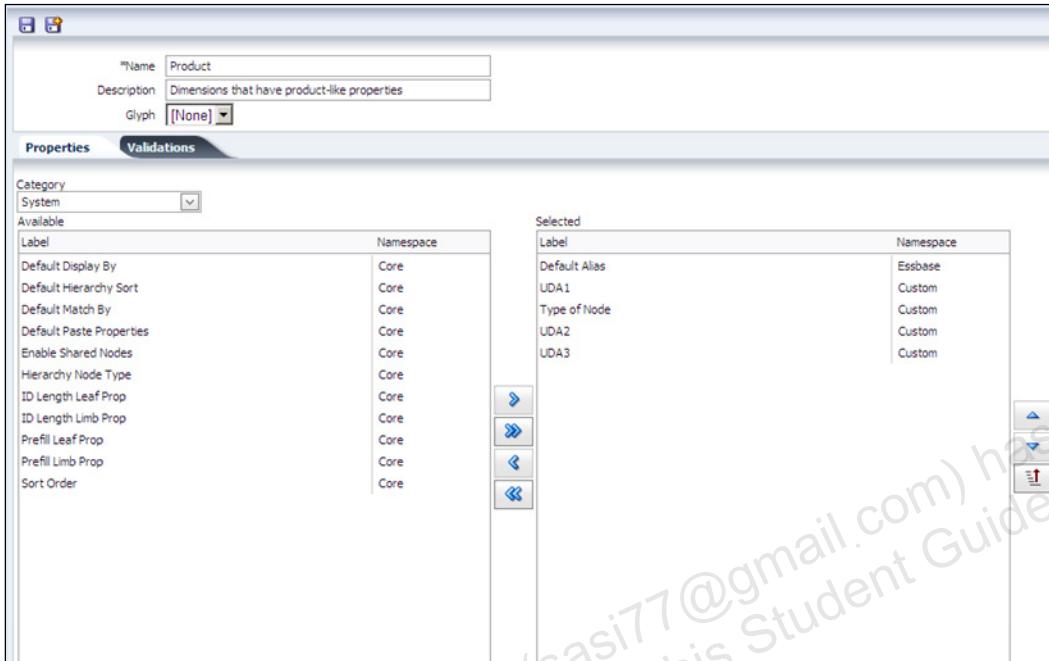
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The general process for setting up node types is as follows:

1. Create node types, assigning properties, and validations to each node type.
2. **Optional:** Create glyphs for node types and assign them to node types.
3. Assign node types to nodes, populating the properties manually or with a formula.
4. Configure Hierarchy Node Type property values for each hierarchy that will use node types.

Creating, Editing, and Deleting Node Types



The interface for creating a node type

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You create, edit, and delete node types in the Administer task group. When creating node types, you assign an optional glyph, properties, and validations to the node type.

A validation runs only for nodes that belong to a node type to which it is assigned. For example, you create a validation named AssetTest to verify that all nodes with an account type of asset have a prefix of 100. Because only account nodes have an account type, you assign the AssetTest validation to the Account node type, but not to the Entity node type.

It is important to plan a node type name carefully, because you cannot change it after you save the node type.

To create node types:

1. On the Home page, select the Administer task group.
2. In the New drop-down list, select Node Type.
3. Enter a name and description.
4. **Optional:** Select a glyph.
5. On the Properties tab, perform one or more of the following actions to assign properties to the node type:
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.

- Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
6. **Optional:** Click the Alphabetize button () to sort the properties in alphabetical order.
 7. **Optional:** On the Validations tab, perform one or more of the following actions to assign validations to the node type:
 - Select validations in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all validations in the Available list to the Selected list.
 - Select validations in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all validations in the Selected list to the Available list.
 8. **Optional:** Click the Alphabetize button () to sort the validations in alphabetical order.
 9. On the toolbar, click the Save button ().

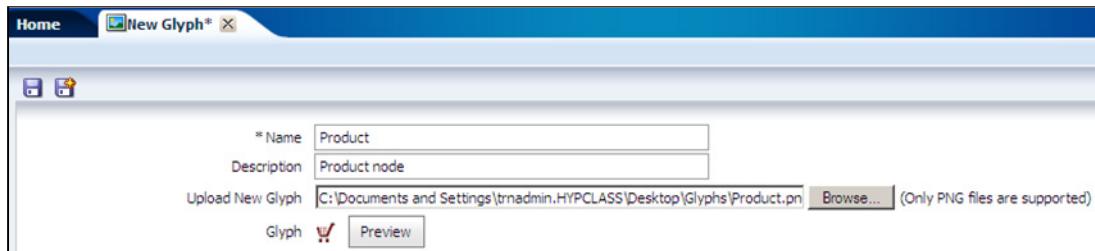
To edit node types:

1. On the Home page, select the Administer task group.
2. Under Metadata, expand Node Types, and select a node type.
3. On the toolbar, click the Edit button ().
The node type tab is displayed.
4. Perform one or more of the following actions:
 - Edit the description.
 - Select a different glyph.
 - Add or remove properties.
 - Add or remove validations.
5. On the toolbar, click the Save button ().

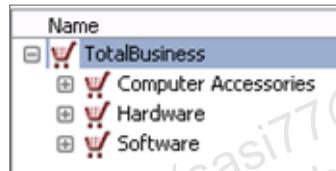
To delete node types:

1. On the Home page, select the Administer task group.
2. Under Metadata, expand Node Types, and select a node type.
3. On the toolbar, click the Delete button ().
The Confirm Delete dialog box is displayed.
4. Select **Delete this Node Type**.

Creating, Editing, and Deleting Glyphs



The interface for creating a glyph



Nodes that have the Product node type

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A **glyph** is an icon that you assign to a node type. The node type glyph is displayed for nodes in hierarchies to visually identify and distinguish nodes of different types. You create, edit, and delete glyphs in the Administer task group. When editing glyphs, you can change the description and select another PNG file, but you cannot change the name. You cannot undo glyph deletions.

To create glyphs:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Glyph**.
3. Enter a name and an optional description for the glyph.
4. Click **Browse**.
The “Choose File to Upload” dialog box is displayed.
5. Browse to a folder, select a PNG file, and click **Open**.
6. Optional: Click **Preview** to view the glyph.
7. On the toolbar, click the Save button ().

To edit glyphs:

1. In the Administer task group, expand **Glyphs**, and select a glyph.
2. On the toolbar, click the Edit button ().
The glyph tab is displayed.
3. Perform one or more of the following tasks:
 - Modify the description.
 - Click **Browse**, and select another glyph.
4. On the toolbar, click the Save button ().

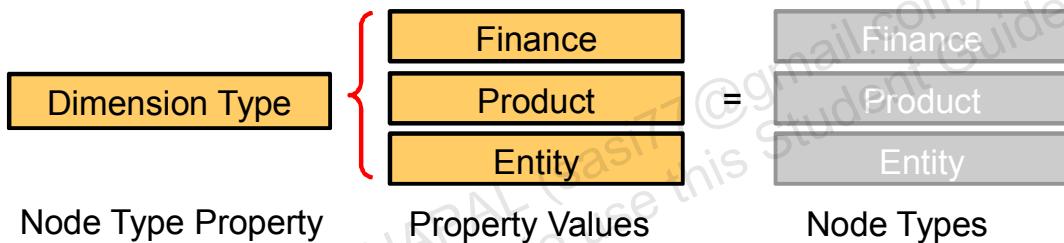
To delete glyphs:

1. In the Administer task group, expand **Glyphs**, and select a glyph.
2. On the toolbar, click the Delete button ().
The Confirm Delete dialog box is displayed.
3. Select **Delete this Glyph**.
The glyph is deleted, and you cannot undo the deletion.

Assigning Node Types to Nodes

You can use the following properties:

- Defined node-level property
- Inherited node-level property
- Lookup node-level property
- Derived node-level property
- Defined or lookup hierarchy-level property
- Defined or lookup version-level property



ORACLE

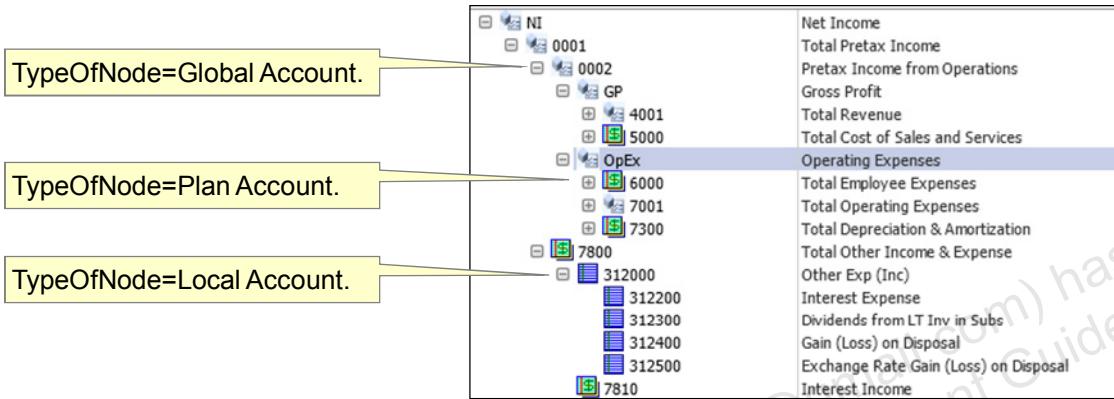
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You assign node types to nodes by using version-, hierarchy-, and node-level properties. The value of the property determines the node type assigned for the node. You can use existing properties or you can create properties specifically for this purpose.

The methods described on the following pages help you choose how to assign node types to nodes.

Defined Node-Level Property

The property value determines the node type for a node.



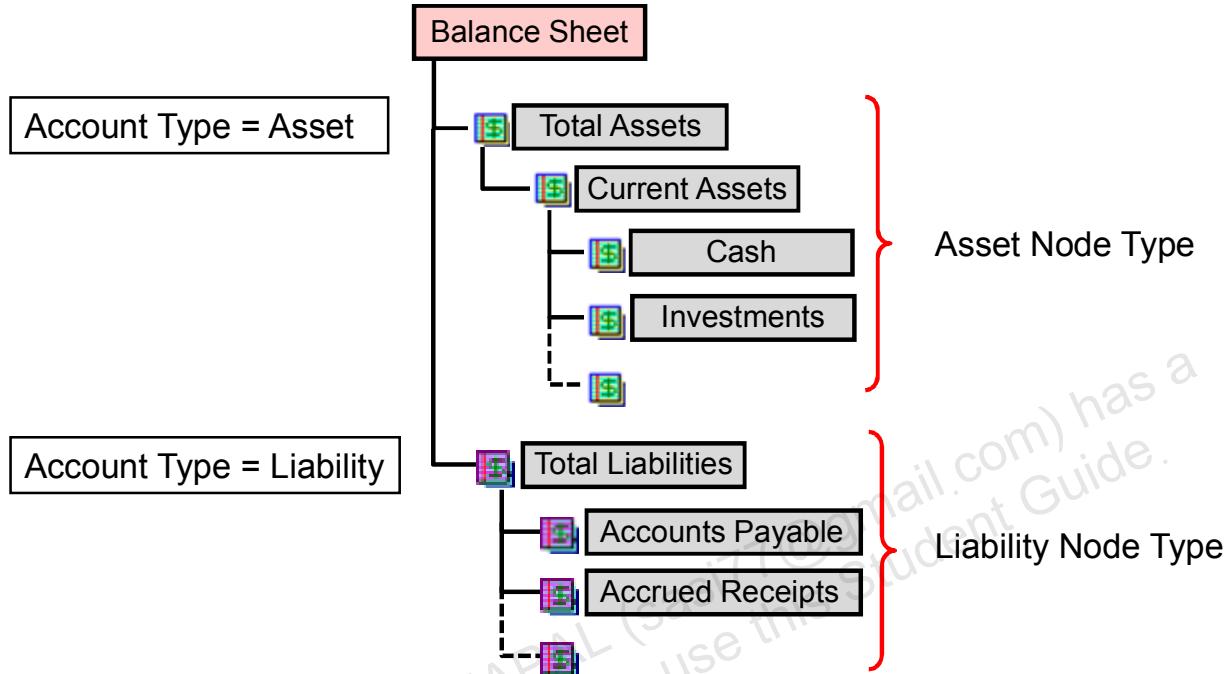
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Method 1: Use an existing defined node-level property

Usage: Use an existing defined property to determine the node type for a hierarchy node. This method is often used when the property is a required value for all nodes in the hierarchy (or has a default value), the hierarchy consists of nodes of many different types, and the node type cannot be determined based on the node position in the hierarchy.

Example: Use a node-level property named TypeOfNode with a list of values that matches the list of possible node types. End users select the appropriate node type from the list for each node. Although this is the simplest method, it requires a significant amount of user input and is not a best practice.

Inherited Node-Level Property



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

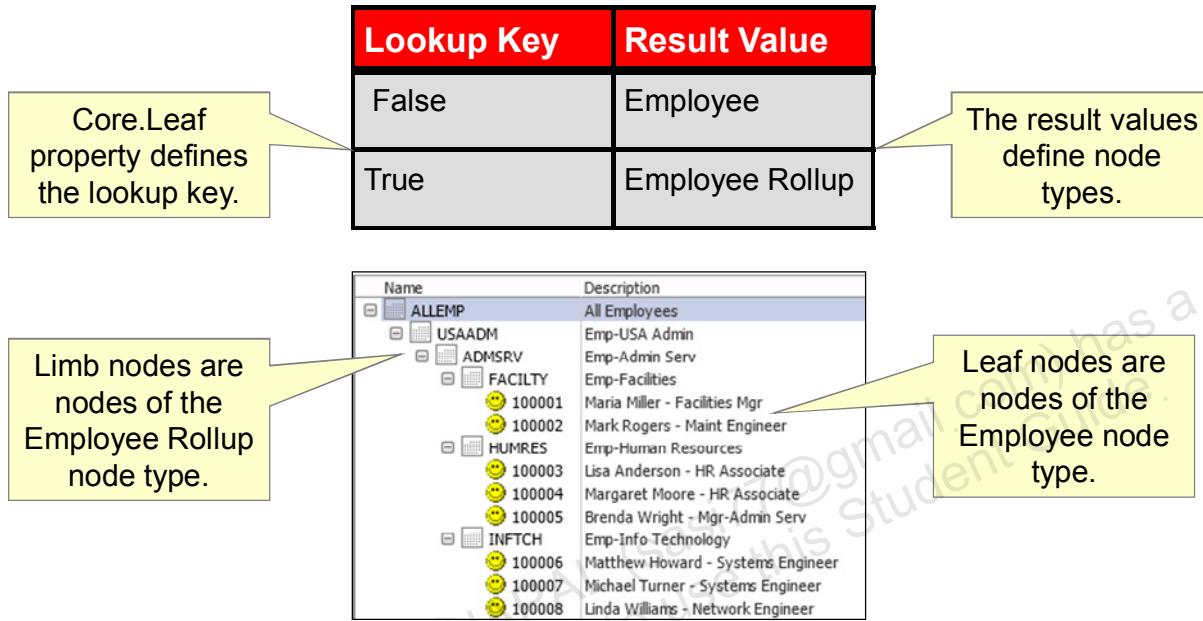
Method 2: Use an existing inherited node-level property

Usage: Use an existing inherited property to determine the node type for a node. This method is often used when a hierarchy has nodes of multiple types, and the node type can be determined based on the node position in the hierarchy.

Example: Use an Account Type property in a BalanceSheet hierarchy to control the node type. The Account Type property is inherited so that all nodes have the same account type (Asset, Liability, Equity) in certain sections of the hierarchy.

Lookup Node-Level Property

The lookup table determines the node type for a node.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Method 3: Create a lookup node-level property

Usage: Create a lookup property to determine the node type for a node. The lookup property usually points to another existing node/hierarchy/version-level property to determine the node type. This method is often used when a hierarchy has nodes of multiple types, and the existing property that drives the node type is more granular than the node types themselves. This method is also used when the existing property that drives the node type (such as a Boolean data type) must be converted to a different value (String data type) in order to match the node type name.

Example 1: Create a lookup property that references a core Boolean property named Leaf to determine node types for nodes in a hierarchy. Two node types exist (Employee and Employee Rollup). The lookup property maps True values to the Employee node type, and maps False values to the Employee Rollup node type.

Example 2: A Unit Of Measure property has the following values: US dollars, European euro, kilograms, square meters, cubic feet, or number of employees. Create a lookup property that references the Unit of Measure property to determine the node type for nodes in a Measures hierarchy. Two node types exist (Monetary Account and Statistical Account). The lookup property maps US dollars and European euro to the Monetary Account node type, and maps the other values to the Statistical Account node type.

Derived Node-Level Property

The script determines the node type for a node.

```
if (node.Hier=="Geography" && node.Level<5)
    return ("Nonregional");
else
    return (node.Hier.PropValue("Custom.DimensionType"));
```

Nodes with a level less than 5 are nodes of the Nonregional node type.

Name	Description
■ TotalGeography	
■ Africa	Continent
■ Asia	Continent
■ Europe	Continent
■ North America	Continent
■ USA	Country in North America
■ East	East US Region
■ Connecticut	State in USA
■ 1100	Administration - Holding East USA
■ 3100	Marketing - East USA
■ 4100	Development - East USA
■ 6100	Sales - East USA
■ Massachusetts	State in USA
■ Atlanta	Atlanta Office
■ New York	New York Office
■ West	West US Region
■ California	State in USA

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Method 4: Create a derived node-level property

Usage: Create a derived property to determine the node type for a node. This method is typically used when a hierarchy has nodes of multiple types, and conditional logic is required to determine the node type. This method is also used when the node type is based on a combination of multiple property values.

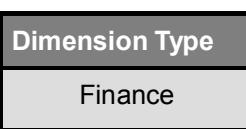
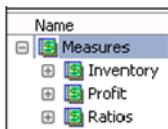
Example 1: Create a derived node-level property that uses the Core.Level property to determine a node type for nodes. For example, level 1 to 4 nodes are mapped to a Nonregional node type and the others are mapped to a node type defined by the hierarchy-level DimensionType property.

Example 2: Create a derived node-level property that concatenates a Dimension Type property and the Core.Leaf property to determine the node type. For example, an Account-Limb value is mapped to a Report Line node type, and an Account-Leaf value is mapped to a GL Account node type.

Hierarchy-Level and Version-Level Properties

Use hierarchy-level or version-level properties when the same node type is assigned to all nodes in the hierarchy or version.

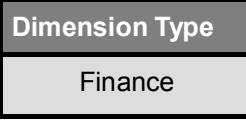
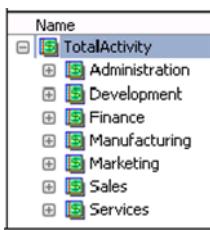
Account



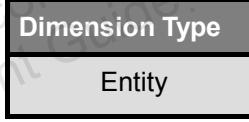
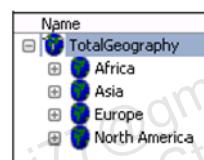
Product



Activity



Geography



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Method 5: Create a defined or lookup hierarchy-level property

Usage: Create a hierarchy-level property to determine the node type for a hierarchy node. This method is typically used when all nodes within a hierarchy have the same node type.

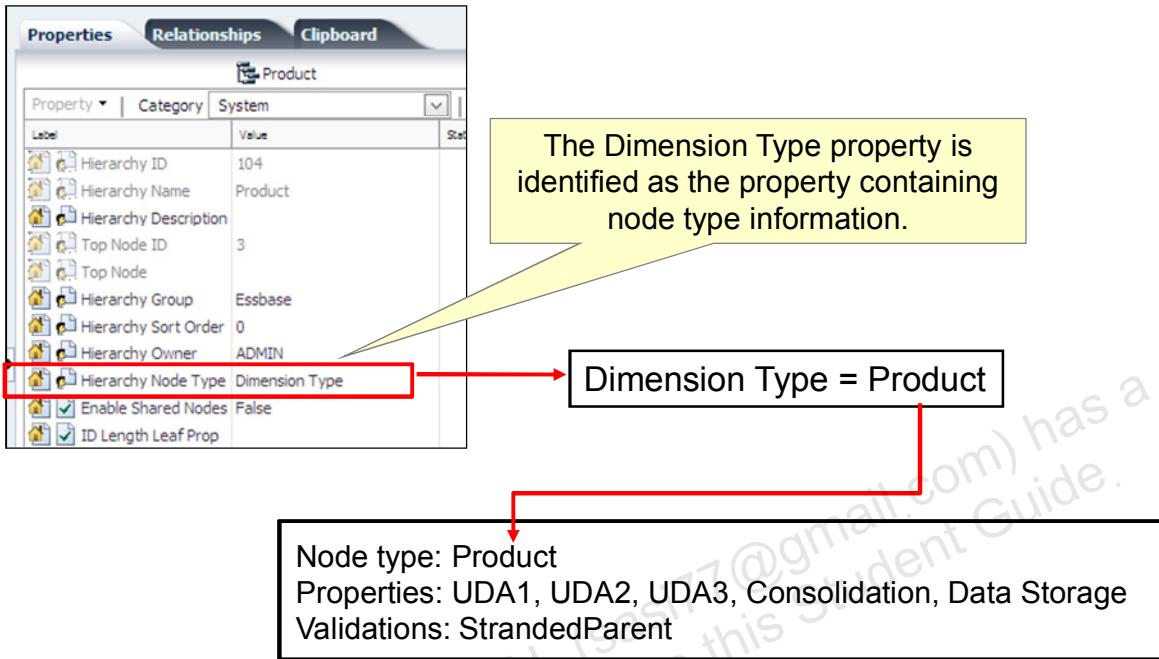
Example: You manage the Account, Entity, and Product dimension types in a version; therefore, you create a Dimension Type property at the hierarchy level with Account, Entity, and Product as possible values. The property value maps to a node type. Because Dimension Type is a hierarchy-level property, all nodes within each hierarchy are assigned the same node type.

Method 6: Create a defined or lookup version-level property

Usage: Create a version-level property to determine the node type for a node. This method is typically used when all nodes within a version have the same node type.

Example: You manage hierarchies of different dimension types within different versions; therefore, you create a Dimension Type property at the version level to control the node type for all nodes in each version.

Configuring Hierarchy Node Type Property Values



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Hierarchy Node Type property is a system-defined property at the hierarchy level. It points to the property that indicates the node type for nodes in the hierarchy. The Hierarchy Node Type property for each hierarchy can point to the same or different node type properties.

In the slide example, the Hierarchy Node Type property for the Product hierarchy points to the Dimension Type property. The Dimension Type property is a hierarchy-level property and is set to the appropriate node type, Product. Nodes that are assigned the Product node type display the UDA1, UDA2, UDA3, Consolidation, and Data Storage properties, and are assigned the StrandedParent validation.

To configure Hierarchy Node Type property values:

1. On the Hierarchies tab in the Browse task group, select a hierarchy.
2. On the Properties tab, select the **Hierarchy Node Type** property, and then select the property that identifies the node type, and click **Save**.

Troubleshooting Node Type Assignments

Improper assignment of node types can result in the following problems:

- Too few properties are displayed for nodes.
- Too many properties are displayed for nodes.
- Validations do not run for nodes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Improper assignment of node types can result in the following problems:

Too Few Properties Are Displayed for Nodes

- **Problem:** A property category displays too few properties than were assigned to it.
- **Solution:** Verify that the Properties tab for the node type includes the correct properties.

Too Many Properties Are Displayed for Nodes

- **Problem:** A property category displays more properties than were assigned to it.
- **Solution:** If a node's node type property is blank, its properties are not filtered; therefore, make sure that the node type property is populated. If the node type is assigned from a derived property, verify that the property formula returns valid values for all nodes.

Validations Do Not Run for Nodes

- **Problem:** A validation fails to run for a node that is assigned a node type.
- **Solution:** Verify that the node is assigned the correct node type and that the correct validations are included on the Validations tab for the node type.

Summary

In this lesson, you should have learned how to:

- Explain node types and their setup process
- Create, edit, and delete node types
- Create, edit, and delete glyphs and assign them to node types
- Determine the best methods for assigning node types to nodes
- Configure Hierarchy Node Type property values
- Troubleshoot node type assignments

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

Which of the following elements are shared by nodes within a specified node type?

- a. Names
- b. Properties
- c. Validations
- d. Glyphs



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b, c, d

Quiz

The difference between node types and property categories is that node types assigned to nodes determine which properties and validations are valid for nodes, whereas property categories organize properties into groups.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Which of the following methods can you use to define properties for assigning node types to nodes?

- a. Use a defined node-level property.
- b. Use an inherited node-level property.
- c. Create a lookup node-level property.
- d. Create a derived node-level property.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Quiz

If all nodes in a hierarchy share the same node type, you should use an existing inherited node-level property to assign node types to nodes.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

12

Exporting Data

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

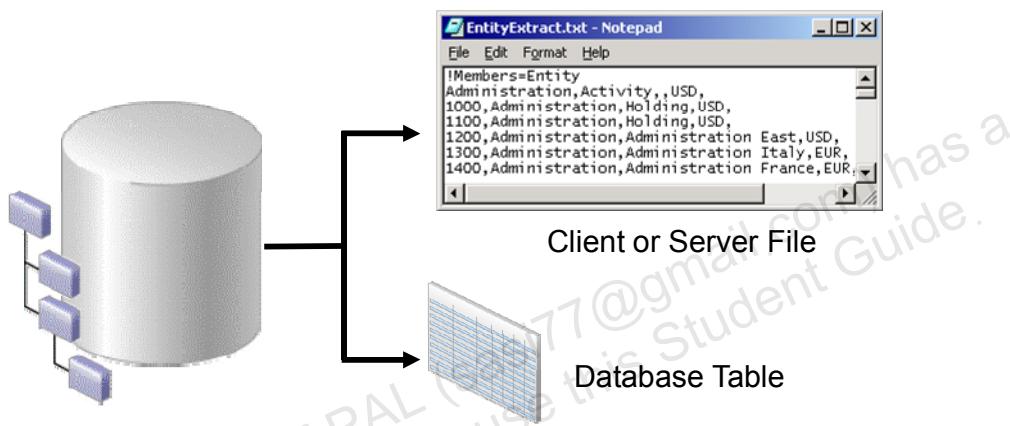
Objectives

After completing this lesson, you should be able to:

- Describe the exporting of data
- Explain export types
- Open, copy, and delete exports
- Create exports
- Run exports
- Create and run export books

Exporting Data

- Output data for transfer to target systems including ERP, BI, EPM applications, data warehouses, and data stores.
- Create reports for review and audit purposes.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Exports enable you to output information from Data Relationship Management for use by other systems and users. They are primarily used to transfer data to target systems, but you can also use them to create external reports for users for review and audit purposes. Typical target systems for exports include ERP, BI, and EPM applications, data warehouses, and data stores for integration processes that transfer data between them. Multiple exports can be grouped and run together using books.

Export Types

- Compare
- Difference
- EPM Architect
- Generation
- Hierarchy
- Hierarchy XML
- Merge Log
- Property List
- Property Lookup
- Transaction Log
- Version

Hierarchy exports are often the most commonly used export type.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

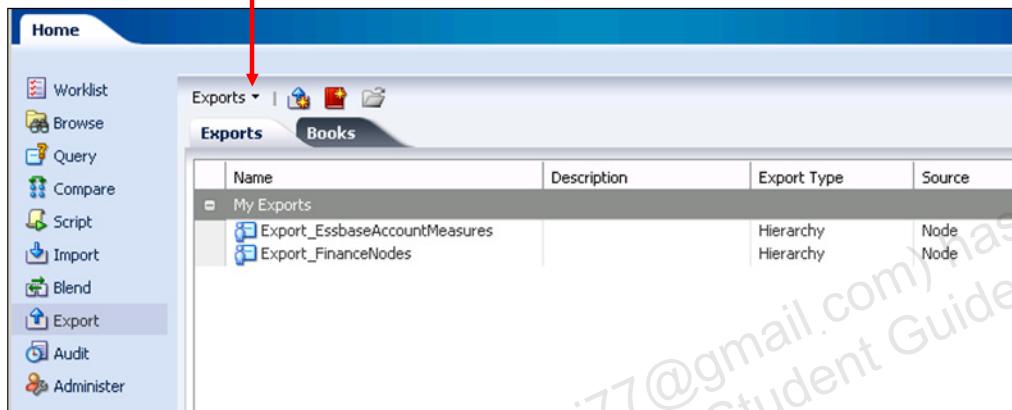
Many export types are available in Data Relationship Management, and you can customize them to suit your needs. The following table describes the export types:

Export Type	Description
Compare	Compares two hierarchies and outputs a record for each node that met the comparison criteria. Use this type to provide incremental changes to target systems.
Difference	Compares two hierarchies and outputs a record for each difference found. Use this type to report incremental changes to users.
EPM Architect	Outputs hierarchies, nodes, and properties from Data Relationship Management, which can be shared with and deployed to multiple EPM applications, such as Oracle Hyperion Financial Management and Planning.
Generation	Creates a generation-based or level-based format for nodes in hierarchies. It outputs a record for each node, its ancestor nodes, and their properties. Use this type to create a format where hierarchy relationships are represented as columns.

Export Type	Description
Hierarchy	Creates a tabular result set for nodes in one or more hierarchies. The result set can represent a flat list of nodes and their properties or a parent-child format of the hierarchy structure. Hierarchy exports are used as the basis for the majority of exports from Data Relationship Management to external systems.
Hierarchy XML	Exports version data to an external file in a generic XML format. This type provides easier access to data for downstream applications that leverage XML as a data source.
Merge Log	Outputs a record for each item in the merge log that meets the filter criteria. This export is applicable only if the Merge feature is used.
Property List	Outputs a record for each item in the list of values for a property definition. This type is for special-use cases (for example, you can output a list of currencies that you can then use as default currencies for an entity).
Property Lookup	Outputs a record for each key-value pair in the lookup table for a property definition. This type is for special-use cases (for example, you can output a table that maps transaction codes from a source system to descriptive attributes in a target application).
Transaction Log	Outputs a record for each item in the transaction log that meets the filter criteria. Use this type for auditing.
Version	Outputs a record for each node and its properties for an entire version. Hierarchy relationships and local properties are not available for this type.

Opening, Copying, and Deleting Exports

Open, Copy, and Delete options



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You open, copy, and delete exports and books in the Export task group, which consists of two tabs: Exports and Books.

To open exports:

1. On the Home page, select the **Export** task group.
2. On the Exports tab, select an export.
3. In the Exports drop-down list, select **Open**.

The export is opened. From this point, you can modify or run the export.

To copy exports:

1. In the Export task group, on the Exports tab, select an export.
2. In the Exports drop-down list, select **Copy**.
The Copy Export dialog box is displayed.
3. Enter a name and an optional description for the export.
4. In the Object Access Group drop-down list, select **User, Standard, System**, or a custom object access group (depending on the role of the user).
5. Click **OK**.

A copy of the export is created and listed on the Exports tab.

To delete exports:

1. On the Exports tab, select an export.
2. In the Exports drop-down list, select **Delete**.
The Confirm Delete dialog box is displayed.
3. Select **Delete the Export**.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Creating Exports

- You create exports in the export wizard, which consists of the following tabs:
 - Source
 - Style
 - Filters
 - Columns
 - Target
- Some tabs are not applicable to all export types.
- Each export type has its own set of selections that are required or that provide optional features.
- You can navigate between tabs when configuring exports.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You create exports in the export wizard, which consists of the following tabs: Source, Style, Filters, Columns, and Target. Some tabs are not applicable to all export types. Each export type has its own set of selections that are required or that provide optional features. Although the tabs are ordered to gather input in a sequence, you can navigate between tabs.

The following is a list of available tabs and selections for the export types.

Hierarchy Export

- **Source:** Specify the version, hierarchies, and top nodes to be exported.
- **Style:** Select types of nodes, batch validations, and output options.
- **Filter:** Select a query, validation, or external file for filtering export results.
- **Columns:** Select properties to be output and export column options.
- **Target:** Specify the target device and formatting and mapping options for the selected device.

Hierarchy XML Export

- **Source:** Specify the version, hierarchies, and top nodes to be exported.
- **Style:** Select types of node, batch validations, and output options.
- **Filter:** Select a query, validation, or external file for filtering export results.
- **Columns:** Select properties to be output.
- **Target:** Specify the target device.

Generation Export

- **Source:** Specify the version, hierarchies, and top nodes to be exported.
- **Style:** Select types of nodes, batch validations, and output options.
- **Filter:** Select queries, a validation, or external file for filtering export results.
- **Columns:** Select properties to be output and export column options.
- **Target:** Specify the target device and formatting options for the selected device.

Version Export

- **Source:** Specify the version to be exported.
- **Style:** Select types of nodes and batch validations.
- **Filter:** Select a query or validation for filtering export results.
- **Columns:** Select properties to be output.
- **Target:** Specify the target device, formatting, and mapping options for the selected device.

Compare Export

- **Source:** Specify versions, hierarchies, and top nodes to be compared.
- **Style:** Select types of nodes, compare options, batch validations, and types of differences to be exported.
- **Filter:** Select queries or a validation for filtering export results.
- **Columns:** Select properties to be output and export column options.
- **Target:** Specify the target device and formatting and mapping options for the selected device.

Difference Export

- **Source:** Specify versions, hierarchies, and top nodes to be compared.
- **Style:** Select types of nodes, compare options, and types of differences to be exported.
- **Filter:** Select a query or a validation for filtering export results.
- **Columns:** Select properties to be output.
- **Target:** Specify the target device and formatting options for the selected device.

Property List Export

- **Source:** Specify the property to be exported.
- **Target:** Specify the target device and formatting options for the selected device.

Property Lookup Export

- **Source:** Specify the property to be exported.
- **Style:** Select whether to output only differences.
- **Columns:** Select lookup table columns to be output.
- **Target:** Specify the target device and formatting options for the selected device.

Transaction Log Export

- **Source:** Specify the version, hierarchies, and nodes to be exported.
- **Filter:** Select query criteria for filtering export results.
- **Columns:** Select transaction log columns to be output.
- **Target:** Specify the target device and formatting and mapping options for the selected device.

Merge Log Export

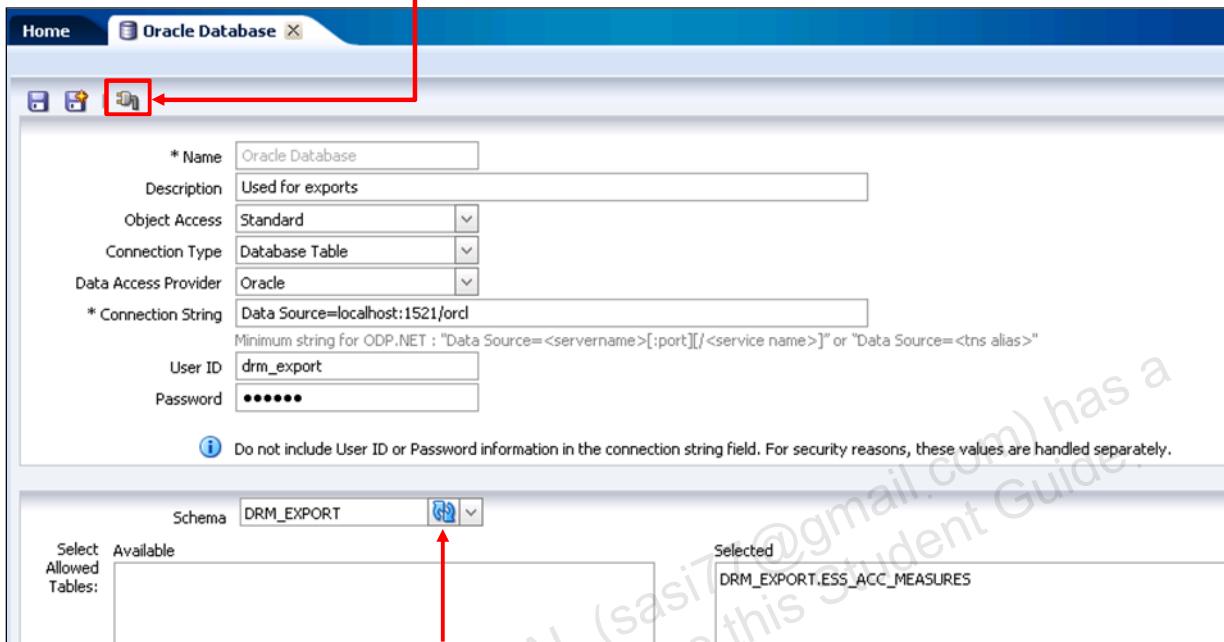
- **Source:** Specify the version to be exported.
- **Filter:** Select query criteria for filtering export results.
- **Columns:** Select merge log columns to be output.
- **Target:** Specify the target device and formatting and mapping options for the selected device.

EPM Architect Export

- **Source:** Specify the version to be exported, target application type, dimension property that determines the dimensions and hierarchies to be exported, and dimension type property that determines the dimension type for each dimension to be exported.
- **Style:** Select batch validations and output options.
- **Filter:** Select query criteria for filtering export results.
- **Columns:** Select the Parent Key Property and Member Key Property.

Creating External Connections

Test Connection button



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You must create external connections in the Administer task group before you can export data network file systems, FTP directories, and relational databases. For database connections, you can export to Oracle or SQL Server or another database type using Object Linking and Embedding Database (OLEDB).

Note: You can also use external connections for importing data from network file systems and relational databases.

To create external connections:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **External Connection**.
The New External Connection tab is displayed.
3. Enter a name and an optional description for the connection.
4. In the Object Access drop-down list, select **Standard**, **System**, or a custom object access group.
5. In the Connection Type drop-down list, select **Server File**, **FTP**, or **Database Table**.
6. If you selected Server File in step 5, in the UNC Path box, enter a path to the server.

Note: The Windows user account used by the Data Relationship Management application server is automatically used for server file connections.

7. If you selected FTP in step 5, perform the following actions:
 - a. In the Host Server box, enter a server name.
 - b. In the User ID and Password boxes, enter user credentials.
8. If you selected Database Table in step 5, perform the following actions:
 - a. In the Data Access Provider, select **Oracle**, **SqlServer**, or **OleDb**.
 - b. In the Connection String box, enter a connection string (for example, `Data Source=localhost:1521/orcl`).

Note: Oracle and SQL Server connections support bulk insert and delete operations, whereas OLEDB connections do not.

 - c. In the User ID and Password boxes, enter user credentials (for example, `drm_export` and `oracle`).
9. On the toolbar, click the Test Connection button ().
A Confirmation dialog box states that the test succeeded. If the connection does not succeed, review your server file, FTP, or database settings, and try the connection test again.
10. Click **OK**.
11. If you selected Database Table in step 5, perform the following actions:
 - a. To the right of the Schema drop-down list, click the Refresh List button () to populate the list of schemas and available tables.
 - b. Optional. In the Schema drop-down list, select a schema to filter the list of available tables that belong to the selected schema.
 - c. Perform one or more actions to assign access to database tables:
 - Select tables in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all tables in the Available list to the Selected list.
 - Select tables in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all tables in the Selected list to the Available list.
12. On the toolbar, click the Save button ().

Starting New Exports and Selecting Export Types



ORACLE®

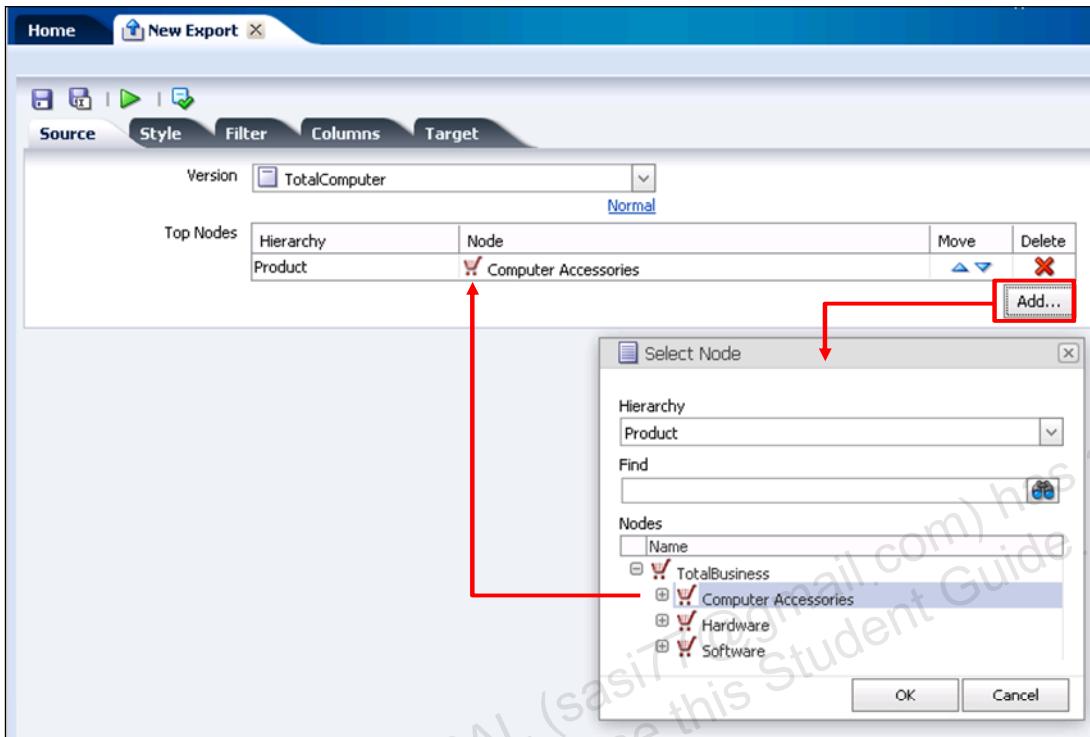
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You start new exports in the Export task group. When starting an export, you select an export type in the Choose Export type dialog box. This dialog box organizes export types into the following categories: Hierarchy Exports, Comparison Exports, Integration Exports, Version Export, Property Exports, and Log Exports.

To start new exports and select export types:

1. On the Home tab, select the Export task group.
The Exports tab of the Export task group is displayed.
2. On the toolbar, click the New Export button ().
The Choose Export Type dialog box is displayed.
3. Select an export type.
The export wizard is displayed in a new tab.

Configuring a Hierarchy Export Source



ORACLE

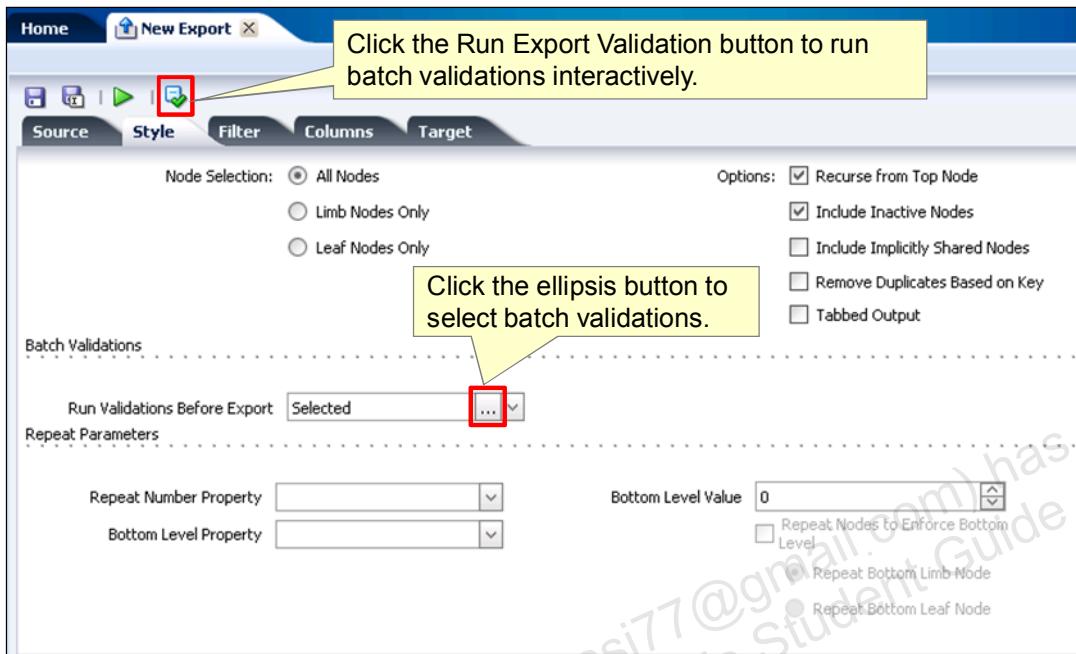
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Source tab in the export wizard, you specify the source of data that you want to export. You select a version, hierarchies, and top nodes. You can export from a normal, baseline, or As-Of version. Data for the selected hierarchy top nodes is exported. You can select as many top nodes from each export as required. The order of the top nodes in the list determines the order of the rows in the export results.

To configure the source:

1. In the export wizard, click the **Source** tab.
2. In the Version drop-down list, select a version.
3. **Optional:** Select **Normal** to display the Version Parameters dialog box, and perform one of the following actions:
 - Select **Use Baseline** as the version type, and click **OK**.
 - Select **Create As-Of** as the version type, specify As-Of criteria, and click **OK**.
4. Configure top nodes:
 - a. Click **Add**.
 - b. In the Select Node dialog box, select a hierarchy and a node, and click **OK**.
 - c. **Optional:** Select a top node, and in the Move column, click the Move Up button () or the Move Down button () to reorder it.

Configuring a Hierarchy Export Style



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Style tab in the export wizard, you select the types of nodes to be included in the export (limb or leaf nodes, or both types), output options, batch validations, and repeat parameters.

Style Options

On the right side of the Style tab, you can select the following style options:

- **Recurse from Top Node** includes descendant nodes below the top node. If cleared, only the top node is included (which is typically not the desired result).
- **Include Inactive Nodes** includes inactive nodes in the export.
- **Include Implicitly Shared Nodes** includes descendant nodes below shared limb nodes that are explicitly shared. Implicit shared nodes are exported only if they are not filtered out by another option. If a query filter is used in the export, you must explicitly include the shared nodes.
- **Remove Duplicates Based on Key** suppresses the export of duplicate records based on the primary key columns that you specified on the Column Options tab.
- **Tabbed Output** indents the nodes for each level of a hierarchy by using tab characters to produce a structured output format. This option is applicable only for exports to files.

Batch Validations

To ensure that the data being exported adheres to the appropriate business rule before further distribution, you can specify validations to run prior to executing the export. If the data fails to pass the specified validation, the export stops.

You can select one of the following options:

- **None** to not run validations.
- **Assigned** to run validations assigned to versions, hierarchies, and nodes included in the export.
- **Selected** to select validations to run from the list.

You can run batch validations interactively by clicking the Run Export Validations button () from the toolbar.

Repeat Parameter Options

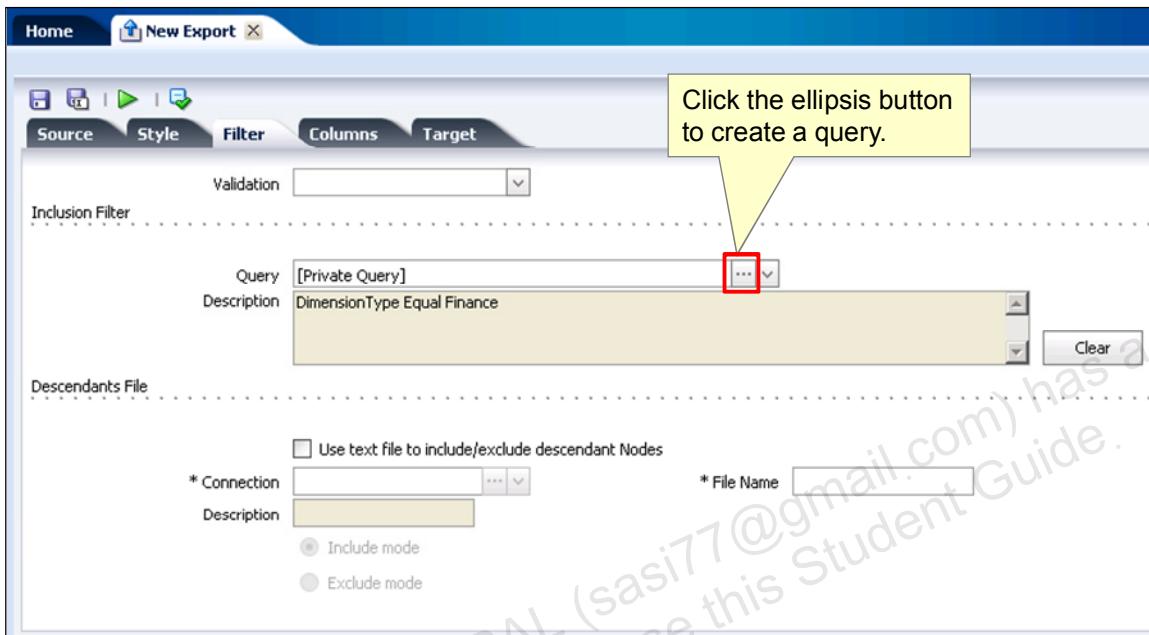
At the bottom of the Style tab, you have the following repeat parameters options:

- **Repeat Number Property** references a local node-level integer property that repeats a node within an export structure.
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. You can define this value as a hierarchy-level integer property (Bottom Level Property) or as an absolute value (Bottom Level Value.) If you define a bottom level, the repeat options are enabled to automatically repeat nodes down to the defined level in the export structure.

To configure the style:

1. In the export wizard, click the **Style** tab.
2. Select whether to include **All Nodes**, **Limb Nodes Only**, or **Leaf Nodes Only**.
3. **Optional:** Select one or more style options.
4. **Optional:** Select batch validations.
5. **Optional:** Select repeat parameters.

Configuring a Hierarchy Export Filter



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Filter tab in the export wizard, you can define filter criteria for the exported data by using a validation, a query, and a text file. Only nodes that fail the validation or that meet the query criteria are included in the export results. The text file must contain a list of node names (one per line) to be included or excluded from the export.

To configure the filter:

1. In the Export wizard, click the **Filter** tab.
2. **Optional:** In the Validation drop-down list, select a validation to use to filter the export.
3. **Optional:** Select or create a query to apply to the export.
4. **Optional:** Configure a text file to filter the export data:
 - a. Select **Use text file to include/exclude descendant nodes**.
 - b. Select a connection, enter a file name, and enter a description of the connection object itself. The description is not an editable field in the export.
 - c. Select **Include mode** or **Exclude mode** to include or exclude nodes from the export.

Configuring Hierarchy Export Columns

The screenshot shows the 'Select Columns' tab of the Oracle Data Relationship Management export wizard. It has two main sections: 'Available' and 'Selected'.

Available:

Label	Namespace
AsOf Transaction ID	Core
Baseline ID	Core
Baseline Name	Core
Copy Time	Core
Default Display By	Core
Default Hierarchy Sort	Core
Default Match By	Core
Default Paste Properties	Core
Description	Core
Display By String	Core
Enable Shared Nodes	Core
Exact Copy Flag	Core

Selected:

Label	Namespace
1 - Name	Core
2 - Allow Adj	FM
3 - Allow Children Adj	FM
4 - Default Parent	FM
5 - Default Currency	FM
6 - FM Description	FM
7 - Holding Company	FM
8 - IC Partner	FM
9 - Security Class	FM
10 - UDA1	Custom
11 - UDA2	Custom
12 - UDA3	Custom

A yellow callout box with the text 'This order determines the column order in the output.' points to the 'Selected' table.

Column Options:

Column Name	Pivot	Skip Defaults	Primary Key
[Core] Name	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[FM] Allow Adj	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[FM] Allow Children Adj	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[FM] Default Parent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[FM] Default Currency	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Columns tab in the export wizard consists of two subtabs: Select Columns and Column Options. On the Select Columns tab, you choose the properties to display as columns in the export results.

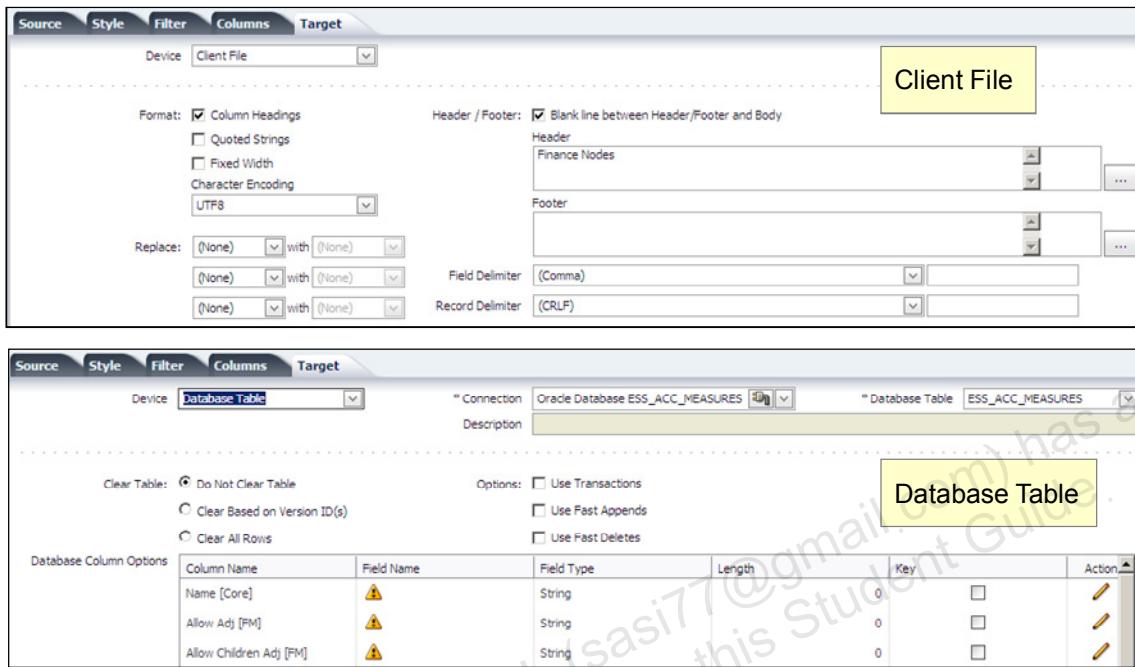
On the Column Options tab, you can select the following options:

- Pivot:** If the field contains a comma-delimited value, then for each value in the list for that column, a row is exported with all other fields set to the same values. You can select the Pivot option on one column.
- Skip Defaults:** Place a blank in the export if the field value equals the default value of the property.
- Primary Key:** Define a field (or fields) as the primary key to be used when determining whether duplicates exist. The Remove Duplicates option on the Style tab uses this key to identify duplicate nodes.

To configure the columns:

1. In the export wizard, click the **Columns** tab.
2. Perform one or more of the following actions to configure the properties to display as columns:
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
3. **Optional:** In the Selected list, select a property, and click the Move Up button () or the Move Down button () to reorder it.
Note: The order of the selected properties controls the order of the columns in the export results. The first item in the list is the first column, the second item in the list is the second column, and so on.
4. Click the **Column Options** tab, and then select **Pivot**, **Skip Defaults**, and **Primary Key** options for properties.

Configuring a Hierarchy Export Target



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On the Target tab in the export wizard, you configure where the data is exported to and the format of the export results. You can export data to a client file, server file, or database table.

Client and Server File Options

You have the following formatting options when exporting data to a client or server file:

- **Column Headings** includes column headings in the export.
- **Quoted Strings** puts quotation marks around strings in the export.
- **Fixed Width** enables you to set field widths, set left or right justification for columns, and specify the number of pad characters.
- **Character Encoding** exports the results as UTF8, ANSI, ASCII, UTF16BE, or UTF16LE.
- **Replace options** enables you to specify up to three characters in the export file to replace; for example, you can replace all commas with tabs.
- **Header/Footer** enables you to add a line between the header and footer and the body or enter text to be printed in the header and footer of the export results. You can use customization tags to insert runtime values into the header or footer of the export results.
- **Delimiter options** enables you to select a field delimiter character and a record delimiter character.

Database Table Options

You have the following database options when exporting data to database tables:

- **Do Not Clear Table** appends the export data 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** performs the export 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 on 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** enables you to map each export column to a database field.

To configure the target:

1. In the export wizard, click the **Target** tab.
2. In the Device drop-down list, select **Client File**, **Database Table**, or **Server File**.
3. **Optional:** If you selected to export to a client file, configure formatting options.
The Fixed Width Option section is displayed when you select the Fixed Width option.
Note: For the header and footer, click the ellipsis button () to open a text editor.
4. If you selected to export to a database table, perform the following tasks:
 - a. In the Connection drop-down list, select an external connection.
 - b. In the Database Table drop-down list, select the database table to which you want to export the data.
The list of available tables is controlled by the selected tables for the database connection.
 - c. Select an option for clearing the database table.
 - d. **Optional:** Select one or more of the following options:
 - **Use Transactions**
 - **Use Fast Appends**
 - **Use Fast Deletes**
 - e. In the Database Column Options table, map column names to field names. If you selected Clear Based on Version ID(s) in step c, specify a Key field.
5. If you selected to export to a server file, perform the following tasks:
 - a. In the Connection drop-down list, select an external connection.
 - b. In the File Name box, enter a file name.
 - c. **Optional:** Configure formatting options.

Saving Exports

- Save exports before running them.
- Specify the following when saving exports:
 - Name of export
 - Description of export
 - Object access group
 - User
 - Standard
 - System
 - Custom object access group

ORACLE

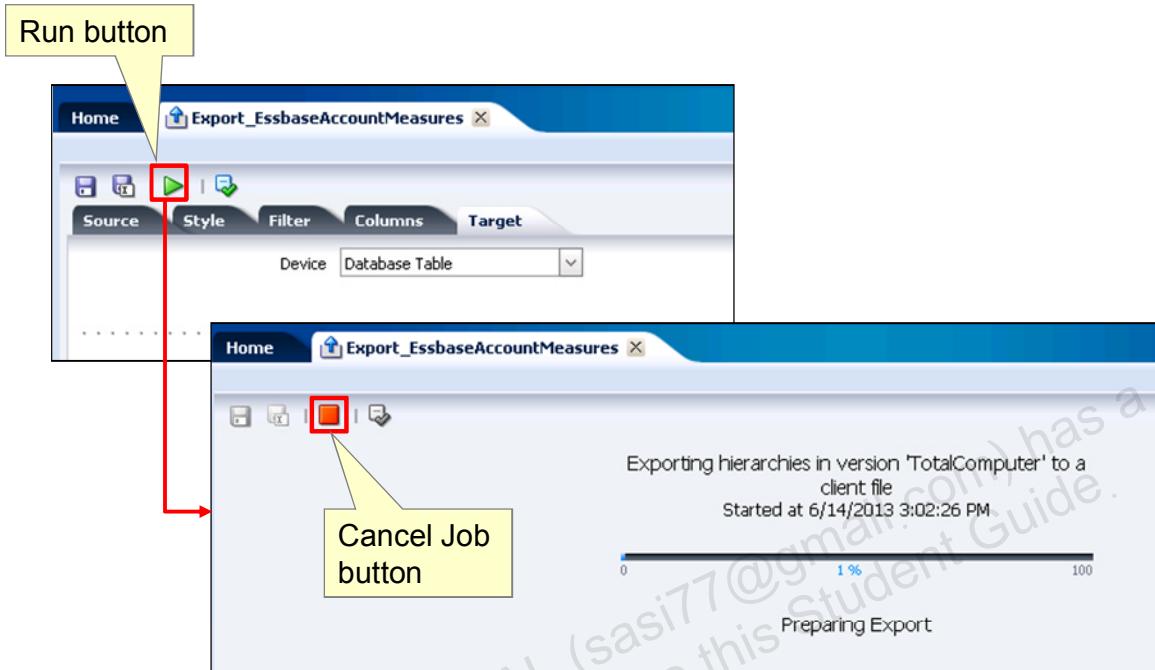
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You should save exports before running them.

To save exports:

1. On the toolbar, click the Save button ().
The Save As dialog box is displayed.
2. Enter a name for the export and an optional description.
3. Select an object access group (User, Standard, System, or a custom object access group).
4. Click **OK**.

Running Exports



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You must open an export and run it in the export wizard. When exporting to a file, you are prompted to save or open the export file.

To run exports:

1. If necessary, perform the following tasks:
 - a. Open an export in the Export task group.
 - b. On the Source tab, select a version.
The list of assigned hierarchies and top nodes for the export are verified for the selected versions. Only valid hierarchies and top nodes are included.
 - c. For a compare or difference export, select a From and To version.
The version parameters default to an existing version if the Default Current Version or Default Previous Version system preferences are set.
2. **Optional:** Edit export parameters. You can save or discard these changes.
3. On the toolbar, click the Run button ().
The export is being run as a job. If you export to a file, the File Download dialog box prompts you to open or save the file.
4. Open or save the export file.

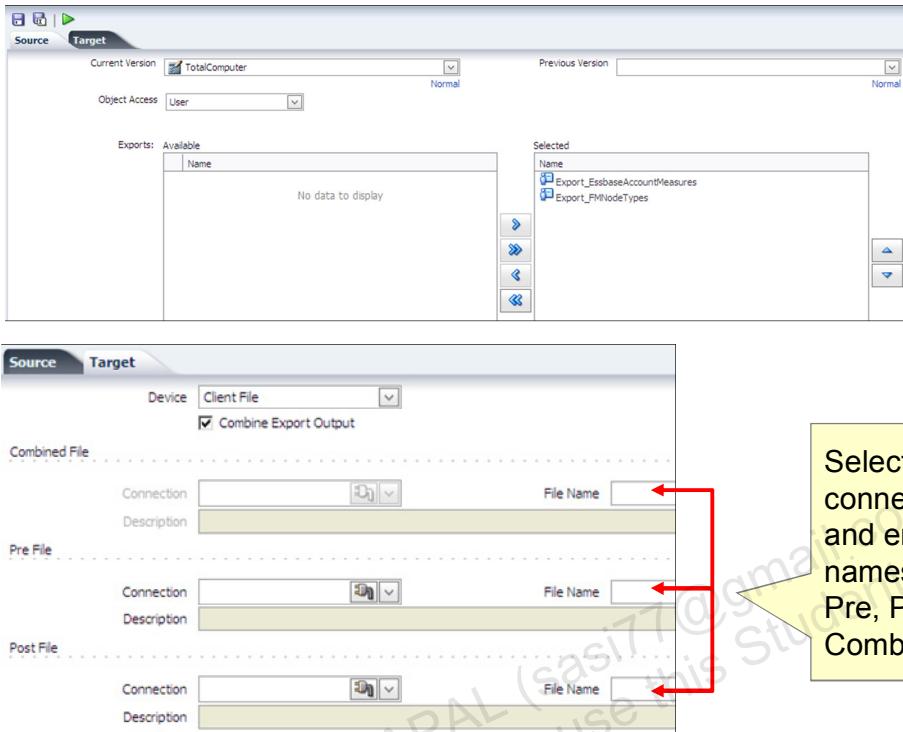
Batch Validations

The following rules apply for batch validations assigned to exports:

- When an export is run, assigned batch validations are executed as a pre-processing step to verify that all data to be exported is valid before it is output. If a failure is encountered, the export will be aborted and the validation failure will be reported.
- You can run batch validations interactively from the export instance page using the Run Export Validations button to the right of the Run button. This will run validations assigned to the export profile for the specific data set to be exported. All validation failures will be returned as validation results on the Browse-Home page.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Creating and Running Export Books



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create books to group multiple exports to be run together with a single request. You can output the exports in the book to separate targets, or you can combine them into a single output file. When you create a book, you work through two tabs: Source and Target.

Books are restricted to certain exports based on object access rules. You can add standard exports to any book but can only add exports in an object access group to a book from the same group.

If the output is combined, you must specify the combined file, connection, and file names. Use the Pre File and Post File options to specify data that is prepended and appended to the combined output file. When a book contains both exports configured to output to a file and some configured to output to a database table, the Combine Export Output option affects only the exports that are writing to files.

To create export books:

1. On the toolbar in the Export task group, click the New Book button ().
2. The New Book tab is displayed, and the Source subtab is displayed by default.

3. Configure the **Source** tab:

- a. In the Current Version drop-down list, select a version from which to select exports for the book.
Note: All exports for a book must reference the same version.
- b. **Optional:** In the Previous Version drop-down list, select a version for compare and difference exports in the book.
- c. In the Object Access drop-down list, select an access type for the book: **User**, **Standard**, **System**, or a custom object access group.
- d. Perform one or more of the following actions to specify the exports for the book:
 - Select exports in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all exports in the Available list to the Selected list.
 - Select exports in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all exports in the Selected list to the Available list.

4. Click and configure the **Target** tab:

- a. In the Device drop-down list, select **Server File** or **Client File**.
- b. **Optional:** If the output is combined, select **Combine Export Output** and specify the device (combined file), connection, and file names. If needed, configure the Pre File and Post File sections.
- c. On the toolbar, click the Save As button ().
The Save Book As dialog box is displayed.
- d. Enter a name and an optional description, and click **OK**.

To run export books:

1. In the Export task group, click the **Books** tab, and double-click a book to open it.
2. On the Source tab, select a current and previous version.
3. On the toolbar, click the Run button ().

Summary

In this lesson, you should have learned how to:

- Describe the exporting of data
- Explain export types
- Open, copy, and delete exports
- Create exports
- Run exports
- Create and run export books

Quiz

To which of the following can you export data?

- a. Database tables
- b. Server files
- c. Client files
- d. Web pages



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz

Which one of the following export types is used as the basis for the majority of exports?

- a. Generation Export
- b. Hierarchy Compare Export
- c. Hierarchy Export
- d. Hierarchy XML Export



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz

A Hierarchy Compare Export is used as a delta (difference) export to generate only the differences between hierarchies.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

If you want to export data to a database table, you need to create an external connection to that database table.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

13

Managing Security

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain security levels
- Create and manage users
- Assign access to nodes
- Assign access to property categories
- Assign access to user metadata objects
- Grant anonymous access

Security Levels

User roles

determine high-level access to data and features.



Property access

determines the type of access to properties.



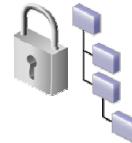
Object access

determines the user metadata objects that users can run and manage.



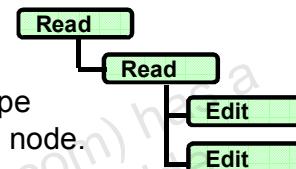
Version status

determines whether the version can be viewed or modified.



Node access

determines the type of access to each node.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

Data Relationship Management provides security at the following levels:

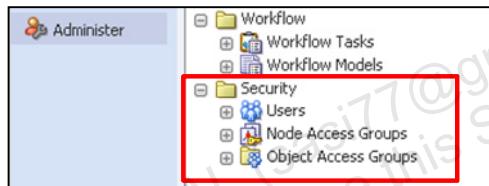
- **User roles** provide user-authenticated and anonymous access to data and Data Relationship Management features. You can assign permissions at a granular level in order to allow lower-level users to perform higher-level functions without giving them complete higher-level permissions. Roles segregate metadata administration permissions from security administration permissions. The following roles map to Data Relationship Management audiences:
 - **Access Manager** manages users and their assignment to security features.
 - **Anonymous User** provides views of data that users can access anonymously through a URL.
 - **Application Administrator** manages system metadata for an application.
 - **Data Creator** creates new versions and hierarchies, but cannot manage data created by other users.
 - **Data Manager** manages all versions and hierarchies, user group access to the data, and standard metadata objects.
 - **Interactive User** views, creates, and edits nodes in hierarchies based on node access group and property category security.

- **Governance User** submits and approves requests for changes to nodes and properties using workflow tasks, workflow models, and workflow node access groups.
- **Workflow User** participates in external workflow processes for changes to hierarchies and nodes using node access group and property category security.
Note: For more information about roles and their default permissions, see the *Oracle Data Relationship Management Administrator's Guide, Release 11.1.2.3*.
- **Version status** determines whether a version is available for viewing or modifying. There are four statuses:
 - **Working** enables any user with edit access to view and modify the version.
 - **Submitted** enables any user to view the version. Version owners and data managers can modify it.
 - **Finalized** enables any user to view the version, but not modify it.
 - **Expired** enables version owners and data managers to view the version.
- **Property access** determines property access for versions, hierarchies, and nodes. Property access is assigned through property categories.
- **Node access** determines the nodes that users can view or edit. Node access is assigned through node access groups.
- **Object access** determines the user metadata objects that users can run and manage. User metadata objects include queries, compares, imports, blenders, exports, and books.

Managing User Security

You manage user security in the Administer task group by performing the following tasks:

- Define users and their roles.
- Define node access groups and assign users to groups.
- Assign users to existing property categories or define new property categories to give users access to specific properties.
- Define object access groups and assign users to groups.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You create and manage users in the Administer task group.

User Authentication Modes

Data Relationship Management security supports three user authentication modes:

- **Internal:** Data Relationship Management (default)
- **Common Security Services:** Oracle Hyperion Shared Services
 - OID
 - LDAP
 - MSAD
- **Mixed:** Based on individual user settings



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Data Relationship Management security model includes three methods for authenticating users:

- **Internal:** User authentication is managed internally by Data Relationship Management.
- **Common Security Services:** User authentication is managed by Oracle Hyperion Shared Services, and user login information is maintained in a central authentication directory, such as Oracle Internet Directory (OID), Lightweight Directory Access Protocol (LDAP), or Microsoft Active Directory (MSAD).
- **Mixed:** System and security administrators can assign either of the previous authentication methods to Data Relationship Management users.

Creating Users

The screenshot shows the 'New User' creation interface. The 'Roles' tab is active. The 'Available' list contains roles such as Access Manager, Anonymous User, Application Administrator, Data Creator, Data Manager, Governance User, and Workflow User. The 'Selected' list contains 'Interactive User'. Between the two lists are four blue double-headed arrows: a top-left arrow, a top-right arrow, a bottom-left arrow, and a bottom-right arrow. At the bottom right of the interface is an 'ORACLE' logo.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To create users:

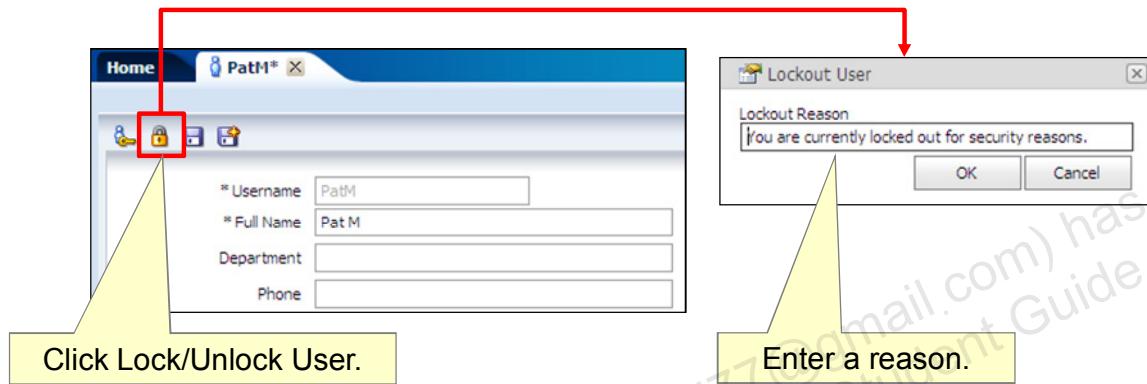
1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **User**.
3. In the Username box, enter a user name.
4. In the Full Name box, enter the full name of the user.
5. **Optional:** In the Department, Phone, and Email Address boxes, enter the appropriate information.
6. On the Roles tab (which is displayed by default), perform one or more of the following actions to assign roles to the user:
 - Select roles in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all roles in the Available list to the Selected list.
 - Select roles in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all roles in the Selected list to the Available list.

7. On the Node Access Groups tab, perform one or more of the following actions to assign node access groups to the user:
 - Select node access groups in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all node access groups in the Available list to the Selected list.
 - Select node access groups in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all node access groups in the Selected list to the Available list.

Note: If you selected the Data Manager role in step 6, you need not assign node access groups to the user.
8. On the Property Categories tab, perform one or more of the following actions to assign property categories to the user:
 - Select property categories in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all property categories in the Available list to the Selected list.
 - Select property categories in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all property categories in the Selected list to the Available list.
9. For each property category in the Selected list, click the Edit button (), set the user's access to **Read** or **Edit**, and then click the Update button ().
10. Click the Save button ().
11. If the user authentication mode was defined as Internal during installation, the Change User Password dialog box is displayed.
 - a. Enter and reenter a password for the user.
 - b. **Optional:** Select **User must change password at next login**.
 - c. Click **OK**.

Locking and Unlocking Users

When you lock out a user, you can provide a reason for the lockout.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you lock out a user, you can provide a reason for the lockout. When a locked user attempts to log on, the lockout message is displayed in the Error dialog box.

To lock out users:

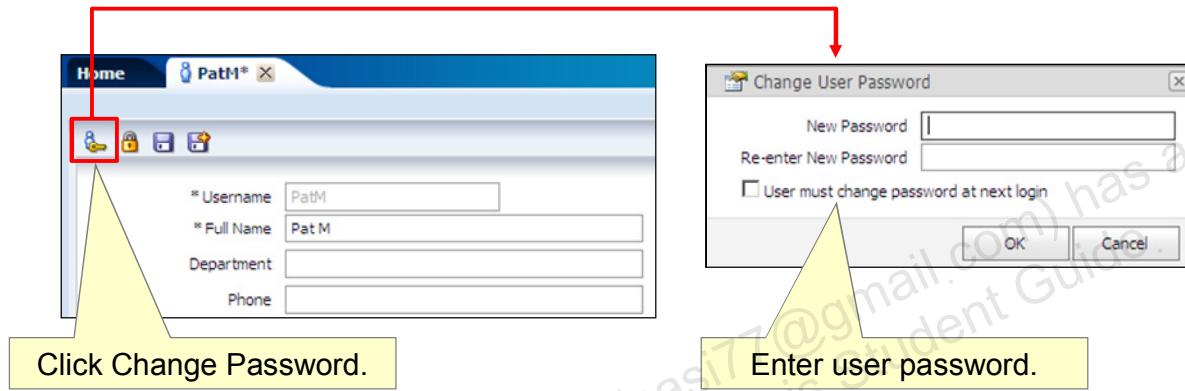
1. On the Home page, select the **Administer** task group.
2. Under Security, expand **Users**, and select a user.
3. On the toolbar, click the Edit button ().
The user tab is displayed.
4. On the toolbar, click the Lock/Unlock User button ().
The Lockout User dialog box is displayed.
5. In the Lockout Reason box, enter a reason for the lockout, and click **OK**.

To unlock users:

1. Under Security in the Administer task group, expand **Users**, and select a user.
2. On the toolbar, click the Edit button ().
The user tab is displayed.
3. On the toolbar, click the Lock/Unlock User button ().

Changing User Passwords

- Access managers can change user passwords.
- Users can change their own passwords through user preferences.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the user authentication mode was defined as Internal during installation, both users and access managers can change passwords.

To change user passwords as an access manager:

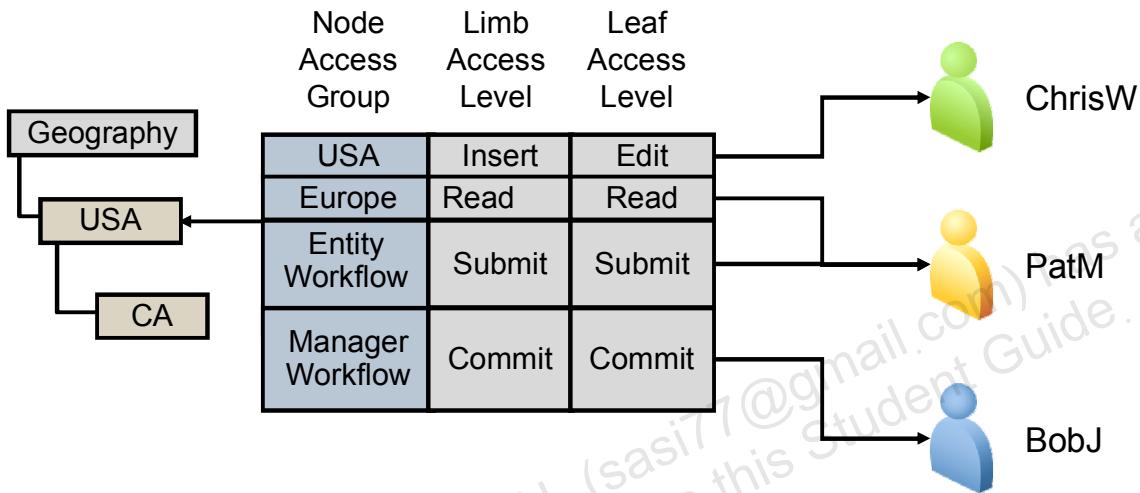
1. Under Security in the Administer task group, expand **Users**, and select a user.
2. On the toolbar, click the Edit button ().
The user tab is displayed.
3. On the toolbar, click the Change Password button ().
4. In the Change User Password dialog box, enter and reenter a password, and click **OK**.
5. **Optional:** Select **User must change password at next login**.
6. Click **OK**.
7. Click **Save**.

To change your own password:

- At the top of Web Client, click **Preferences**, click **Change My Password**, and then change your password.

Managing Access to Nodes

Node access groups control user access to nodes in hierarchies.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To control granular user access to nodes in hierarchies, you use node access groups. Node access groups are defined by access managers and typically represent functional areas of an organization. When you assign a user to a node access group, the user is granted access to nodes according to the access levels assigned to nodes for the group. You can assign users to multiple node access groups. If there is a conflict in access levels because a user is a member of multiple node access groups, the highest security level is used.

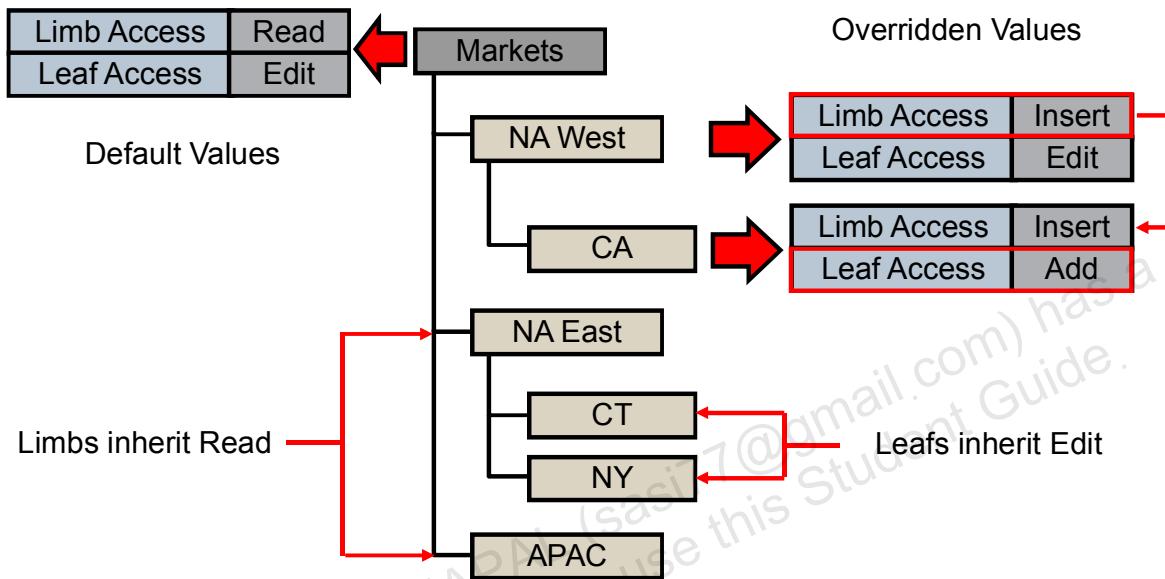
There are two types of node access groups. The group type controls the type of data access that can be assigned to users of that group. Each node access group can be of only a single group type.

- **Interactive** enables direct access to browse, search, and modify data based on the level of access assigned to the user.
- **Workflow** enables restricted access to browse, search, and modify data using governance workflows based on the level of access assigned to the user.

In the slide example, access to the USA node is defined for four node access groups (USA, Europe, EntityWorkflow, and ManagerWorkflow). The USA group can insert limb and edit leaf nodes. The Europe group can read both limb and leaf nodes. The EntityWorkflow group can submit requests for changes to both limb and leaf nodes. The ManagerWorkflow group can commit workflow requests for changes to both leaf and limb nodes.

Interactive Group-Type Node Access Levels

Inheritable properties control the assignment of access levels to hierarchy nodes.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If an interactive node access group has access to any node in a hierarchy, the entire hierarchy is visible to all users who are members of that node access group. This access allows users to navigate to the nodes to which they have access. Conversely, if a node access group has no access to any node in a hierarchy, users who are members of that node access group cannot view or edit the hierarchy.

To simplify the administration of user access to hierarchies, you assign node access levels to nodes with an inheritable property for each node access group. Two special property categories are available for these node access group properties:

- **Limb Access** determines access level for the node and its limb descendants.
- **Leaf Access** determines access level for the node and its leaf descendants.

By default, node access groups inherit assignment values locally in each hierarchy. You can override those values for a specific node, in which case the corresponding property values for all dependants default to the overridden value.

In the slide example, the value of Limb Access for NA West was overridden to Insert; it was inherited by CA. The value of Leaf Access for CA was overridden to Add. The rest of the nodes inherited values from the top node Markets.

Note: Because you can assign multiple tiers of access levels to a specified group, security based on node access groups is more flexible than the access determined by a user's assignment to a property category.

The following table describes interactive group-type node access levels and provides usage examples:

Access Level	Description	Example Usage
Read	Provides read-only access No changes are permitted.	Viewing and reporting
Limited-Insert	Allows insertion of a node for which the user has (at least) global insert privilege	Insert
Edit	Allows editing of property values	Edit
Insert	Allows insertion, copying, moving, and removal of nodes	Edit Insert Copy Move Remove
Inactivate	Allows inactivation and reactivation of nodes	Edit Insert Copy Move Remove Inactivate Reactivate
Add	Allows addition and deletion of nodes	Edit Insert Copy Move Remove Inactivate Reactivate Add Delete

Workflow Group-Type Node Access Levels

- Notify
- Submit
- Approve
- Enrich
- Commit



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

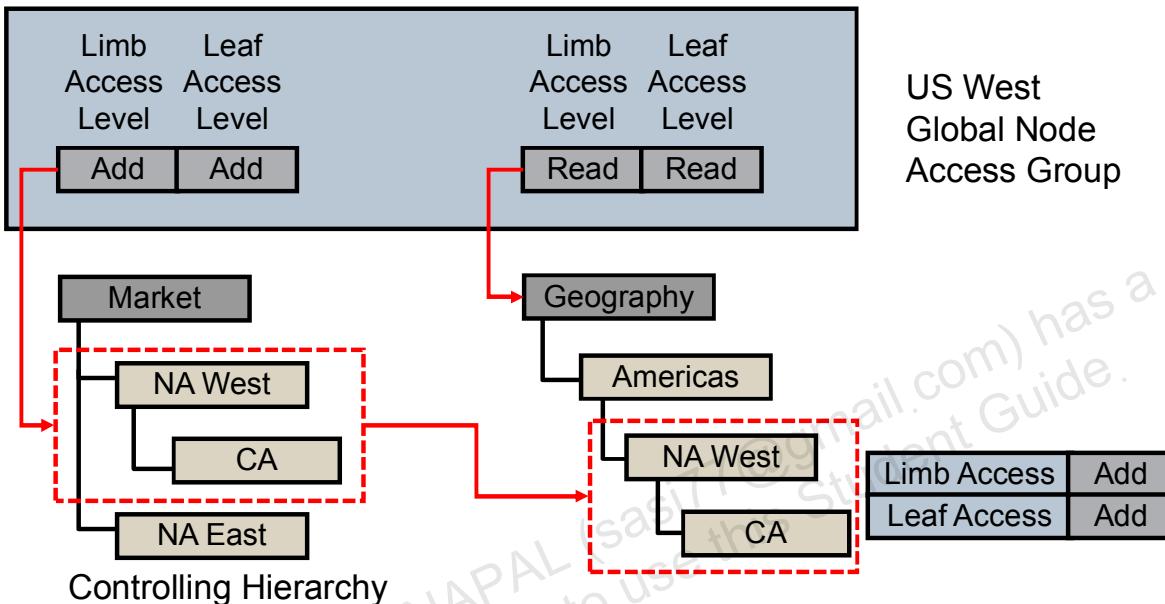
The following table describes workflow group-type node access levels:

Access Level	Description
Notify	Allows users to be notified of requests for changes to nodes as part of a governance workflow. The Notify level is also used to provide read-only access to nodes selected as part of a request.
Submit	Allows users to submit requests for changes to nodes as part of a governance workflow
Approve	Allows users to approve requests for changes to nodes as part of a governance workflow
Enrich	Allows users to enrich requests for changes to nodes as part of a governance workflow
Commit	Allows users to commit requests for changes to nodes as part of a governance workflow

Note: Workflow node access levels are cumulative; assignment of the Enrich access level implies that users can also approve, submit, and be notified.

Global Node Access Groups

Global node access groups control user access to nodes across hierarchies.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

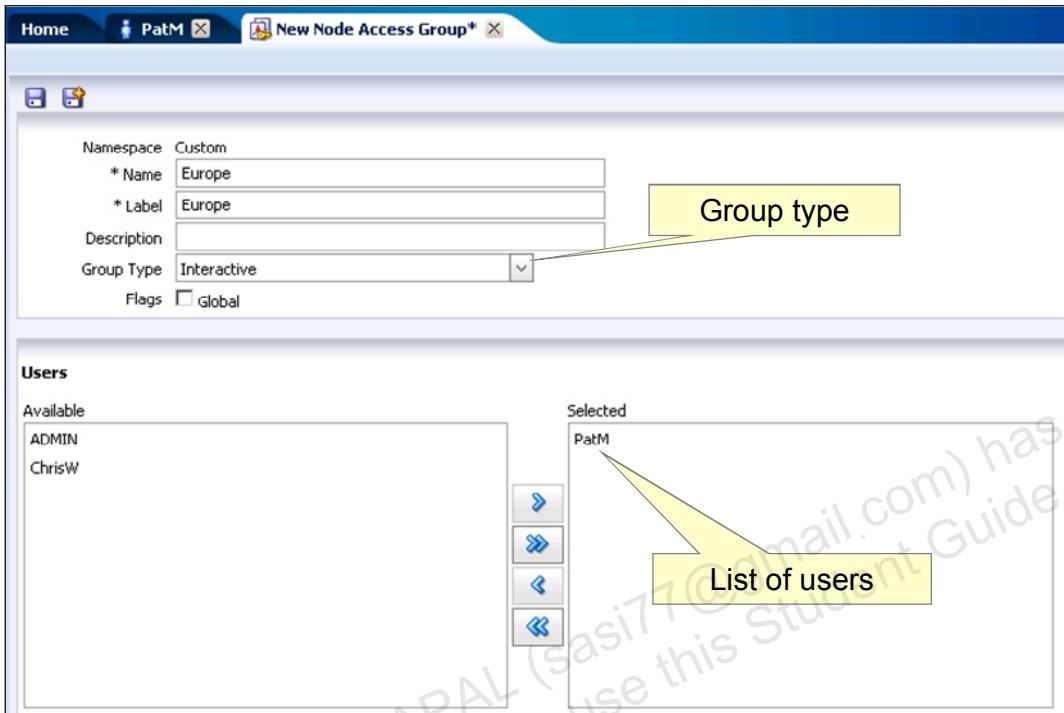
Nodes may exist in more than one hierarchy in a version. If this situation occurs, you must assign node access in each hierarchy to ensure the same access level to the same nodes.

Alternatively, you can use global node access groups and specify a controlling hierarchy for the group. For global groups, the leaf and limb access levels to nodes in the controlling hierarchy are used for all hierarchies in which the nodes are located.

In the slide example, the NA West node and its child node CA are in both the Markets and Geography hierarchies. You have two options:

- You can create a node access group and assign it specific limb and leaf access levels to the NA West node in the Markets and Geography hierarchies. For example, in the Markets hierarchy, the node access group can add limb and leaf nodes to the NA West node. In the Geography hierarchy, the node access group can only read limb and leaf nodes at the NA West node level.
- You can create a global node access group named US West and specify the Markets hierarchy as the controlling hierarchy. Because the NA West node in the Markets hierarchy has add leaf and limb access levels, the NA West node in other hierarchies also have add leaf and limb access levels.

Creating, Editing, and Deleting Node Access Groups



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

You can create, edit, and delete node access groups in the Administer task group. When creating node access groups, you can choose whether to make the group global. For a global group, you must also define a controlling hierarchy.

To create node access groups:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Node Access Group**.
3. In the Name box, enter a name for the group.
4. In the Label box, enter a label for the group.
The label is used for display on the Properties tab in a hierarchy tab.
5. **Optional:** In the Description box, enter a description for the group.
6. In the Group Type drop-down list, select one of the following group types:
 - **Interactive** to use interactive access levels
 - **Workflow** to use workflow-oriented access to versions, hierarchies, and nodes in the context of submitting, enriching, approving, committing, and being notified of requests
7. **Optional:** Select **Global** to make the group a global node access group.

8. Perform one or more of the following actions to assign users to the node access group:
 - Select users in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all users in the Available list to the Selected list.
 - Select users in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all users in the Selected list to the Available list.
9. On the toolbar, click the Save button ().
10. If you selected Global in step 7, assign a controlling hierarchy for the global node access group:
 - a. Return to the **Browse** task group.
 - b. Right-click the controlling hierarchy, select **Assign Control**, and then **Node Access Groups**.
The Set Controlled Node Access Groups dialog box is displayed.
 - c. Select the global node access group in the Available list, and click the Select button () to move it to the Selected list.
 - d. Click **OK**.

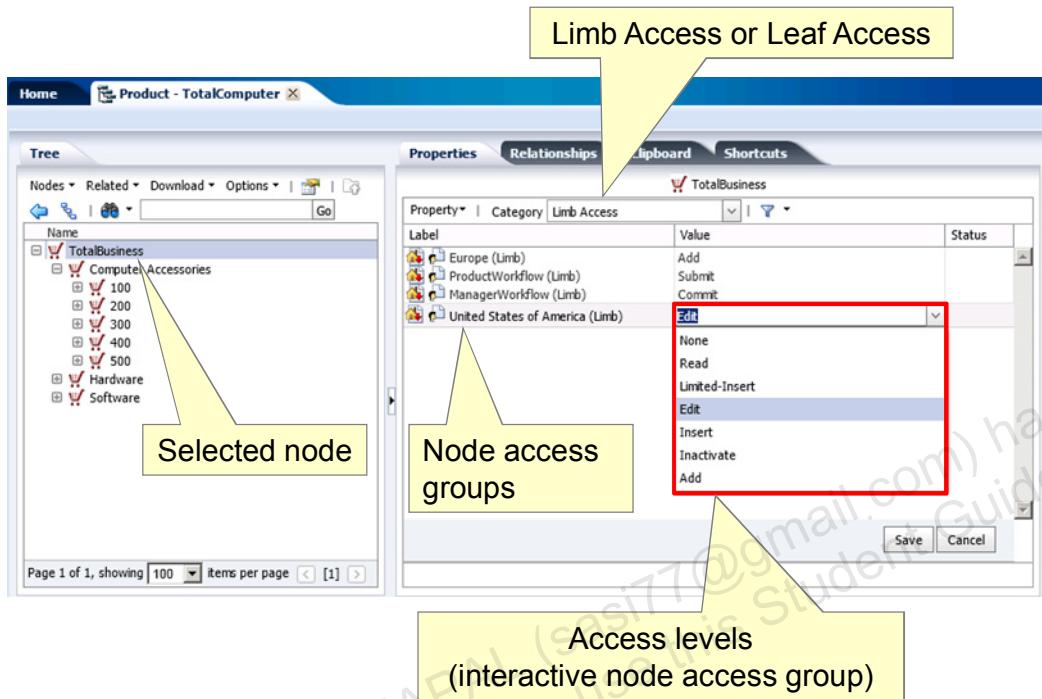
To edit node access groups:

1. Under Security in the Administer task group, expand **Node Access Groups**, and select a group.
2. On the toolbar, click the Edit button ().
3. **Optional:** Modify the label and description.
4. Perform one or more of the following actions to update the assigned users in the node access group:
 - Select users in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all users in the Available list to the Selected list.
 - Select users in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all users in the Selected list to the Available list.
5. On the toolbar, click the Save button ().

To delete node access groups:

1. Under Security in the Administer task group, expand **Node Access Groups**, and select a group.
2. On the toolbar, click the Delete button ().
The Confirm Delete dialog box is displayed.
3. Select **Delete this NodeAccessGroup**.
All access level assignments for the group are removed from all versions.

Configuring Limb and Leaf Access for Node Access Groups



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data managers can configure limb and leaf access for node access groups on the Properties tab for nodes in open hierarchies. The Properties tab has Leaf Access and Limb Access property categories for this purpose.

You can assign a user to multiple node access groups, which you can assign to multiple hierarchies. For interactive node access, if the user belongs to multiple node access groups with different access levels, the Leaf Access and Limb Access properties display the highest access level.

To configure leaf and limb access for node access groups:

1. Ensure that appropriate node access groups are created and appropriate users are assigned to the groups.
2. In the Browse task group, open a version and a hierarchy.
3. In the hierarchy tree, right-click a node, and select **Assign** and then **Node Access**.

On the Properties tab, the Leaf Access and Limb Access categories are available. Node access groups are listed.

4. In the Category drop-down list, select **Leaf Access** or **Limb Access**.
5. In the Value column for node access groups that need access, select an access level.
6. Click **Save**.

Assigning Access to Property Categories

Interactive property access is controlled through property categories.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Interactive property access is controlled through property categories. All user-defined properties are assigned to one or more property categories. Users are assigned Read, Edit, or no access to each property category. Determining whether users need full access or read-only access to property categories may help you to determine how properties should be categorized.

By default, users without the Data Manager role have access only to the System category. You can, however, grant users access to other categories if needed. For the Shared Info and Stats property categories, you can only assign Read access.

If needed, you can make the System category read-only. This enables users to edit properties for their area of responsibility in other property categories, but does not enable users to modify common attributes that are used across multiple business functions.

If a user has no access to a property category, the category and its properties are not accessible to the user at the version, hierarchy, or node level. To modify the value of a property for a node, users must have at least Edit access to the node and Edit access to the property category to which the property belongs.

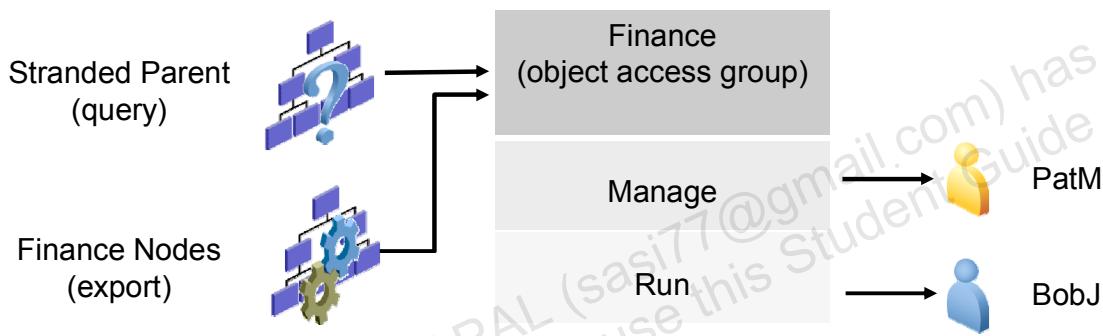
To assign access to property categories:

1. In the Administer module, expand **Property Categories**, and select a property category.
2. On the toolbar, click the Edit button ().
The property category tab is displayed.
3. Click the **Users** tab.
4. Perform one or more of the following actions to assign users to the property category:
 - Select users in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all users in the Available list to the Selected list.
 - Select users in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all users in the Selected list to the Available list.
5. On a user row, click the Edit button ().
A drop-down list is displayed in the Access column.
6. In the drop-down list in the Access column, select an access level (**Read** or **Edit**).
7. On the user row, click the Update button ().
8. Repeat steps 5 to 7 for other selected users.
9. On the toolbar, click the Save button ().

Note: A user can be assigned to multiple property categories while creating or editing the user.

Assigning Access to Metadata Objects

- Object access groups control user access to metadata objects (exports, books, imports, blenders, compares, queries, versions variables, and external connections).
- Permission levels are:
 - Run
 - Manage



ORACLE

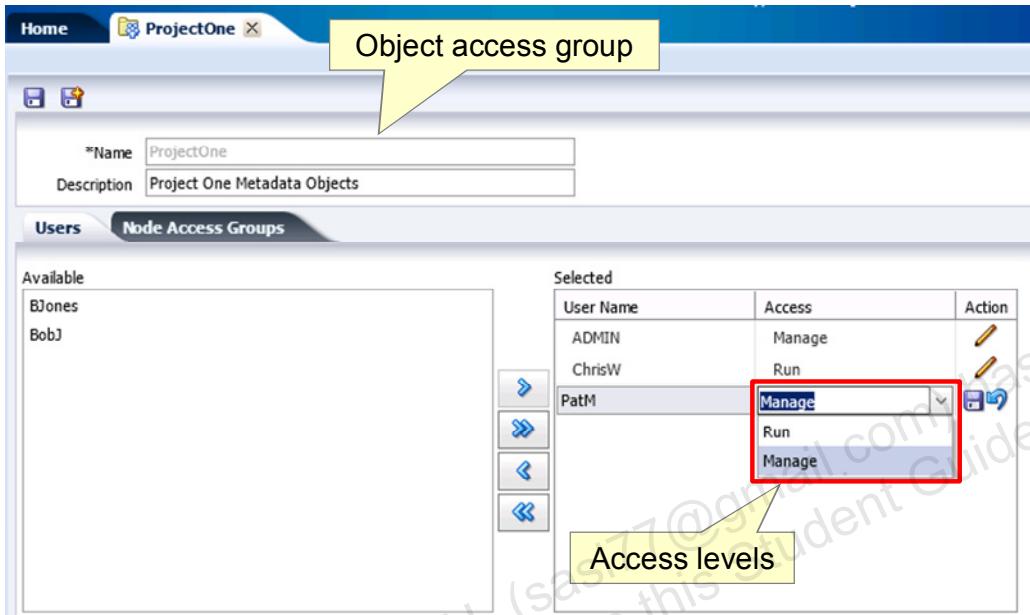
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Object access groups in Data Relationship Management control delegated user access to specific metadata objects, including exports, books, imports, blenders, compares, queries, versions variables, and external connections.

Individual users may be directly assigned to object access groups or groups of users can be indirectly assigned using node access groups. Metadata objects are assigned to object access groups at the time they are created, and they may subsequently be copied or moved to a different group. There are two object access permission levels:

- **Run** enables users to run objects in the group; however, users cannot edit and save changes to the objects.
- **Manage** enables users to create, edit, or delete objects in the group and run them.

Creating, Editing, and Deleting Object Access Groups



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create, edit, and delete object access groups in the Administer task group.

To create object access groups:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Object Access Group**.
3. In the Name box, enter a name for the group.
4. **Optional:** In the Description box, enter a description for the group.
5. On the Users tab, perform one or more of the following actions to assign users to the node access group:
 - Select users in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all users in the Available list to the Selected list.
 - Select users in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all users in the Selected list to the Available list.

Note: By default, each user is granted Run access. To change a user's access, click the Edit button (). Then from Access, select **Manage**, and click the Save button ().

6. On the Node Access Groups tab, select node access groups from the Available list to assign to the object access group. Use the arrows to move node access groups to the Selected list.

Note: By default, each node access group is granted Run access. To change a group's access, click the Edit button (). Then from Access, select **Manage**, and click the Save button ().

7. On the toolbar, click the Save button ().

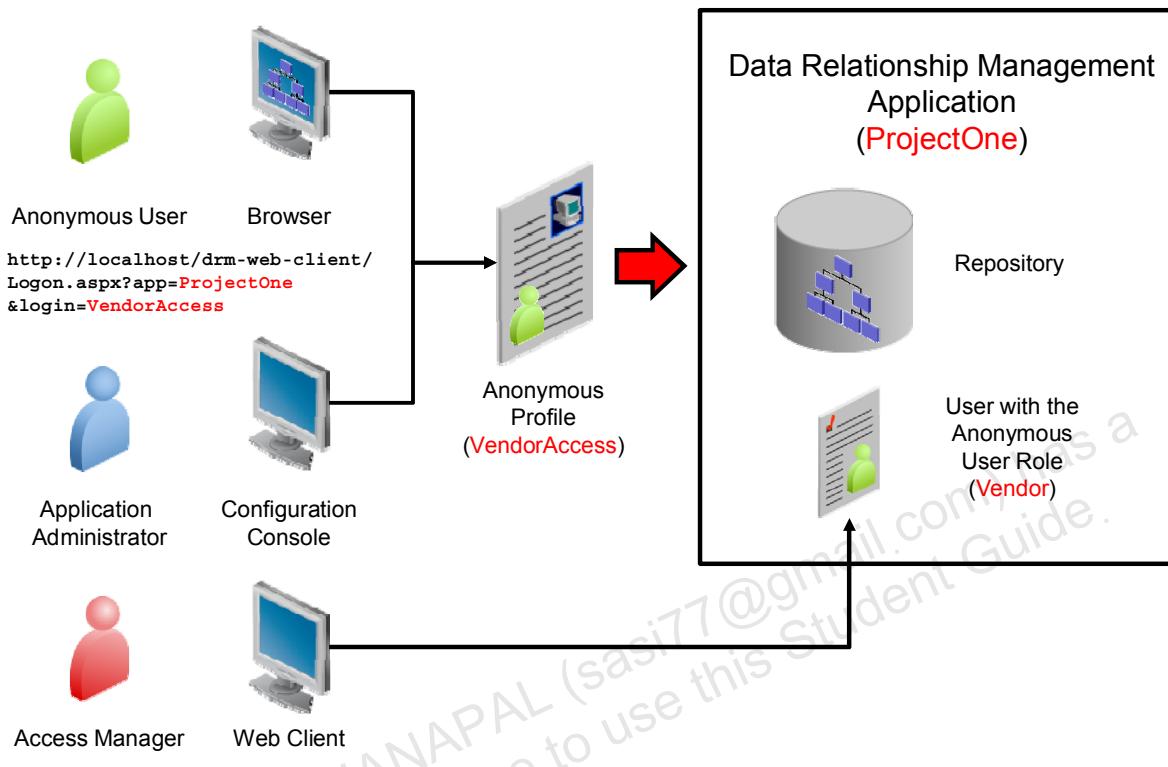
To edit object access groups:

1. Under Security in the Administer task group, expand **Object Access Groups**, and select a group.
2. On the toolbar, click the Edit button ().
3. **Optional:** Modify the description.
4. On the Users and Node Access Groups tabs, make changes to selected users and node access groups and to access permissions.
5. On the toolbar, click the Save button ().

To delete object access groups:

1. Under Security in the Administer task group, expand **Object Access Groups**, and select a group.
2. On the toolbar, click the Delete button ().
The Confirm Delete dialog box is displayed.
3. Select **Delete this Object Access Group**.
The object access group and all assignments for the group are deleted.

Granting Anonymous Access



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Anonymous profiles, which you can define in Data Relationship Management Configuration Console, enable users to access a URL that provides them access to a public view of data without being authenticated. Users with anonymous access can access the Browse, Query, Compare, and Export task groups in Web Client. The URL exposes only the profile name. When a user accesses Web Client, the associated user name and password are retrieved internally, and the logon page is not shown.

The anonymous access URL is created in this format:

```
http://<hostname>/drm-web-client/Logon.aspx?app=<Application_Name>&login=<Anonymous_Profile_Name>
```

Note: Anonymous profiles must be set up separately for each machine that hosts Web Client.

To grant anonymous access to Data Relationship Management applications:

1. In the Administer task group in Web Client, create a user, assign the **Anonymous User** role to the user, assign node access groups and property categories to the user, and specify the access level for each property category.
2. In Configuration Console, create and save an anonymous profile with username and password equal to the user name and password that you defined in step 1.

Summary

In this lesson, you should have learned how to:

- Explain security levels
- Create and manage users
- Assign access to nodes
- Assign access to property categories
- Assign access to user metadata objects
- Grant anonymous access

Quiz

Which of the following are security levels for Data Relationship Management?

- a. Version status
- b. User role
- c. Node access
- d. Property access
- e. Object access



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d, e

Quiz

Identify the statements that correctly describe user creation.

- a. You can rename users.
- b. You can assign users to node access groups.
- c. You can assign users to object access group.
- d. You can lock out users.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b, c, d

Quiz

Which of the following are valid workflow node access group levels?

- a. Notify
- b. Update
- c. Approve
- d. Commit



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c, d

Quiz

A user can be assigned to an object access group directly or by using a node access group to which the user is assigned.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

14

Configuring Governance Workflows

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

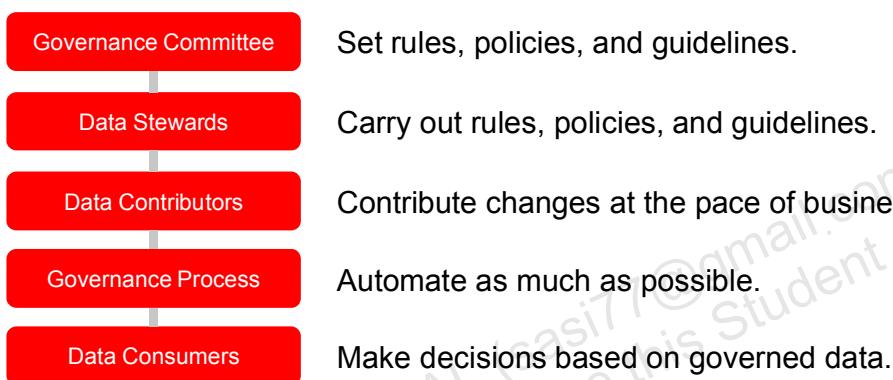
Objectives

After completing this lesson, you should be able to:

- Describe data relationship governance
- Create and manage workflow tasks and models
- Create, submit, open, claim, and approve requests
- Navigate the worklist
- Receive and view alerts and notifications

Data Governance

- “Data governance encompasses the people, processes, and information technology required to create a consistent and proper handling of an organization’s data across the business enterprise.” (Wikipedia)
- Governance responsibilities:



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

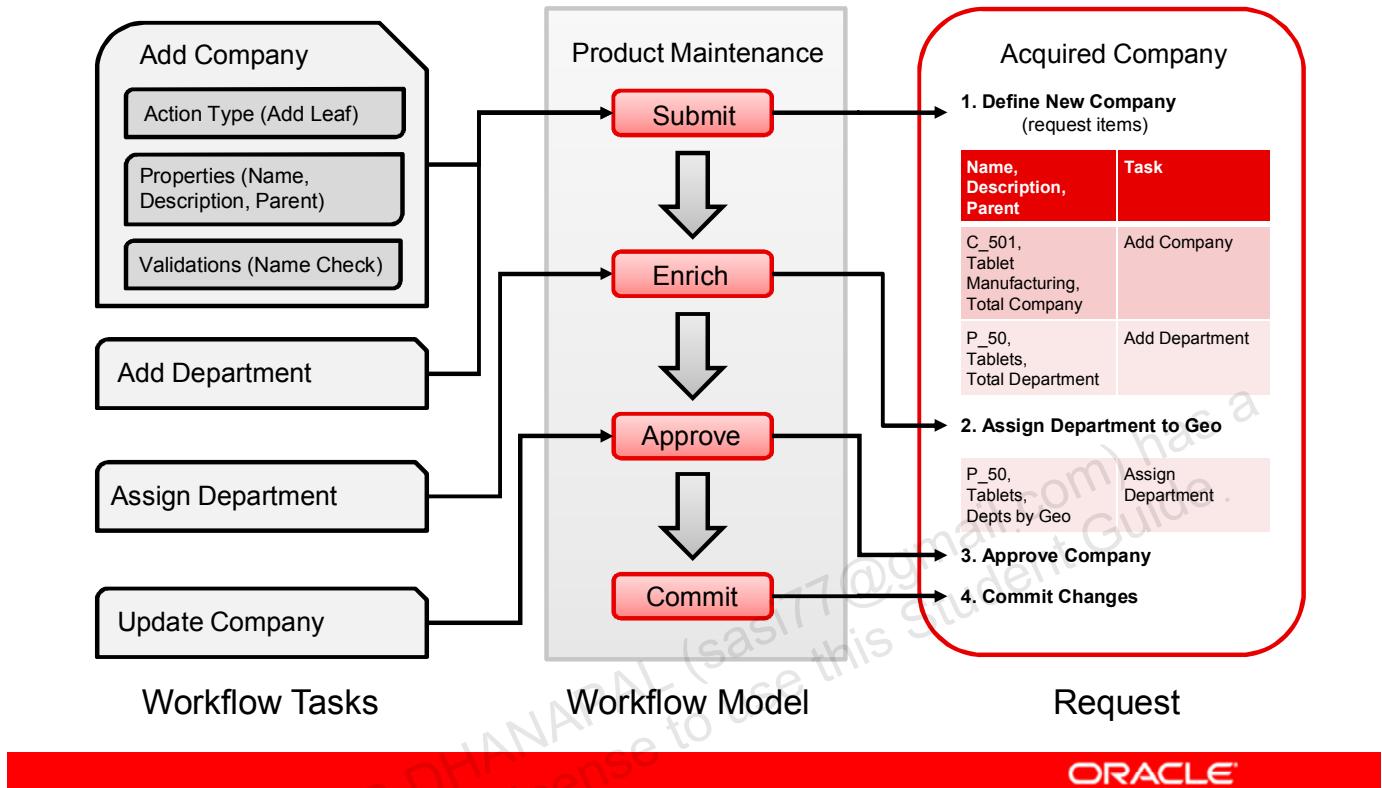
Data governance represents the convergence of data quality, data management, data policies, business process management, and risk management surrounding the handling of information as an asset within the enterprise. Through data governance, organizations can exercise positive control over the processes and methods used by their data stewards and data custodians to handle data.

The Oracle Data Relationship Governance module addresses the governance challenge in two critical ways:

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

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

Governance Workflow Objects



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

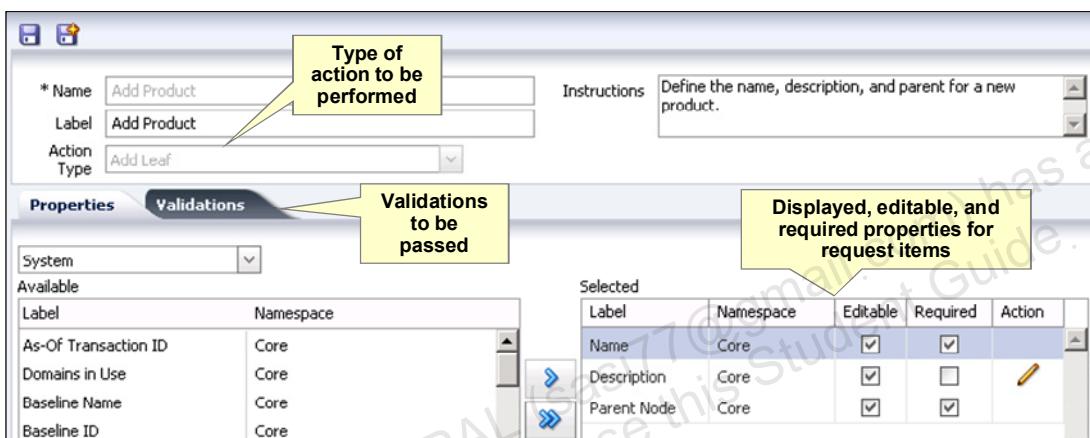
The Oracle Data Relationship Governance module implements workflows using the following objects:

- Workflow tasks represent change actions in the system. The same workflow task can be used in many workflow models to ensure consistency of changes performed in different business processes.
- Workflow models define business processes by controlling tasks, stages of workflow, and types of data involved to govern a particular set of changes in Data Relationship Management.
- Requests are based on a particular workflow model and represent a collection of changes for a specific set of data. Requests can be categorized as change requests and remediation requests. Change requests are direct requests created by governance users who have Submit access to both a data set and a workflow model. Remediation requests are typically created from the results of another operation such as a batch validation, query, or comparison. Governance users can then add nodes from those results to a remediation request.
- Request items define individual changes to the actual data based on the workflow tasks.

The diagram in the slide shows relationships between the workflow objects.

Workflow Tasks

- Define custom governance tasks to simplify and control the types of changes that can be made in a request or during a particular workflow stage
- Can be reused across multiple workflow models



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Workflow tasks are building blocks of workflow models. Each workflow task defines a single set of changes performed for a local node; it consists of an action type, properties to be viewed or edited, and validations.

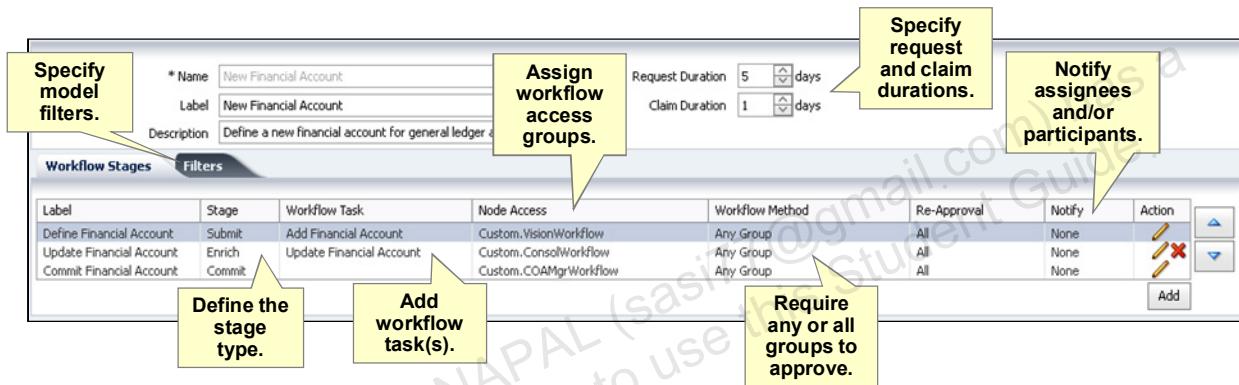
- The action type specifies the basic type of action being performed, such as adding, moving, or updating the node. Each action type defines rules regarding the selection of nodes and parents, application of property updates, and the actions to perform when the request is validated and committed.
- Task properties may be configured to control whether property values are display-only or can be edited. Editable properties may be configured as required. Default properties for an action type may not be removed from the task.
- Task validations are optional node-level batch validations that must be successfully executed before the request can be approved for a particular workflow stage. For each validation, task properties may be assigned in order to associate an error with the associated property values. The validations run just for the local nodes associated with request items in a request.

To create workflow tasks:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Workflow Task**.
The New Workflow Task tab is displayed.
3. Enter a name for the workflow task.
4. **Optional:** Enter a label and instructions.
5. In the Action Type drop-down list, select one of the following options:
 - **Add Leaf** to add a leaf node with global and local properties
 - **Add Limb** to add a limb node with global and local properties
 - **Delete** to update a node's global/local properties and delete the node
 - **Inactivate** to update a node's global and local properties and deactivate the node
 - **Insert** to insert a node into a hierarchy and update its global/local properties
 - **Move** to move a node to a different parent and update its global/local properties
 - **Remove** to update a node's global/local properties and remove the node
 - **Update** to update global and local properties for a nodeThe default set of properties for the action type is displayed in the Selected list on the Properties tab.
6. Perform one or more of the following actions to configure the properties to display as columns:
 - In the Category drop-down list, select a property category to ease navigation.
 - Select properties in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all properties in the Available list to the Selected list.
 - Select properties in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all properties in the Selected list to the Available list.
7. **Optional:** In the Selected list, select a property, and click the Move Up button () or the Move Down button () to reorder it. The order of the properties here controls the order of the properties when viewing or editing properties when performing the task within a request.
8. For each property in the Selected list, click the Edit button () to change the Editable or Required option, and then click the Update button ().
9. On the Validation tab, select validations from the Available list to assign to the task. Use the arrows to move validations to the Selected list. Use the up and down arrows to order the validations.
10. For each validation in the Selected list, click the Edit button () to configure the properties for the validation, and then click the Update button (). This configuration enables flagging the specified properties in a request with the validation message when the validation fails
11. On the toolbar, click the Save button () to save the workflow task.

Workflow Models

- Workflow models structure a governance process for a change or remediation request into an ordered set of workflow stages for approval, enrichment, and commitment.
- Workflow stages control which users can participate, when they can participate, which actions and tasks they can perform, and who is notified of their activity.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Workflow models control the user tasks, stages of workflow, and types of data involved to govern a particular set of changes to data in Data Relationship Management. Each workflow model consists of multiple workflow stages that define approvals and enrichment steps required before the changes can be committed to a version. When a stage is assigned to a workflow model, the stage type attribute defines the type of participation for users in that stage of the workflow. A workflow model has:

- A single Submit stage that is defined by a list of allowable workflow tasks that can be used to submit changes. The Submit stage is always the first stage of a workflow model.
- An optional series of Approval or Enrichment stages, each consisting of a single workflow task. Properties for workflow tasks are read-only for Approval stages (regardless of how they are configured for the task) but can be edited in an Enrichment stage if configured for editability in an Enrichment stage.
- A single Commit stage used for final approval and commitment of all workflow tasks performed. The Commit stage is always the last stage of a workflow model.

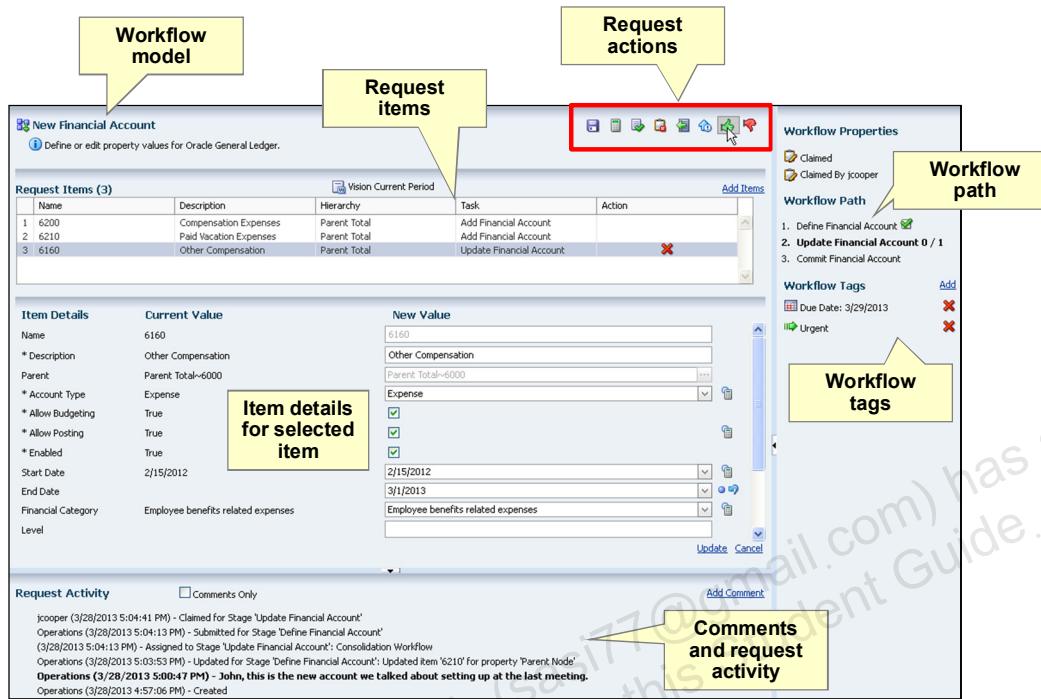
To create workflow models:

1. On the Home page, select the **Administer** task group.
2. In the New drop-down list, select **Workflow Model**.
The New Workflow Model tab is displayed.
3. Enter a name for the workflow model.
4. **Optional:** Enter a label and description.
5. **Optional:** Enter the number of days for these options:
 - **Request Duration** indicates the expected amount of time a request should take to be approved and committed. If the request duration is less than the age of the request, the request is marked as Overdue.
 - **Claim Duration** indicates the expected amount of time a user should have a request claimed for a workflow stage. If the stage age exceeds the claim duration, the request is automatically unclaimed to allow other users to participate.

Note: A value of zero for either option indicates that the Overdue and automatically Unclaimed functionality is disabled for the workflow model.
6. On the Workflow Stages tab, for each workflow stage, click the Edit button () and configure the following options:
 - In the Label text box, enter descriptive text to be used for the stage in workflow paths.
 - In the Stage drop-down list, select the stage type.
 - In the Workflow Task drop-down list, select workflow tasks to be available for users of the stage.
 - In the Node Access drop-down list, select node access groups to be associated with the workflow stage. Only node access groups of the Workflow type can be assigned to a stage and only the users of the selected node access groups are able to participate for the stage.
 - In the Workflow Method drop-down list, select one of the following options to specify which node access groups must approve the stage in a request:
 - **Any Group** to specify that a user from any assigned node access group may approve the request in order to advance it to the next workflow stage. The node access group must be assigned to the hierarchy with access to the current stage type or greater. If none of the assigned access groups to the stage has proper data access to the request items in the request, the stage may be skipped as long as required values are provided and validations pass for all request items.
 - **All Groups** to specify that at least one user from all assigned node access groups must approve the request before it advances to the next stage. If none of the access groups assigned to the stage has proper data access to the request items in the request, the request is escalated to Data Managers for resolution.
 - In the Re-Approval drop-down list, select whether changes made in each stage while in pushback mode are required to be reapproved by other users. Select one of the following options:
 - **Current** to specify that the change to the request in this stage must be reapproved for the current stage only. After approval, the request is assigned to the user who previously pushed back the request.

- **All** to specify that the change to the request in this stage must be reapproved for subsequent stages.
 - In the Notify drop-down list, select one of the following options to specify whether and when to send alerts and notifications to workflow users each time there is request activity that should be communicated to users of the workflow model:
 - **None** to disable notification of actions performed for the stage
 - **Assignees** to notify users who are assigned to the current stage of the request
 - **Participants** to notify users who have participated in the request
 - **Assignees and Participants** to notify assignees and participants
7. **Optional:** Click **Add** to add Approve or Enrich stages to the workflow model, and then follow the previous step for each stage added.
 8. **Optional:** On the Filters tab, make selections to restrict the versions, hierarchies, and node types that users can view and select for a particular type of request.
 9. On the toolbar, click the Save button () to save the workflow model.

Working with Requests



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Requests are used to initiate changes for a specific set of data to be performed after the request is validated, approved, and committed. Each request uses a single workflow model to control which users participate in the request, when they participate, and their type of participation.

Request Items

A request may consist of one or more request items that may be added during initial submission or later enrichment of the request. The order of request items is important because it controls the order in the changes applied to the target version for validation and commitment purposes.

Request Actions

All activities taking place for a request as well as user comments are recorded in the request activity. Each activity includes a timestamp and name of the user if applicable. Activities are displayed with the most recent activity at the top of the list.

When you are working with a request, action buttons in the upper-right corner of the page header allow you to perform actions on the request. Request actions include:

- **Save** () saves the request to the Data Relationship Management repository in its current state without validating or approving the request.
- **Calculate** () calculates request item details based on proposed changes in the request.

- **Validate** () validates proposed changes for request items in the request.
- **Submit** () validates the request items in the request and, if successful, assigns the request to the next stage in the workflow path.
- **Copy** () copies request items to a new request.
- **Claim** () claims the request for the active workflow stage.
- **Unclaim** () removes the claimed lock from the request but does not validate nor assign the request to a different stage, user, or group.
- **Pushback** () pushes back the request to a previous user in the current stage or a previous stage to correct some element of a request item or ask for more information about the request.
- **Escalate** () escalates the request to the Data Manager user role.
- **De-escalate** () reassigned the request to the next node access group for the current workflow stage.
- **Approve** () approves and commits the changes in the request to the target version.
- **Reject** () rejects the request.

Workflow Path

The workflow path helps you understand how long a request may take, how many approvals may be involved, and where a request is positioned in the overall approval process. The workflow path is automatically updated as users perform actions.

Workflow Tags

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

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

Navigating the Worklist

The Worklist is a central location for interacting with requests.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

From the Worklist, governance users submit change requests or review and participate in requests assigned to their user group. Data managers submit remediation requests to governance users or participate in escalated requests submitted by them.

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

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

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

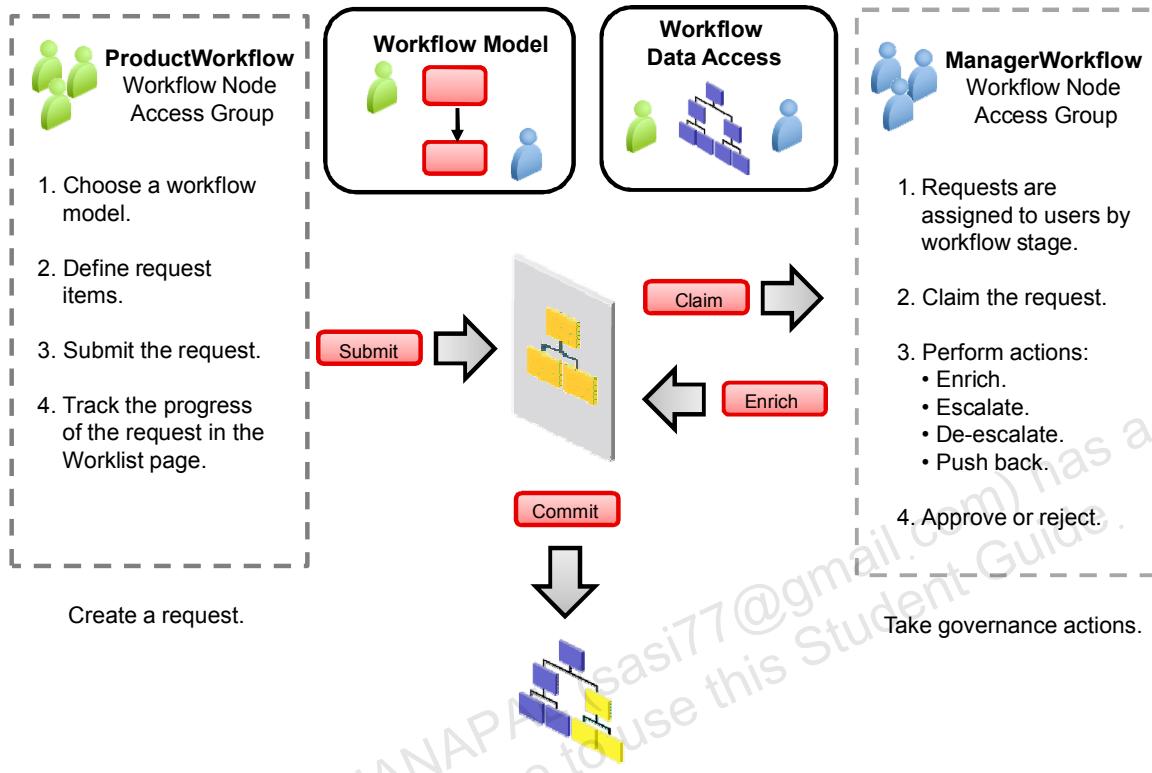
Governance users can search requests by the request header, request items, workflow tags, or comments.

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

Governance users can preview, open, and delete requests from the Worklist.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Request Life Cycle



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The life cycle of a request is determined by its workflow model. The workflow model controls the stages of workflow, the tasks that can be performed, and the users who can participate in the request. During its life cycle, a request must go through at least two workflow stages (Submit and Commit) but, depending on the model, it may need to go through additional approval or enrichment stages before it can be committed. The actions that can be performed by each participating user depend on the stage to which they are participating and their workflow access to the data referred to by the request items.

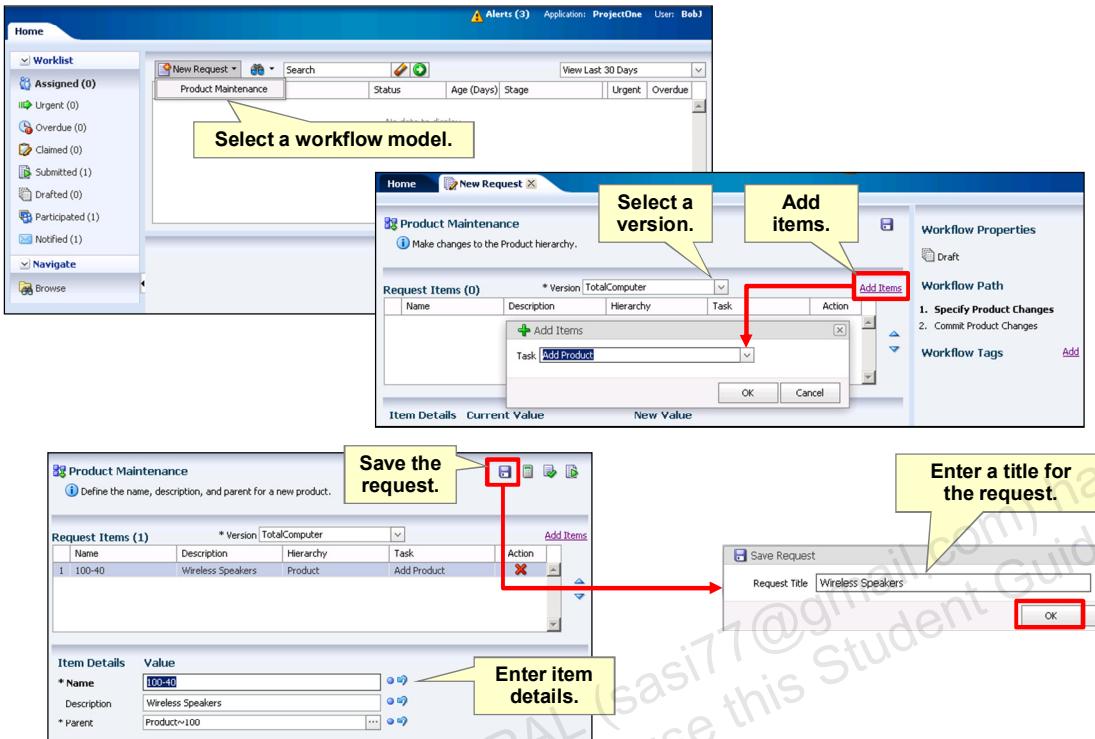
In a typical workflow, governance users create requests and submit them for approval. After the request has been submitted, the request is assigned to the next workflow stage in the workflow model and the request can no longer be edited by the submitting user. Governance users can track the progress of submitted requests from the Worklist page.

An assigned request must be claimed by a user before any other action can be performed on it. The user who claims the request makes it unavailable for any other assigned user to claim it and can perform operations based on the active workflow stage. If the user approves, pushes back, escalates, or rejects the request, the request is automatically unclaimed and the user no longer can take action on the request for that workflow stage. The user can also explicitly unclaim the request, which makes the request available for any other assigned user to claim for the active workflow stage. The approval of a request signifies that the user has performed the tasks assigned to them during their workflow stage and agrees to allow the request to proceed toward being committed.

Users assigned to the final Commit stage in the workflow model must provide their approval to commit the changes in the request to the target version of data. During the Commit stage, all required values for all request items must be provided and all validations must pass. An approver can edit any request item to address remaining issues or make final adjustments. After the request is ready to be committed, the approver provides his or her approval to initiate the commit operation. If there are multiple approvers for the Commit stage, the last approval will trigger the commitment of changes in the request. If the approver rejects the request, the workflow is cancelled and users are no longer able to perform any action on the request. A rejected request may be later copied and submitted as a new request if applicable.

For more details about request life cycles, see the chapter titled “Governance Workflows” in the *Oracle Data Relationship Management User’s Guide Release 11.1.2.3*.

Creating Requests



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Governance users or data managers who are assigned to the initial Submit stage of a workflow model can create requests.

To create requests:

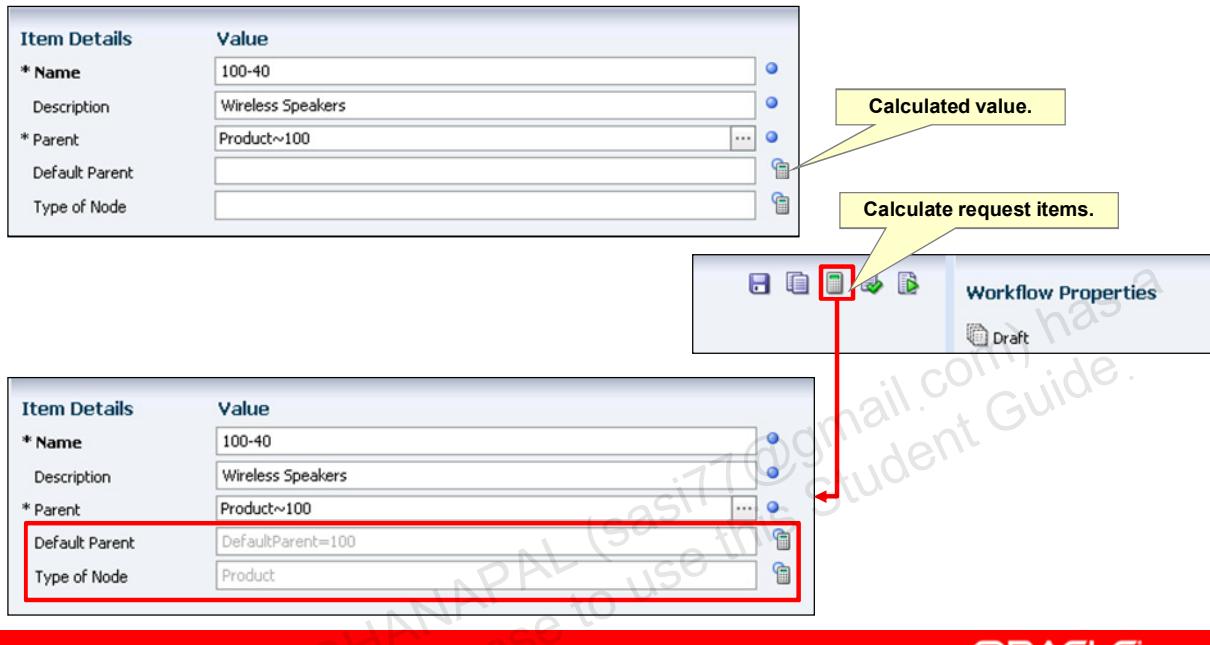
1. On the Home page, select the **Worklist** task group.
2. In the New Request drop-down list, select a workflow model to use for the request. The New Request tab is displayed.
3. In the Version drop-down list, select a target version for the request.
Note: If the workflow model uses a version variable, the version is preselected and displayed as a read-only label and cannot be changed.
4. Click **Add Items**.
The Add Items dialog box is displayed.
Note: Depending on the scenario, the Add Items dialog box displays one or more fields.
5. In the Task drop-down list, select a workflow task and click **OK**.
The Item Details section displays properties for the request item. Required values are marked with an asterisk (*).
6. Enter the request item details and click **Update** to temporarily save changes for the item or click **Cancel** to discard the changes.

7. Repeat steps 4 through 6 to add items to the request.
8. In the Request Actions toolbar, click the Save button ().
The Save Request dialog box is displayed.
9. **Optional:** Perform the following steps to add a comment to the request:
 - In the Request Activity section, click **Add Comment**.
The Add Comment dialog box is displayed.
 - Enter a comment and click **OK**.
The comment is displayed in the Request Activity section with user and timestamp information.
10. **Optional:** Perform the following steps to add tags to the request:
 - In the Workflow Tags section, click **Add**.
The Add Tags dialog box is displayed.
 - Select **Due Date** and enter a date.
 - Select **Urgent**.
 - Click **OK**.
The selected tags are assigned to the request and displayed in the Workflow Tags section.
11. In the Request Title box, enter a title for the request.
12. Click **OK**.

The request is saved as a draft and is ready to be submitted for approval.

Calculating Request Items

Verify inherited and derived values for request items before submitting the request.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

If the workflow task displays inherited and derived properties, you can verify their values before submitting the request by calculating request items. Calculated request items are identified in the Item Details list by the “Calculated value” icon () next to the item value. After calculating request item values, you may decide to override the calculated value if the property is configured to be overrideable.

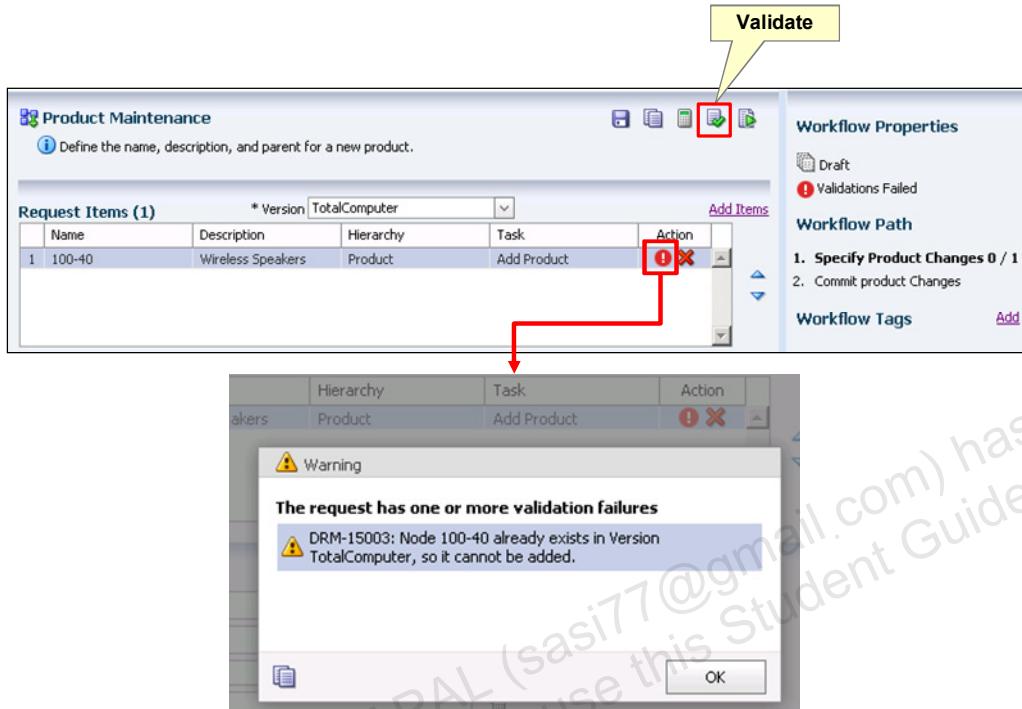
To calculate request items:

- In the Request Actions toolbar, click the Calculate button ().

Values of the calculated request items are calculated based on the values entered for the non-calculated request items.

Note: Changes to request items must be saved in order to calculate values based on those changes. The Calculate action automatically saves the request in order to calculate values.

Validating Requests



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Validate action checks required values and runs batch validations configured for the task for the request item or the current workflow stage. Validation failures are returned for correction.

Requests can be validated if:

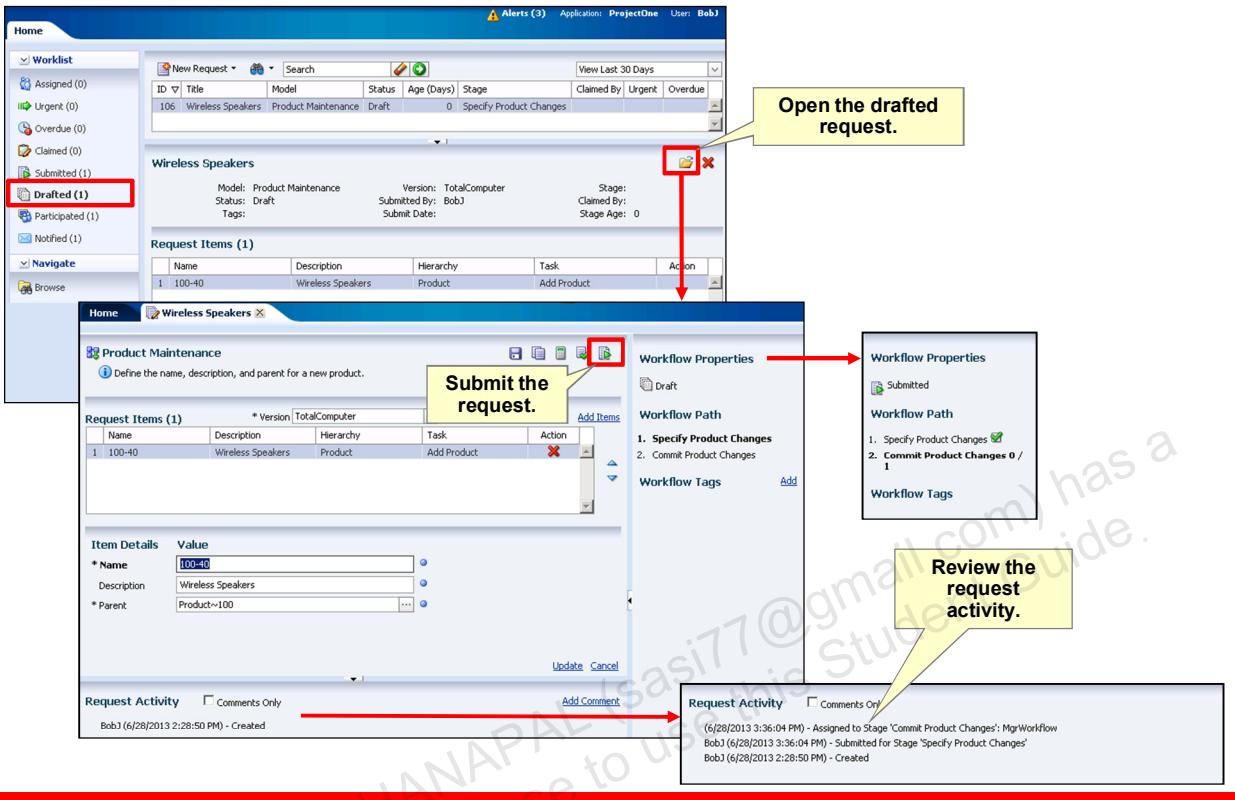
- The request is in the Draft workflow status and has at least one request item.
- The request is in the Claimed workflow status.

To validate the request:

1. In the Request Actions toolbar, click the Validate button (). Values of the request items are validated. The request items that failed the validation are marked with validation failure indicators.
2. If the validation fails, click the Show Validation Errors button () for the failed request item to display the validation error.
3. In the Item Details section, the Validation Failure icon indicates which properties may need to be corrected to resolve a specific validation failure.

Note: Changes to request items must be saved in order to validate those changes. The Validate action automatically saves the request in order to validate request items.

Submitting Requests for Approval



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

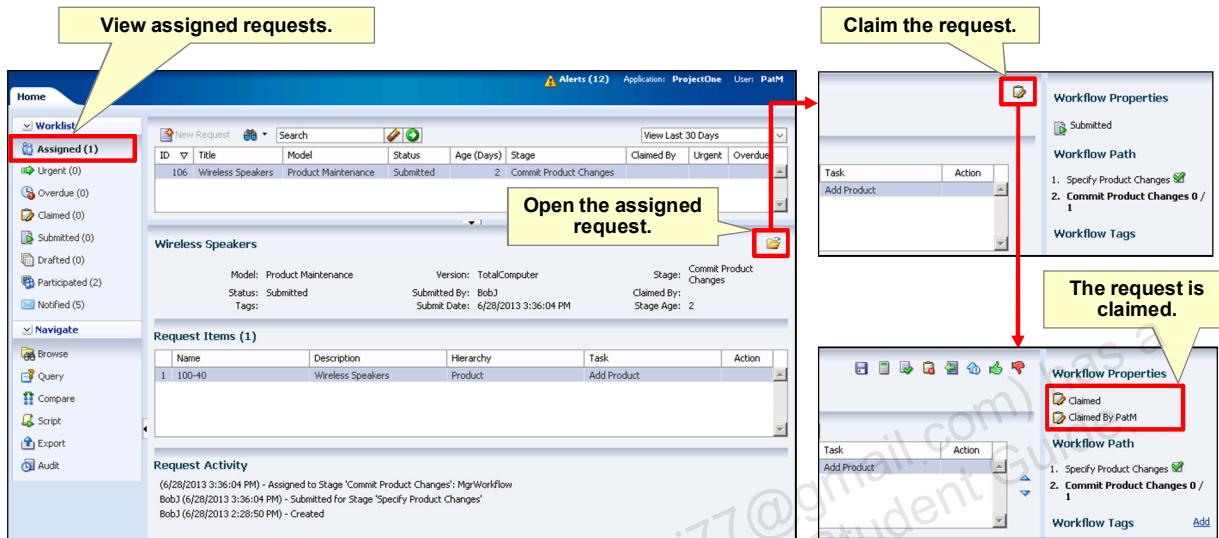
ORACLE

After a request has been created and saved, the request can be submitted for approval, enrichment, and commitment by other governance users using the workflow model of the request.

To submit drafted requests:

1. In the Worklist task group, select **Drafted**.
A list of all drafted requests is displayed.
2. Select a request and click the Open button ().
The request is displayed in a new tab.
3. In the Request Actions toolbar, click the Submit button ().
The request is submitted for approval and its status changed to Submitted. The Recent Activity area displays taken actions.

Claiming Requests



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a request is assigned to a workflow stage, all user groups associated with the stage have the request assigned to them. An assigned request must be claimed by a user before any other action can be performed on it. The user who claims an assigned request makes it unavailable for any other assigned user to claim it. While the request is claimed, the user can perform any operations made available to them based on the active workflow stage.

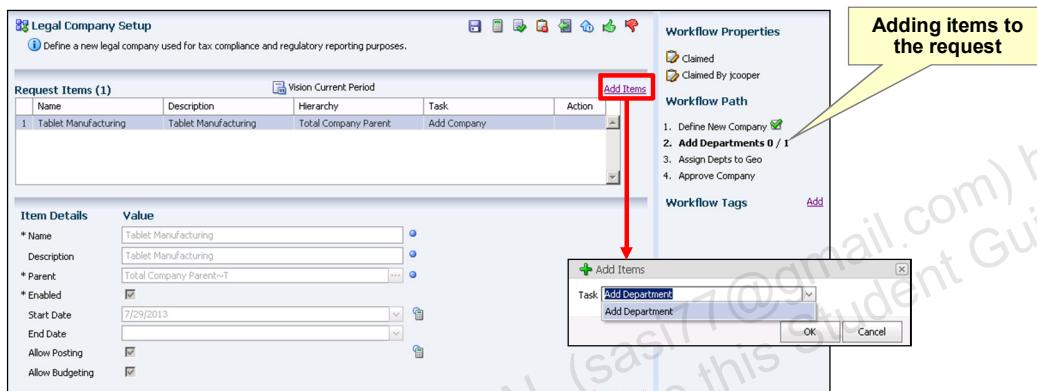
To claim assigned requests:

1. In the Worklist task group, select **Assigned**.
A list of all assigned requests is displayed.
2. Select a request and click the Open button ().
The request is displayed on a new tab.
3. In the Request Actions toolbar, click the Claim button ().
The request is claimed for approval and its status changed to Claimed. The Request Actions toolbar displays available actions.

Enriching Requests

Make changes to the request:

- Update the existing request items.
- Add new request items.
- Calculate request items.



ORACLE

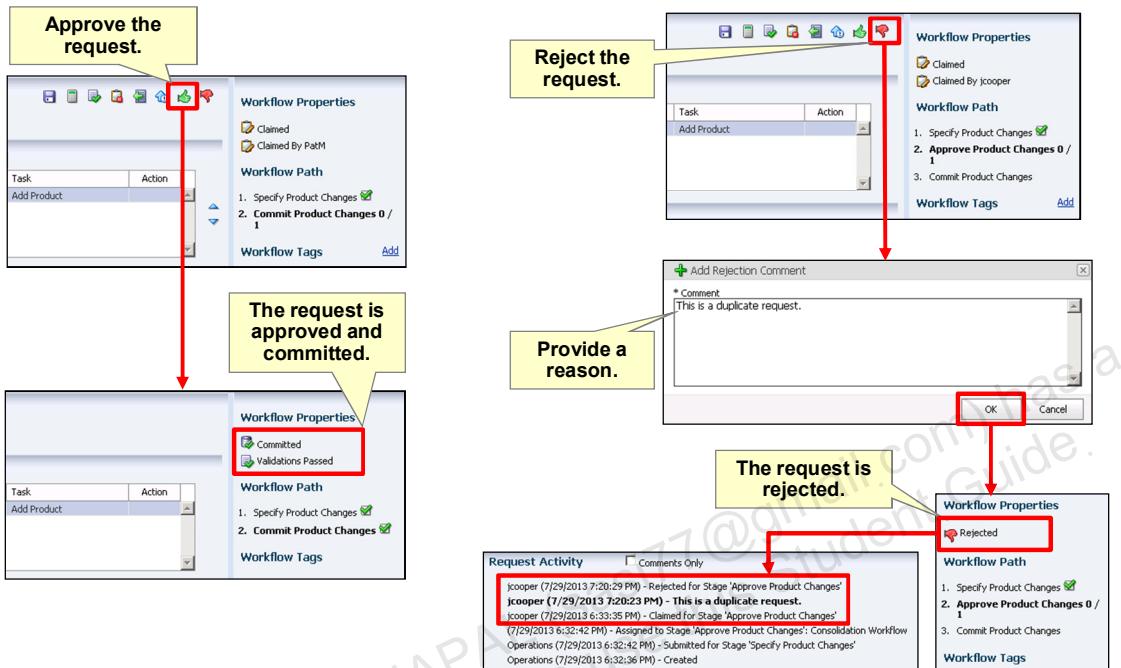
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Enrichment stages enable governance users to update request items that were added during submission, to add request items for making local changes to the same nodes in different hierarchies, or to make changes to entirely different nodes that are related.

Note: The request items cannot be removed during an Enrichment stage. They can only be deleted from the stage where they were originally added to the request.

An Enrichment stage can only have a single workflow task. If a request is claimed by a user for an Enrichment workflow stage, the request items can be edited based on the active workflow task assigned to the stage. If the active workflow task only permits the update of existing request items added by users in other workflow stages, the item details can be edited for all request items that are accessible based on the user's workflow access rights to the data. If the active workflow task for the stage performs any other type of data change (such as adding a new value or maintaining hierarchy relationship), new request items can be added to the request, but request items added in previous stages cannot be modified. After a request has been enriched, changes to the request should be saved and validated.

Approving and Rejecting Requests



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To approve claimed requests:

1. Open the request
2. Perform the assigned tasks (for example, you may be able to update request details).
3. Click the Approve button ().

The request is being validated. After successful validation, the request is moved to the next workflow stage. If the next workflow stage request is Commit, the data changes are applied to the target version and the request status is changed to Committed.

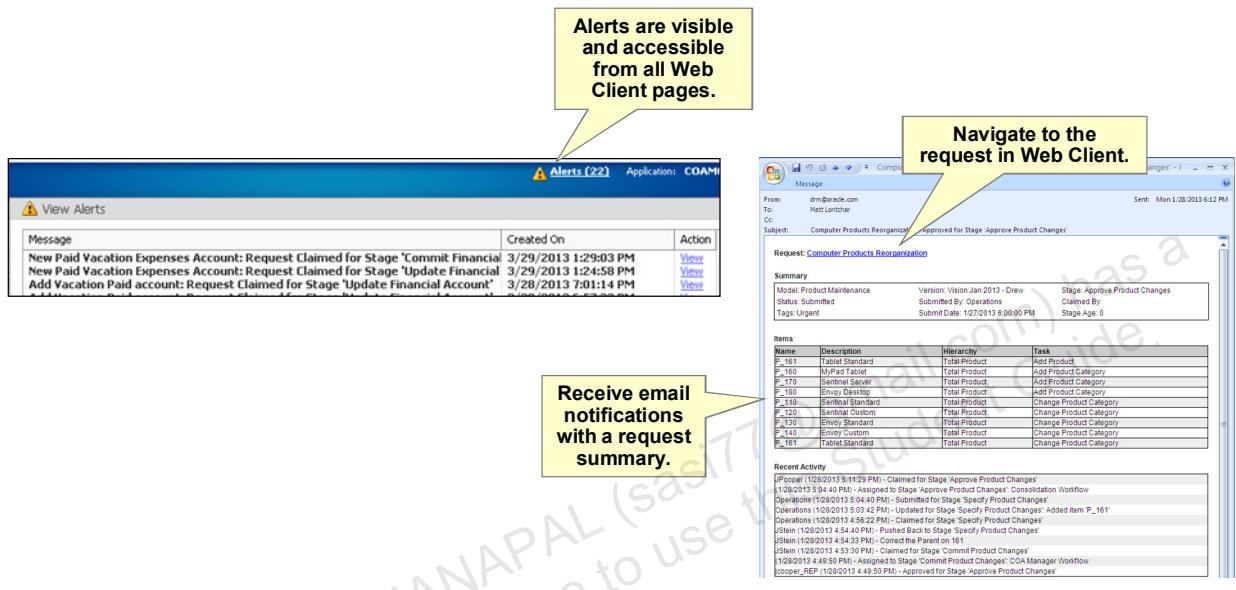
To reject claimed requests:

1. Open the request
2. Click the Reject button ().
3. Enter a comment to explain the rejection and click **OK**.

After a request is rejected, the workflow model is cancelled and users are no longer able to perform any action on the request. A rejected request may be later copied and submitted as a new request if applicable.

Alerts and Notifications

- Web Client alerts
- Email notifications



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Governance users and data managers can be made aware of request activities using alerts and notifications. The configuration of workflow model stages controls who is notified of activities taking place in a particular stage.

Users who are logged in to Data Relationship Management can be immediately alerted of request activity by using Web Client alerts. When a user receives an alert, an Alerts link appears at the top of any Web Client page. The Alerts link displays a count of alerts to be viewed. You can also configure email notifications for users who may not frequently access Data Relationship Management. In this case, an email is sent after a Web Client alert is raised.

Summary

In this lesson, you should have learned how to:

- Describe data relationship governance
- Create and manage workflow tasks and models
- Create, submit, open, claim, and approve requests
- Navigate the worklist
- Receive and view alerts and notifications

Quiz

Which of the following workflow objects define workflows in the Oracle Data Relationship Governance module?

- a. Workflow Models
- b. Workflow Tasks
- c. Workflow Requests
- d. Request Items
- e. All of the above



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: e

Quiz

Workflow tasks can be reused across multiple workflow models.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Identify the statements that correctly describe workflow requests.

- a. Requests represent a collection of changes for a specific set of data.
- b. A request can be based on one or more workflow models to control which users participate in the request, when they participate, and their type of participation.
- c. A request may consist of one or more items that can be added during initial submission or later enrichment of the request.
- d. Governance users or data managers who are assigned to the initial Submit stage can create requests.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c, d

Analyzing Data Changes

15

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe auditing data
- Query and view transaction history
- Analyze data changes by viewing baseline versions, creating As-Of versions, and viewing change-tracking properties

Auditing Data

- You can identify and audit changes that users make to data and metadata.
- Some options are available only:
 - For certain product features
 - When specific settings are enabled
 - If data is saved to the Data Relationship Management repository



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Relationship Management provides several ways to identify and audit changes that users make to data and metadata. Some options are available only for certain product features, or when specific settings are used, or after data is saved to the Data Relationship Management repository.

Querying and Viewing Transaction History

Run Query button

Transaction history

Version	Hierarchy	Node	Action	Description	Timestamp	User Name	Leaf	Property	From Value	To Value	Action Level
TotalComputer			CopyVersion	Copied versi...	12/29/2010 ...	ADMIN	<input type="checkbox"/>				Logged
TotalComputer			OpenVersion	Opened Ver...	12/29/2010 ...	ADMIN	<input type="checkbox"/>				Logged
TotalComputer			SaveVersion	Saved Versi...	12/29/2010 ...	ADMIN	<input type="checkbox"/>				Logged
TotalComputer	Geography		UpdateHierP...	*[Val.CUSTOM...	12/29/2010 ...	ADMIN	<input type="checkbox"/>	Val.Custom...	None	RealTime	Core
TotalComputer			GlobalQuery	Ran Global P...	12/29/2010 ...	ADMIN	<input type="checkbox"/>				Logged
TotalComputer	Account	Measures	Query	Ran Local Pr...	12/29/2010 ...	ADMIN	<input type="checkbox"/>				Logged
TotalComputer	Account	Margin Percent	UpdateNode...	Property Es...	12/30/2010 ...	ADMIN	<input checked="" type="checkbox"/>	Essbase.For...		Margin % S...	Core
TotalComputer	Account	Profit Percent	UpdateNode...	Property Es...	12/30/2010 ...	ADMIN	<input checked="" type="checkbox"/>	Essbase.For...		Profit % Sales	Core
TotalComputer	Product		MoveHier...	Core.UserDi...	12/30/2010 ...	ADMIN	<input type="checkbox"/>	Core.UserDi...	Core.Abbre...		Core
TotalComputer	Product		MoveHier...	Core.UserDi...	12/30/2010 ...	ADMIN	<input type="checkbox"/>	Core.UserDi...	Core.Abbre...		Core
TotalComputer	Product		UpdateHierP...	[Core] User...	12/30/2010 ...	ADMIN	<input type="checkbox"/>	Core.UserDi...		Core.Abbrev	Core

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The transaction history records all changes to Data Relationship Management application data (versions, hierarchies, nodes, and properties) and metadata and all user actions. Each operation is recorded with a time stamp, the user name, the type of action performed, and other relevant information.

To view transaction history, you query transactions for a specific version, hierarchies, or nodes. You can filter the query results to a specific date and time range, transaction ID range, action, level, user, or property.

To query and view transaction history:

1. On the Home page, select the **Audit** task group.
2. In the Audit Type drop-down list, select **Transactions**.
3. Configure the **Source** tab:
 - a. Specify a version, hierarchies, or nodes for which to find transaction data.
 - b. Select whether to return system transactions, data transactions, or both.
4. **Optional:** Click and configure the **Filters** tab:
 - a. Specify action levels, transaction log action types, properties, and users.
 - b. Select **Include Child Nodes** to include child nodes for the nodes returned by the query.

- c. Select **Include Shared Nodes** to include shared nodes for the nodes returned by the query.
 - d. Select **From Date**, **To Date**, or both, and specify a date range.
 - e. Select **Filter to Current Session** to filter transactions to those performed during the current session.
 - f. Select **From Transaction**, **To Transaction**, or both, and specify a range of transaction IDs.
 - g. In the Max Records box, specify a limit for the number of transactions displayed.
5. Select and configure the **Columns** tab:
- a. Perform one or more of the following actions to specify the fields to be displayed as columns in the query results:
 - Select fields in the Available list, and click the Select button () to move them to the Selected list.
 - Click the Select All button () to move all fields in the Available list to the Selected list.
 - Select fields in the Selected list, and click the Remove button () to move them to the Available list.
 - Click the Remove All button () to move all fields in the Selected list to the Available list.
 - b. Click the Move Up button () and the Move Down button () to reorder the fields.
6. On the toolbar, click the Run Query button ().
The transaction history query results are displayed.

Viewing Transaction History Details

Expand the row to view transaction history details.

The screenshot shows a transaction history grid with columns: Version, Hierarchy, Node, Action, Description, Timestamp, User Name, Leaf, and Property. A row for 'TotalComputer' is selected and expanded. The expanded row shows tabs for Action, Object, and History. The History tab displays transaction details: Username (ADMIN), Timestamp (12/29/2010 8:45:54), Action (CopyVersion), Action Level (Logged), and Description (Copied version 'Accessories v1' to 'TotalComputer'). A yellow callout box points to the expanded row with the text 'Expand the row to view transaction history details.'

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

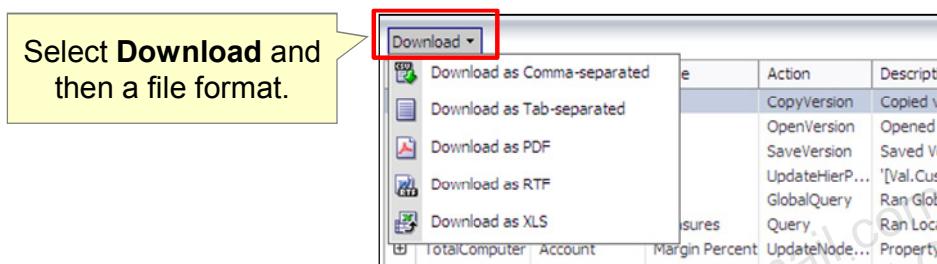
You can view details, such as transaction ID, version ID, hierarchy ID, and node ID for each transaction. If the transaction updated a property value, you can also view the value before and after the transaction. The slide shows an example of a transaction history detail that has the CopyVersion action. For metadata object changes, a side-by-side view enables you to compare the parameters of the object before and after the transaction.

To view history details for specific transactions:

1. In the transaction history grid, expand a row.
The Action, Object, and History tabs are displayed.
2. View transaction history details on the tabs.

Downloading Transaction History

You can download transaction history to a file in a variety of formats (comma-separated, tab-separated, PDF, RTF, or XLS).



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

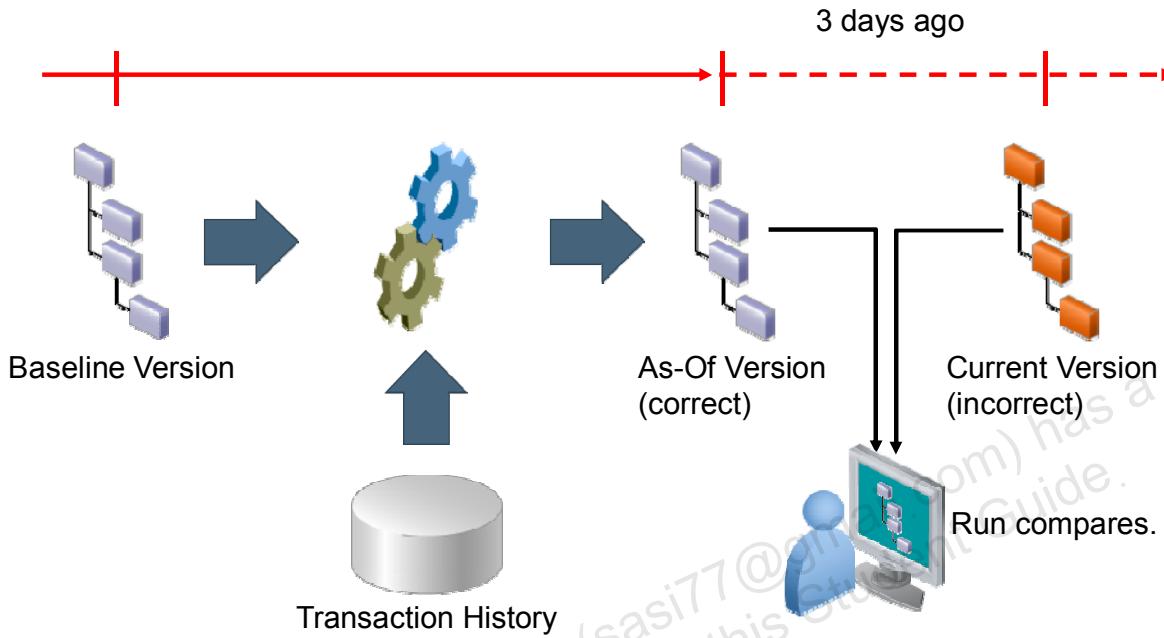
You can download transaction history to a file in the following formats:

- Comma-separated or tab-separated
- PDF, RTF, or XLS

To download transaction history:

1. Above the transaction history grid, in the Download drop-down list, select a file format option.
The File Download dialog box is displayed.
2. Click **Save**.
The Save As dialog box is displayed.
3. Browse to a location, enter a file name, and click **Save**.

Analyzing Changes Over Time



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Baseline and As-Of versions enable you to analyze changes during a certain period of time in the data management cycle. To identify changes, you can compare normal versions to baseline and As-Of versions or baseline and As-Of versions to each other. This approach is particularly useful when analyzing changes to shared hierarchy relationships or inherited and derived property values.

The slide shows a baseline, As-Of, and current version. The As-Of version is a snapshot of the data from three days ago. You run a compare to analyze the differences between the As-Of version (which you know to be correct) and the current version (which may be incorrect). You view the transactions between the baseline and As-Of versions.

Viewing Baseline Versions

The baseline version is an initial, read-only snapshot of the version where you start the data management cycle.

A screenshot of the Oracle Data Relationship Management application. The main window shows a 'Versions' table with the following data:

Name	Description	Type	Status	Saved	Load Status
Accessories	Computer Accessories	Normal	Working	<input checked="" type="checkbox"/>	Initialized
Copy Of Accessories	Normal	Working	<input checked="" type="checkbox"/>	Initialized	
HardwareSoftware	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Initialized
TC	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Initialized
TotalComputer	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Loaded
TotalComputer - Baseline	Computer Hardware and Software Version	Baseline	Expired	<input checked="" type="checkbox"/>	Loaded

An annotation with a yellow callout points to the 'TotalComputer' row, specifically the 'Baseline' column, with the text: 'Expand a version to view its baseline.'

Below the table, the 'Hierarchies' tab is selected, showing a list of hierarchies:

Name	Description
Account	
Activity	
Geography	
Product	

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you create a new version or copy a version, Data Relationship Management creates two versions:

- The **baseline version** is an initial, read-only snapshot of the version where you start the data management cycle. Changes that you make to the data do not apply to the baseline version.
- Note:** A version must be saved to the repository and the AllowAsOf system preference set to True to enable baseline versions.
- The **normal version** is a working version where you make data changes.

To view baseline versions:

1. In the version list in the Browse task group, expand a normal version.
The baseline version is displayed.
2. Select the baseline version.
Hierarchies in the baseline version are listed on the Hierarchies tab.

Creating As-Of Versions

An As-Of version is a snapshot of a normal version at a given point in time.

The screenshot illustrates the process of creating an As-Of version. At the top, a dialog box titled "Create As-Of Version" is displayed, showing the configuration for a new version type. The "Version Type" is set to "Create As-Of" and the "As-Of Type" is set to "Transaction Date/Time". The "Transaction Date" is set to "7/1/2013" and the "Transaction Time" is set to "12:00:00 AM". A red arrow points from this dialog down to the "Versions" view below, which lists various versions of the "TotalComputer" node. The newly created "As-Of" version, named "TotalComputer - As-Of: 2013-07-01 00:00:00 (Pacific Daylight Time)", is highlighted in blue and has its status set to "Expired".

Name	Description	Type	Status	Saved	Load Status
Accessories	Computer Accessories	Normal	Working	<input checked="" type="checkbox"/>	Initialized
Copy Of Accessories	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Initialized
HardwareSoftware	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Initialized
TC	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Initialized
TotalComputer	Computer Hardware and Software Version	Normal	Working	<input checked="" type="checkbox"/>	Loaded
TotalComputer - As-Of: 2013-07-01 00:00:00 (Pacific Daylight Time)		As-Of	Expired	<input checked="" type="checkbox"/>	Loaded
TotalComputer - Baseline	Computer Hardware and Software Version	Baseline	Expired	<input checked="" type="checkbox"/>	Loaded

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

It is possible to roll the baseline version forward to any point in time to generate a snapshot at that point. This snapshot is called an As-Of version. You cannot modify the hierarchies or node properties of As-Of versions, but you can display their hierarchies, run compares, and export data.

For example, if you discover that the current version contains incorrect information and you want to investigate what changes to the data caused the problem, you can create an As-Of version of some past date and time when the version contained correct information, and then run compare tools to analyze the differences between the As-Of version and the current version. This process is usually less difficult than analyzing the transaction history for the same period of time.

The slide shows an example of an As-Of version that was created for the TotalComputer version. The As-Of version is configured to be a snapshot of the TotalComputer version as it was on July 1, 2013. The resulting version is named TotalComputer - As Of: [date/time], and its status is set to Expired.

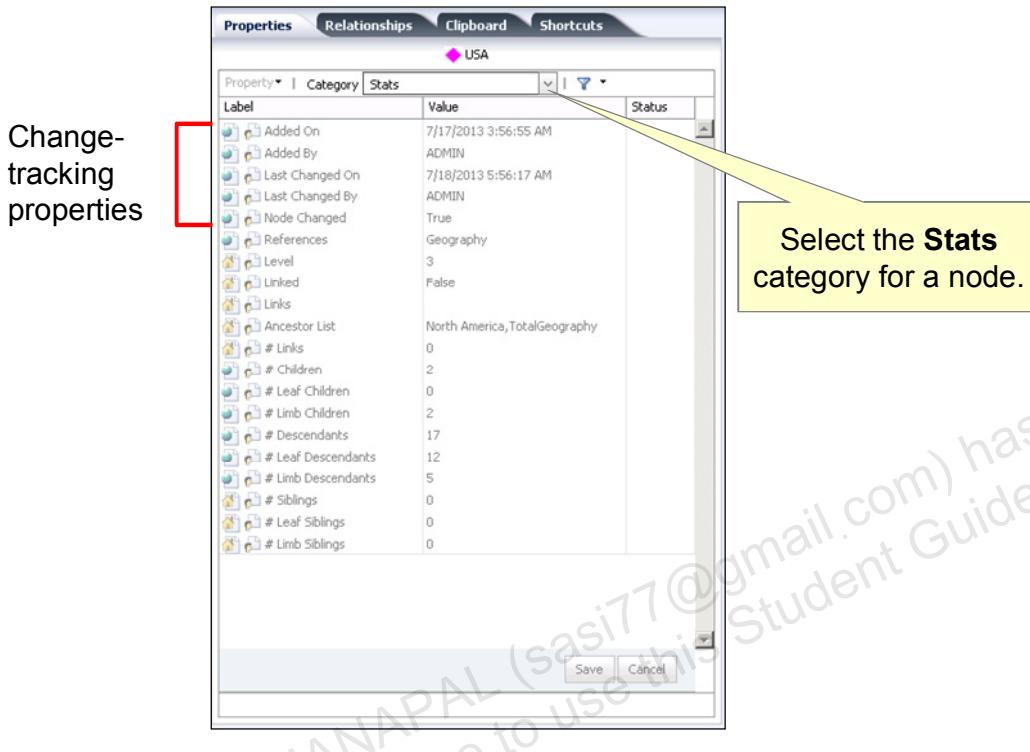
Note: You cannot save As-Of versions because they are intended for temporary analyses.

To create As-Of versions:

1. In the Browse task group, right-click a version, and select **Create As-Of Version**.
The Create As-Of Version dialog box is displayed.
2. In the As-Of Type drop-down list, select one of the following options:
 - Select **Transaction Date/Time** to specify a date and time as of which you want to create the version.
 - Select **Transaction Date Offset (Days)** to specify the number of offset days and an offset time.
 - Select **Transaction ID** to specify a transaction ID.
3. Click **OK**.
The As-Of version is displayed.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Viewing Change-Tracking Properties



Select the **Stats** category for a node.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Change-tracking properties enable you to analyze changes to individual nodes. These properties are updated each time a transaction is processed for a node in a hierarchy. This information is managed only by the system; you cannot modify it. You can use the change-tracking properties with the query, compare, and export features to identify bulk changes to a set of nodes.

The following change-tracking properties for nodes enable you to monitor node changes:

- **Added On** is the date and time when a node is created.
- **Added By** is the user name when a node is created.
- **Last Changed On** is the date and time when a node is modified.
- **Last Changed By** is the user name when a node is modified.
- **Node Changed** is a value of True or False to indicate whether a node was modified since the version was created.

To view change-tracking properties:

1. Open a hierarchy.
2. On the Properties tab, in the Category list, select **Stats**.
3. View the first five properties listed, which are the change-tracking properties.

Summary

In this lesson, you should have learned how to:

- Describe auditing data
- Query and view transaction history
- Analyze data changes by viewing baseline versions, creating As-Of versions, and viewing change-tracking properties



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

Which of the following statements about baseline versions are true?

- a. Baseline versions are displayed in the Audit task group.
- b. Changes that you make to a saved version do not apply to baseline versions.
- c. The AllowAsOf system preference must be set to True to enable baseline versions.
- d. Baseline versions are read-only.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b, c, d

Quiz

Change-tracking properties for nodes enable you to monitor which of the following node changes?

- a. The date and time when a node is created
- b. The user name when a hierarchy is created
- c. The date and time when a node is modified
- d. Whether a node was modified since the version was created



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c, d

Quiz

Transaction history records which of the following changes?

- a. Version changes
- b. Hierarchy, node, and property changes
- c. Change actions for metadata
- d. Log-in and log-out actions



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Migrating Metadata

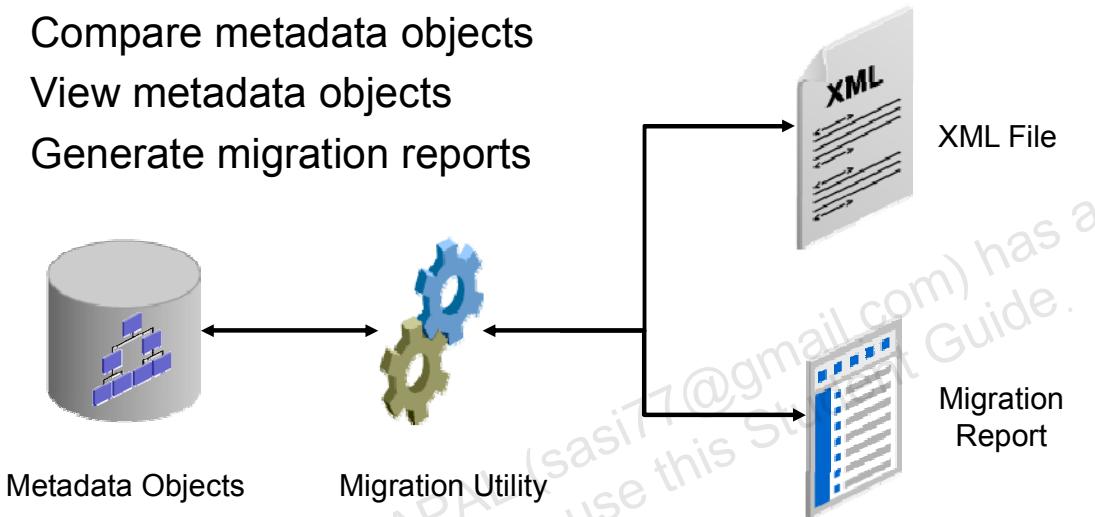
ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Migrating Metadata

With the Migration Utility, you can:

- Extract metadata objects
- Load metadata objects
- Compare metadata objects
- View metadata objects
- Generate migration reports



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

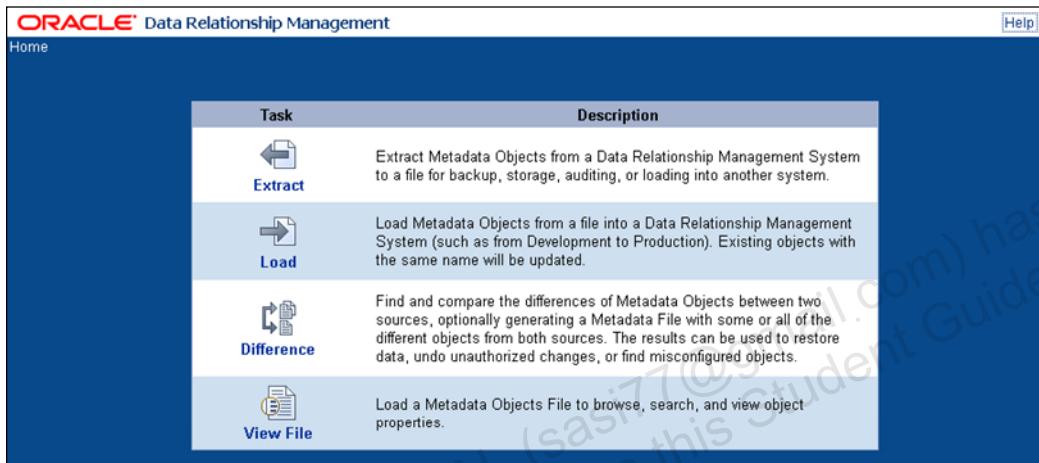
The Data Relationship Management Migration Utility provides the ability to move metadata object types between Data Relationship Management applications.

With the Migration Utility, you can:

- Extract metadata object types from a Data Relationship Management application to an XML file and generate an HTML report from the results
- Load metadata from an XML file into a Data Relationship Management application
- Compare metadata differences between two sources, create an XML file with the differences, and generate an HTML report from the results
- View metadata in an XML file and generate an HTML report from the file

Launching the Migration Utility

- Start > All Programs > Oracle EPM System > Data Relationship Management > Migration Utility
- <http://localhost/drm-migration-client/>



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, the Migration Utility is installed in the following directory:

`MIDDLEWARE_HOME\EPMSystem11R1\products\
DataRelationshipManagement\client`

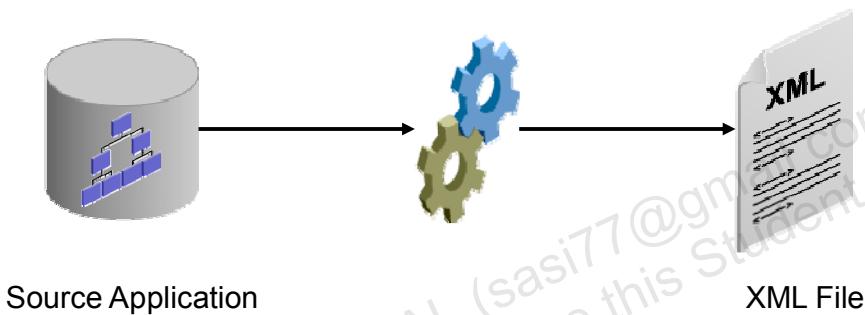
You can launch the Migration Utility by navigating to the installation directory and double-clicking the Data Relationship Management Migration Utility shortcut. Alternatively, you can enter the following URL into your browser:

`http://hostname/drm-migration-client`

In the URL, `hostname` is the hostname of the machine where the Data Relationship Management Server is running.

Extracting Metadata

- Use the Migration Utility to extract metadata objects to a file for audit and restoration purposes.
- Perform extracts before or after you make changes.
- Add header information to the extract file to describe the purpose and use of the file.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Changes to metadata made by application administrators, access managers, and data managers are applied immediately to a Data Relationship Management application and are not versioned by the system. You can use the Migration Utility to extract metadata objects to a file for audit and restoration purposes.

Consider the following:

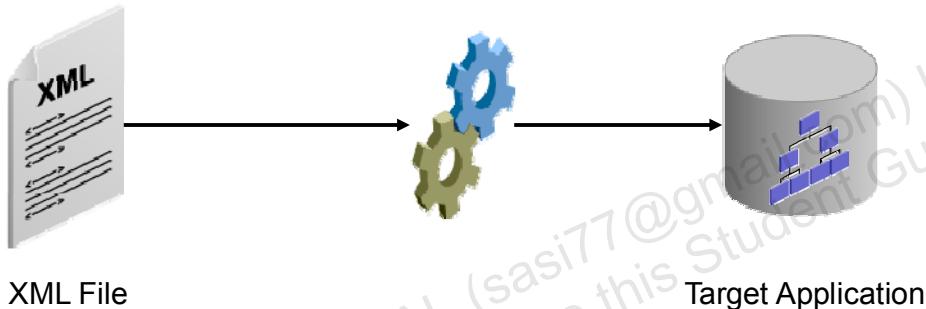
- You can perform extracts before or after you make changes.
- You can add header information to the extract file to describe the purpose and usage of the file.
- You can create a migration report for an extract file from the Migration Utility. The migration report displays the details of the extract file in an HTML format for easy readability.
- You can load the extract file into a Data Relationship Management application to restore or migrate the metadata objects in their original state when the extract file was created.

To extract metadata with the Migration Utility:

1. Launch the Migration Utility.
The main menu is displayed.
2. Select **Extract**.
The Login Connection window is displayed.
3. Configure the connection information, and click **Log In**.
The Select Metadata Objects window is displayed.
4. In the hierarchy tree, click the plus sign (+) to view objects.
5. **Optional:** Select an object name to display the object type definition in a new window.
6. Perform one of the following actions to specify the object types or objects to extract:
 - Select the check box for an object type to select the object type and all of its objects.
 - Select the check box for specific objects that you want to extract.
7. Click **Next**, and review the summary information.
 - The Migration Utility performs additional checks for object types that have dependencies. For example, an export may depend on property definitions or a property definition may reference another property definition. If dependencies are missing in the summary, you may select specific dependencies to include. You can include all excluded dependencies or exclude all dependencies.
 - Increasing the page size enables you to define the number of object types to view on a page.
8. **Optional:** Enter metadata details for this extract. You can enter the following information:
 - **Title:** Maximum of 255 characters
 - **Purpose:** Formatted memo
 - **Usage:** Formatted memo
 - **Application Version:** Maximum of 20 characters
 - **File Version:** Maximum of 20 characters
9. Click **Run Extract**.
The extract file is generated.
10. Perform one or more of the following actions:
 - Click **Download the Metadata File** to open or save the XML file.
 - Click **View the Metadata File** to view the XML file details.
 - Click **Load the Metadata File** to load the XML file into a Data Relationship Management application.
 - Click **Generate Reports for the Metadata File** to generate a report from the XML file.

Loading Metadata

- Use the Migration Utility to load metadata objects to a target Data Relationship Management application.
- Perform extracts or a database backup before loading metadata.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Only files with the Data Relationship Management XML format can be loaded into a Data Relationship Management application. A log file is created after a load is performed and displays the following severities of data: audit, information, warning, and error message.

To load metadata into Data Relationship Management applications:

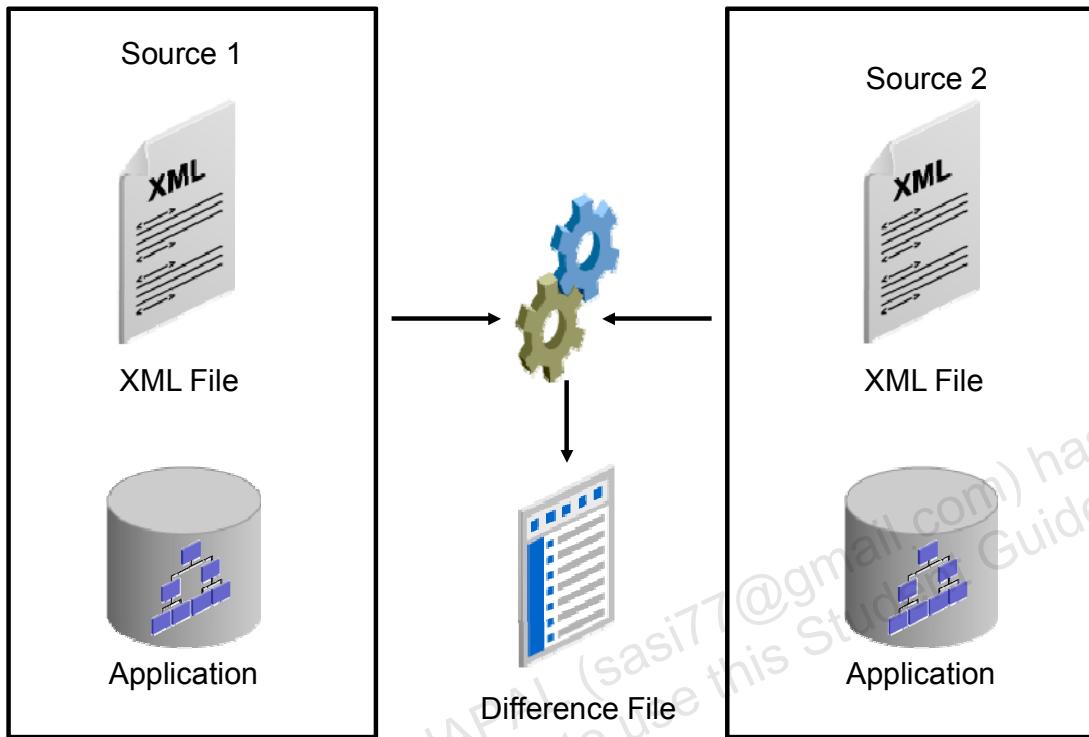
1. Launch the Migration Utility.
The main menu is displayed.
2. Click **Load**.
3. Click **Browse**, select the XML file that you want to load, and click **Upload**.
4. Review the uploaded file information and click **Next**.
5. Enter Data Relationship Management connection information and click **Log In**.
6. Select the object types or objects to load and click Next.

Note: Click the plus sign in the hierarchy tree to see objects. Select the check box for an object type to select the object type and all of its objects, or select the check box for the objects that you want to load. Click an object name to display the object type definition in a new window.

7. Review the summary information and click **Next**.
Note: Page size enables you to define the number of object types to view on a page.
8. **Optional:** Select **Continue Load After Error** for the load to continue even if errors are encountered.
9. Click **Run Load**.
10. Review the load results. You can change the view of the log file by selecting the severity of detail to display: audit, information, warning, and error. To save the log file, click **Download**.
Note: The log items can be sorted by any column using the column header links.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.

Comparing Metadata



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can compare two metadata sources. For example, you can compare metadata differences between two Data Relationship Management applications, between two XML files, or between a Data Relationship Management application and an XML file. You can also generate an XML file containing the differences between the two metadata sources. The results can be used to restore data, undo unauthorized changes, or find wrong object type configurations.

Viewing Metadata and Generating Reports

The screenshot shows two main windows from the Oracle Data Relationship Management application:

- Main Window (Top):** Titled "ORACLE® Data Relationship Management". It displays a tree view of metadata objects under "drm_ProjectOne_20130702115309.xml". Nodes include "Node Access Groups", "Workflow Models", and "Workflow Tasks". A red box highlights the "Reports" button in the top right corner of the window header. A red arrow points from this button to the "View Report" link in the "Select Report Details" dialog.
- Select Report Details Dialog (Top Right):** Titled "Select Report Details". It shows a dropdown for "Report Template" set to "Basic HTML". A red box highlights the "View Report" link. Another red arrow points from this link to the "View Report" link in the "Migration Report" window below.
- Migration Report Window (Bottom):** Titled "ORACLE® Data Relationship Management Migration Report". It contains sections for "ProjectOne Workflow Metadata" and "Purpose". The "Purpose" section states: "To extract workflow objects of the ProjectOne application for migration purposes." A red arrow points from the "View Report" link in the "Select Report Details" dialog to the "View Report" link in this window.

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view metadata objects in XML files:

1. On the Main Menu, click **View File**.
2. Click **Browse**, select the XML file that you want to view, and click **Upload**.
3. Review the uploaded file information and click **Next**.
4. Click the plus signs in the hierarchy tree to view metadata objects.
5. **Optional:** Click **Find** to search for an item in the file.

To generate the migration report after viewing the metadata file:

1. Click **Reports**.
2. Do one of the following:
 - Click **View Report** to display the report.
 - Click **Download Report** to save the report.

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

SASIKUMAR DHANAPAL (sasi77@gmail.com) has a
non-transferable license to use this Student Guide.