Oracle® Hyperion Disclosure Management XBRL Taxonomy Designer

Help

RELEASE 11.1.2.1.00

ORACLE

ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

XBRL Taxonomy Designer Help, 11.1.2.1.00

Authors: EPM Information Development Team

Contents

1 About XBRL Taxonomy Designer

In This Chapter

About XBRL Taxonomy Designer Help

Oracle Hyperion Disclosure Management XBRL Taxonomy Designer Online Help is an interactive help system that answers questions about XBRL Taxonomy Designer, helps troubleshoot problems, and provides general tips and advice.

To get started with XBRL Taxonomy Designer and XBRL, Oracle recommends working through the tutorials in Appendix A, "Tutorials."

Using XBRL Taxonomy Designer Help, you can:

- Search using the Contents view in the left pane
- Search the Index
- Find information using key word queries

XBRL Taxonomy Designer Features

XBRL Taxonomy Designer is the industry's most full-featured taxonomy and instance creator.

XBRL Taxonomy Designer includes a suite of robust creation, editing, mapping, validation, analysis and reporting tools, for managing complex taxonomies, with single and in-reference taxonomy usage patterns and validation capabilities.

The XBRL Taxonomy Designer provides the fundamental and comprehensive functionality required to create, exchange, consume, and manage large numbers of XBRL instance documents and taxonomies. Core features include:

- Support for XBRL 2.1 and XBRL legacy versions (2.0, 2.0a, and 1.0)

- A fully-conforming XBRL processor, as defined by the XBRL 2.1 specification, passing the 275 tests within the XBRL International Specification 2.1 conformance suite
- Support for Financial Reporting Taxonomies Architecture 1.0 (FRTA Candidate Recommendation of 2003-12-31) best practices for building financial reporting taxonomies, and any taxonomy that follows this architecture recommendation
- Creating and editing sophisticated taxonomies, such as the International Financial Reporting Standards (IFRS) and United States Financial Reporting (USFR) frameworks
- Integrated instance document creation, taxonomy creation and extension, and validation environments, enabling the creation and management of robust extension taxonomies and leveraging the full power of XBRL extensibility
- Creating, editing, and reading XBRL instance documents and taxonomies
- Editing taxonomies and instance documents simultaneously
- Validating XML Schema 2.1, XBRL, and best practices to ensure the documents and data you read are valid XBRL, follow the business rules of your organization or domain, and comply with XBRL best practices
- Creating and executing business rules (including semantic validation) to ensure the data you are creating or using are valid.
- Robust mapping, import, and export functionality for migrating source data into XBRL, and then back into your applications' data format
- Productivity features and tools to navigate through, and effectively manage and edit large element taxonomies of 4000 or more elements
- GUI and API interfaces that enable developers and semi-technical users comfortable building Microsoft -Excel-like macros using VBA to leverage XBRL Taxonomy Designer functionality directly from within their applications

System Requirements

- Microsoft Windows Server System 2003
- Microsoft Windows 2000 Professional, Server Edition with Service Pack 4
- Microsoft Windows XP Home or Professional Edition with Service Pack 2
- 128 MB of RAM (256 MB of RAM recommended)
- Microsoft™ .Net Framework Version 1.1 with the latest .NET updates
- Microsoft™ Visual J# .NET Version 1.1 Redistributable Package with the latest .NET updates
- Microsoft™ Internet Explorer 6.0, 6.0 with Service Pack 2 or later release
- Windows 2000 and later
- Internet Explorer 7 or later

Total download package 44 MB:

- 9 MB - XBRL Taxonomy Designer

- 35 MB - Microsoft required components

To use the Import/Export features with Microsoft Excel, Word, Access, and SQL Server, install and register the software listed below:

- Microsoft Excel 2002 or later, Word 2000 or later, Access 2000 or later, or OLE DB
- Provider Databases if needed (for example, SQL Server, Oracle, MySQL)
- Internet Access (for downloading software and access to online help documentation and tutorials.)
- CD-ROM or DVD-ROM drive for CD installation

Required Components

These components must be installed to use XBRL Taxonomy Designer. Failure to do so may interrupt the installation process and prompt you to install these components before continuing.

These English language required components are supplied on the installation CD and download package:

- Microsoft.NET Framework Version 1.1 with latest .NET updates
- Microsoft Visual J#.NET Version 1.1 Redistributable Package
- Microsoft Internet Explorer 6 or later release

These required components are not supplied on the installation CD:

- Microsoft Office 2003/2002 with Word, Excel, and Access required for import/export of taxonomy and/or instance data.

  Note: Microsoft MDAC 2.8 can be installed in place of Microsoft Access.

- An OLE DB Provider (for example, SQL Server, Oracle) must be installed and configured for database imports/exports

Troubleshooting FAQ's

1. **Q:** When I export, XBRL Taxonomy Designer displays an exception or does not respond.

   A: Ensure that the required Microsoft Office product is installed, and registered. If the problem persists, contact: support@oracle.com.

2. **Q:** Occasionally, XBRL Taxonomy Designer does not respond when I use the "Export - tree view" option.

   A: If the product does not respond after using this option, Oracle recommends that you use the other export options to export relationship trees.

3. **Q:** When I launch XBRL Taxonomy Designer, an exception error is displayed. What do I do?

A: An error can occur if the required components are installed, but the machine has not been rebooted yet. Reboot the computer and launch the application. If the problem persists, contact: support@oracle.com.

4. **Q:** Where can I find the product version and creation date?

   A: The product version and creation date is displayed on the XBRL Taxonomy Designer startup screen. Also, you can display the product version and date by selecting About Disclosure Management XBRL Taxonomy Designer from the Help menu.

XBRL Taxonomy Designer Online Tutorials

The tutorials familiarize you with XBRL Taxonomy Designer and the basics of XBRL. Oracle recommends that you work through these lessons in the order presented. Lessons include practical exercises that alter the test data files. Each successive lesson includes exercises that build on the previous exercises. The files referenced in the exercises are included with your XBRL Taxonomy Designer installation.

2 Using XBRL Taxonomy Designer

In This Chapter

Using Keyboard Equivalents

Table 1 lists the general XBRL Taxonomy Designer keyboard equivalents.

Table 1 Shortcut Keys

Keyboard Equivalents	Description
F1	Opens the XBRL Taxonomy Designer Online Help menu
Ctrl+Z	Undoes the last XBRL Taxonomy Designer action
Ctrl+I	Inserts a sibling element
Ctrl+N	Inserts a child element
Ctrl+L	Opens editing of the selected cell
Ctrl+X	Cuts the selected element and its relationships
Ctrl+D	Deletes the selected element and its relationships

Keyboard Equivalents	Description
Ctrl+P	Opens the print dialog box
Ctrl+S	Saves the document
Ctrl+O	Displays the Options menu
Alt+F4	Exits XBRL Taxonomy Designer

Moving Elements

There are several operations that can occur when moving elements within the relationship views.

For each drag and drop operation, color highlighting indicates whether the operation is an allowed edit. Any relationship with the current focus is displayed in blue.

See "Using the Relations Collection Editor" on page 64 and "Creating a Business Rule Using the Properties Grid" on page 232.

Figure 1 Insert Concept into a Tree

(The new location for the inserted element is displayed in purple.)

Figure 2 Insert Concept Above

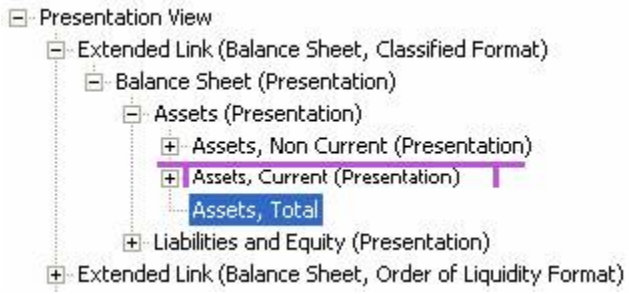
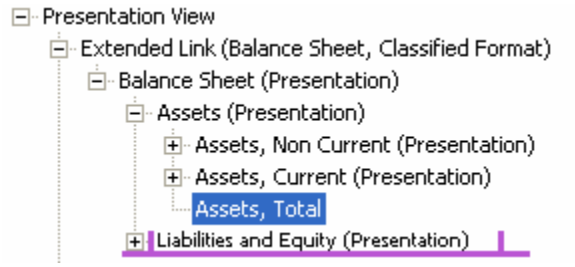
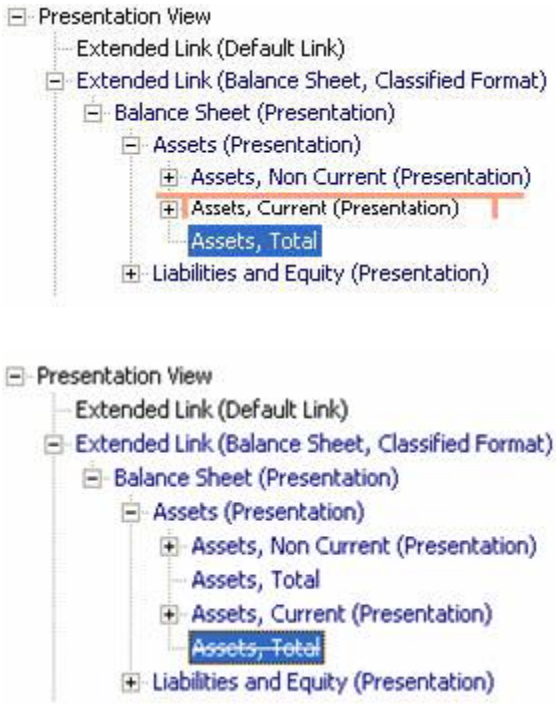


Figure 3 Insert Concept Below



When working with reference taxonomies, if the references are set to read-only, then the color of the selected element reflects that the action results in a prohibit or overriding arc operation.

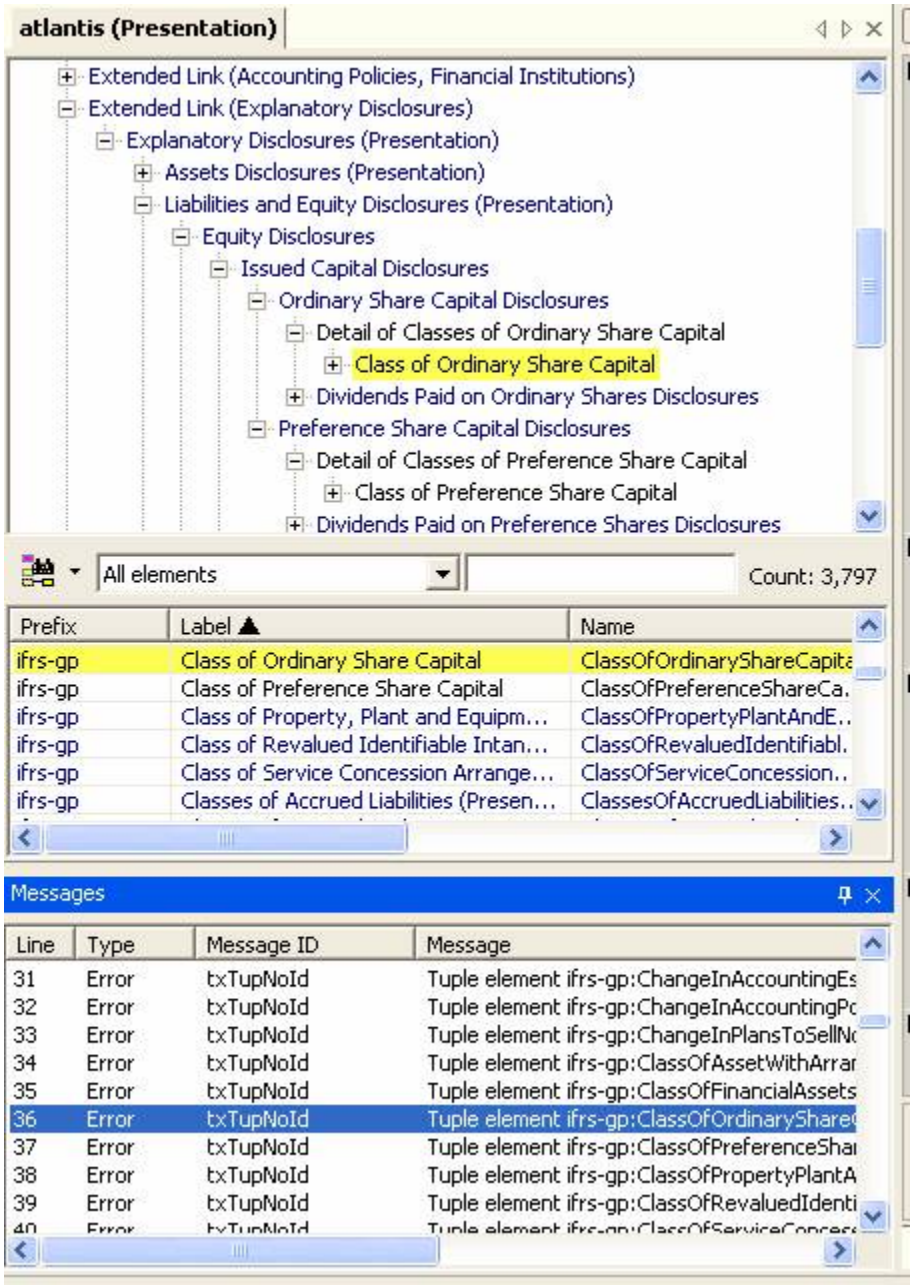
Figure 4 Prohibit and/or Override Relationship Arcs



(The color orange indicates an operation that results in a prohibited relationship. The result of dragging to create a prohibit is that the original arc is prohibited, and an arc is created where the relationship is dropped. Strikethrough indicates an arc which has a prohibit.)

Understanding Errors and Warnings

Whenever an error occurs during validation, you can double-click the error and the application highlights (using the color yellow) the offending concept and/or relationship causing the error.



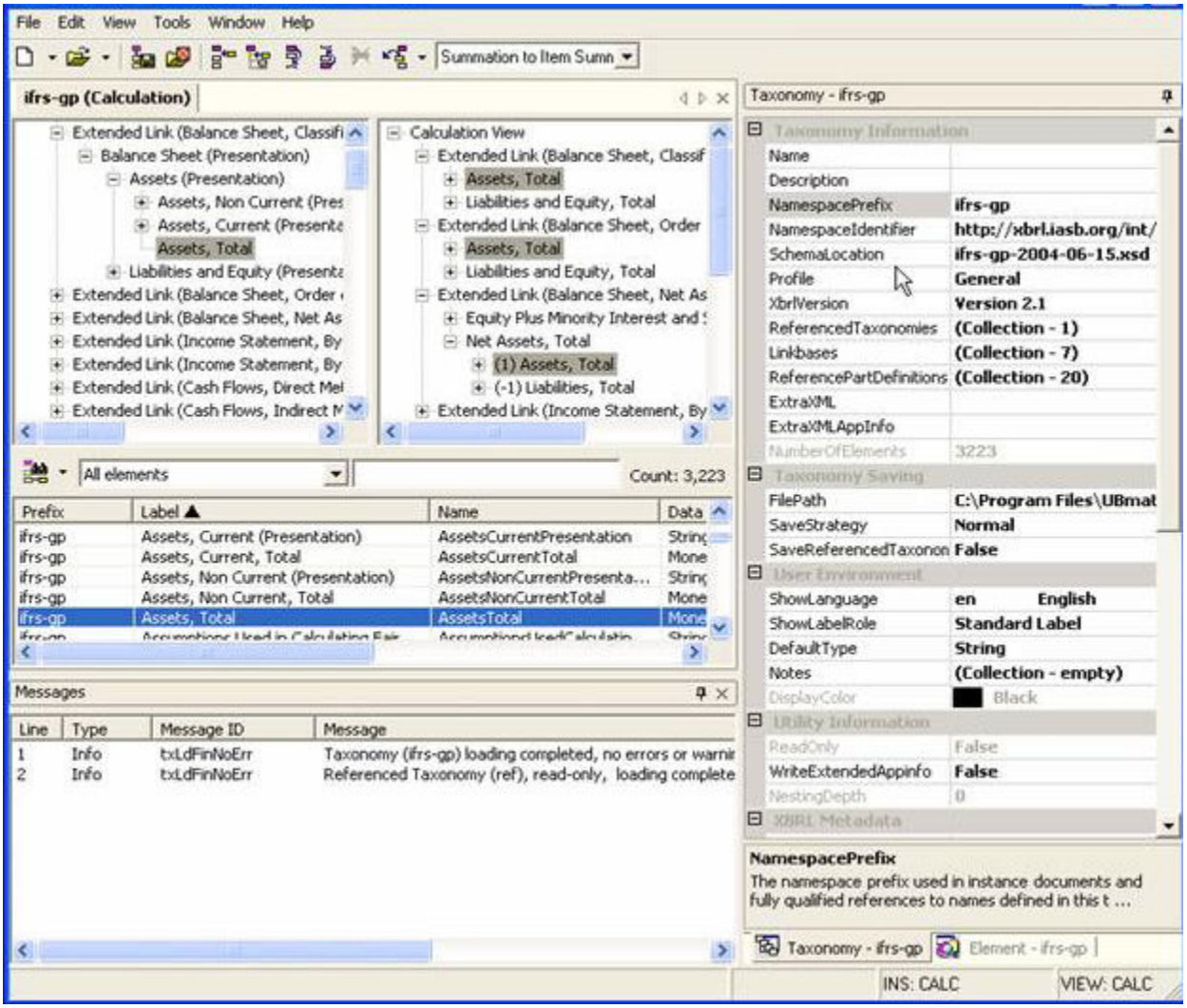
Synchronizing Views

The Sync-to-View feature locates instances of a concept or relationship in the relationship views.

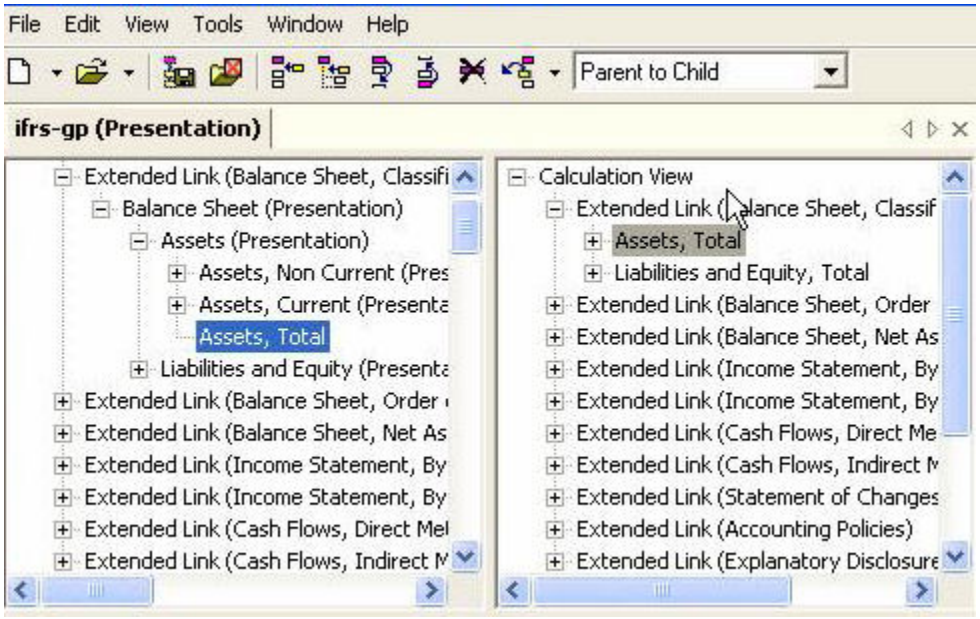
Use Sync to Element List to select concepts from the element list view. The selected concepts are highlighted in the tree view.

If you set Sync to Other Panes in relationship views, selecting a concept in one of these views selects that concept in all currently open tree views with this option set.

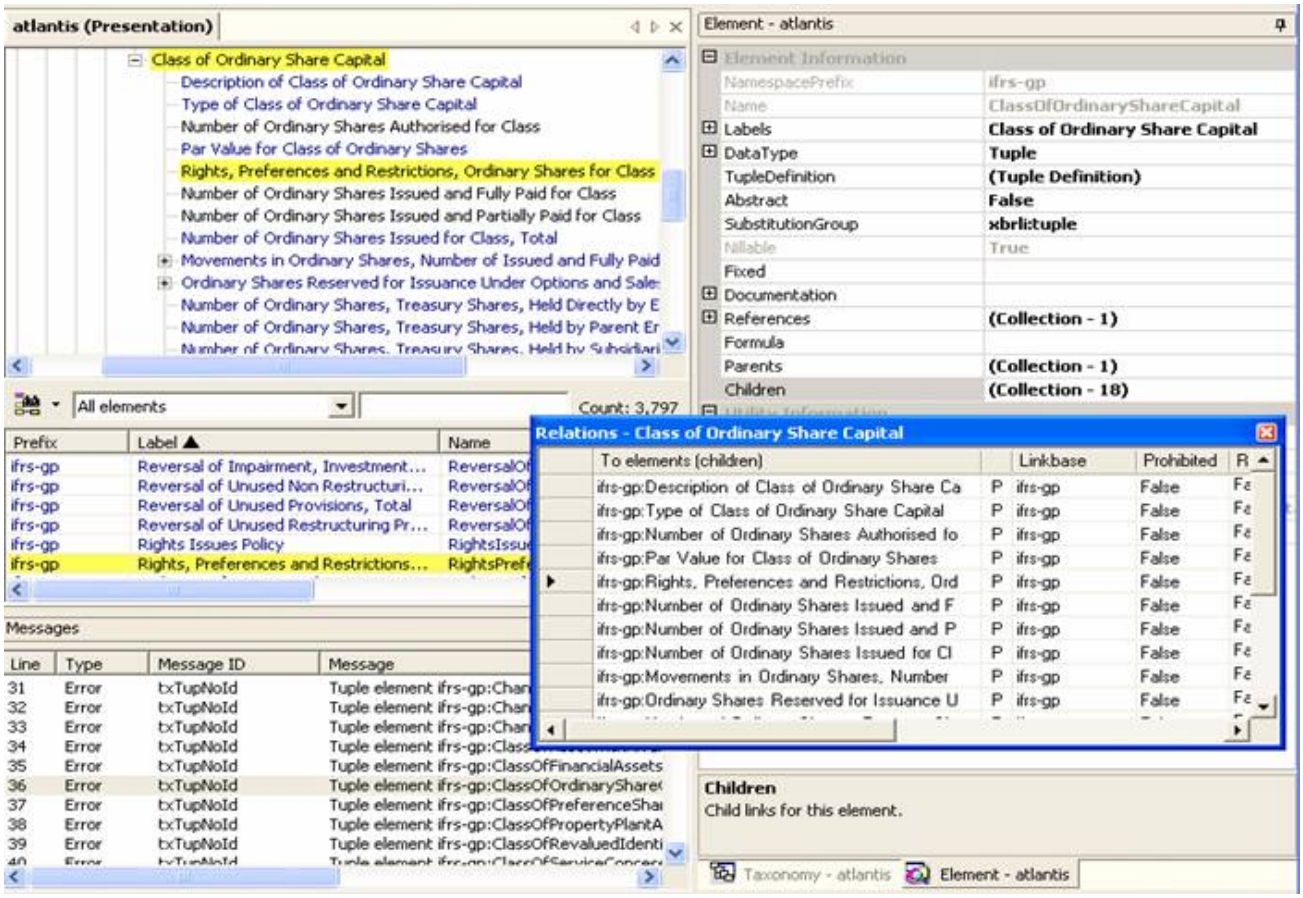
When you select an element in the list view (select **View**, then **Sync to Element List**) the concept is highlighted in grey for every instance in the tree views, as shown below.



When you select a relationship in the tree view (select **View**, then **Show**, and then **Sync to Other Panes**), the concept is selected in the other synchronized relationship view panes currently open within the same extended link. The links highlighted in the other panes are displayed in gray.



You can also synchronize a property from the collection editor with the tree and element list view. Selecting the property highlights it in yellow. In the example below, a relationship between the parent and a selected child is highlighted in yellow.



Searching for Elements

The Elements List view has several filter and search options to help locate elements.

Click the drop-down arrow associated with the Find and Show Elements icon to display the list of choices. You can choose criteria to Filter the dictionary view of the taxonomy elements. The Search choice examines element labels or names within the current selection of shown language and label role (as selected on the taxonomy property grid).

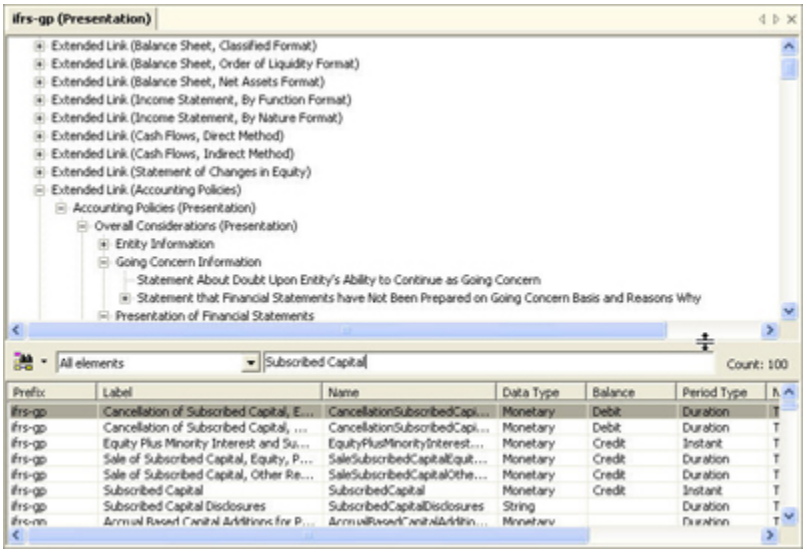
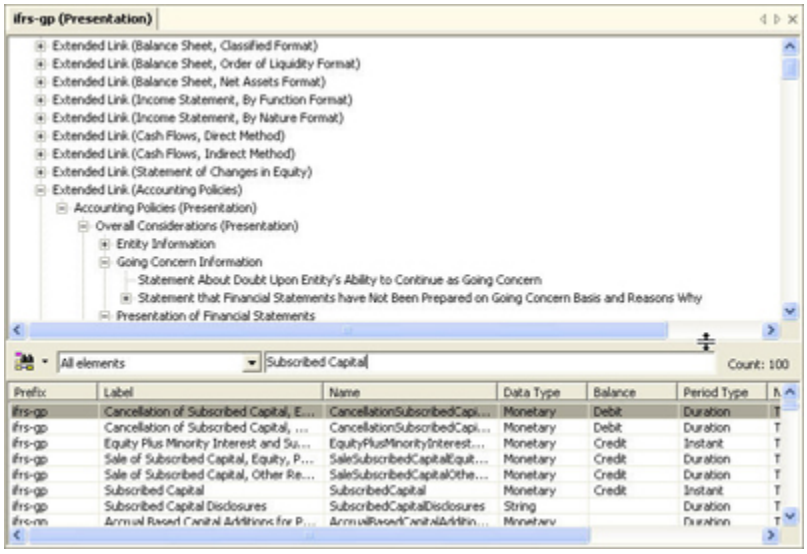


Table 2 Find and Show Options

Option	Description
Filter	Filters the element list based on the selected criteria. It shows only the elements that meet the criteria. Search can then be applied to the filtered results.
Search	Searches incrementally through the element list to find the selected criteria. It shows all elements in the list. Each click of the Search button moves the cursor to the next element that meets the criteria.
Any part of field	Selects an element if it contains the specified text in any part of its label. For example, specifying “Cash” selects any element with “Cash” somewhere in the label, including “Cash”, “Cash and Cash Equivalents”, or “Cash and Marketable Securities”.
Start of Field	Selects an element if it contains specified text at the beginning of the label. For example, specifying “Cash” selects any element beginning with the word “Cash”, including “Cash”, “Cash and Cash Equivalents”, or “Cash and Marketable Securities”. Note that using this option, “Balance of Cash” is not selected.
Compare Labels	Determines whether the label field participates in the action currently selected (language and label role).
Compare Names	Determines whether the name field participates in the action.
Fuzzy match	Attempts to match terms that are similar to the text specified, or follow a specified pattern.

Searching Using Criteria



Search criteria are entered into the search field. The count displays the count of all taxonomy concepts or the count of the filtered list of concepts. XBRL Taxonomy Designer supports these search codes designed to assist with large queries (see Table 3 on page 20).

Table 3 Supported Search Codes

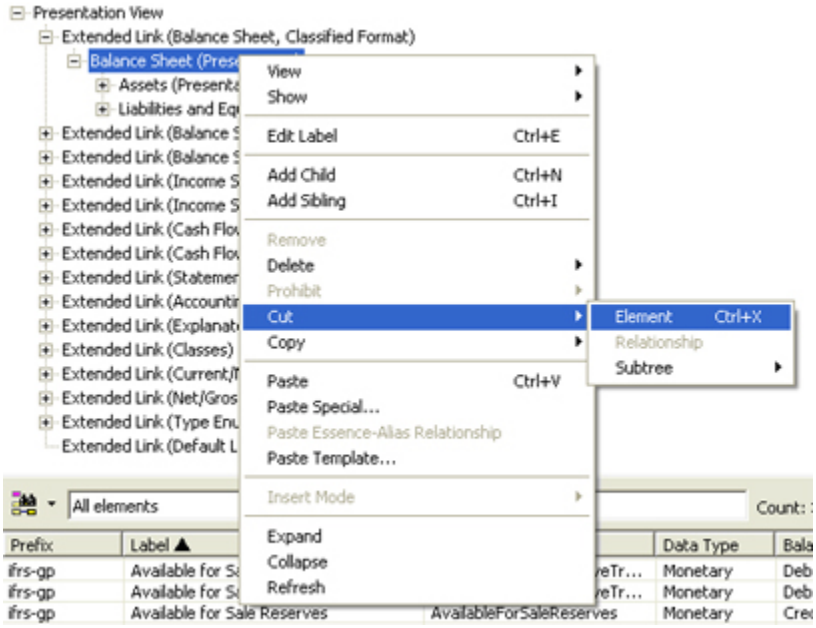
Search Operand	Description	Example
+	Results must include this term	+ Cash + Total
-	Result must exclude the term	+Cash - Total
“”	Search specifically for a string	“Cash Total”

Using Editing Commands

In the relationship views, you can add, remove, and change taxonomy relationships. You can also edit the schema and tuple definitions using the Cut, Copy, Delete, Remove, and Paste menu options.

See Table 4, “Cut, Copy, Delete, Remove, and Paste Actions and Results,” on page 21.

Every cut and paste operation that includes the term element in the menu item affects the schema, and relationship. If the operation involves a concept within a tuple, then this operation also changes the tuple declaration.



Every cut and paste operation that includes the term Relationship in the menu item, affects only arcs (relationship links) and not the schema declaration of the concept.

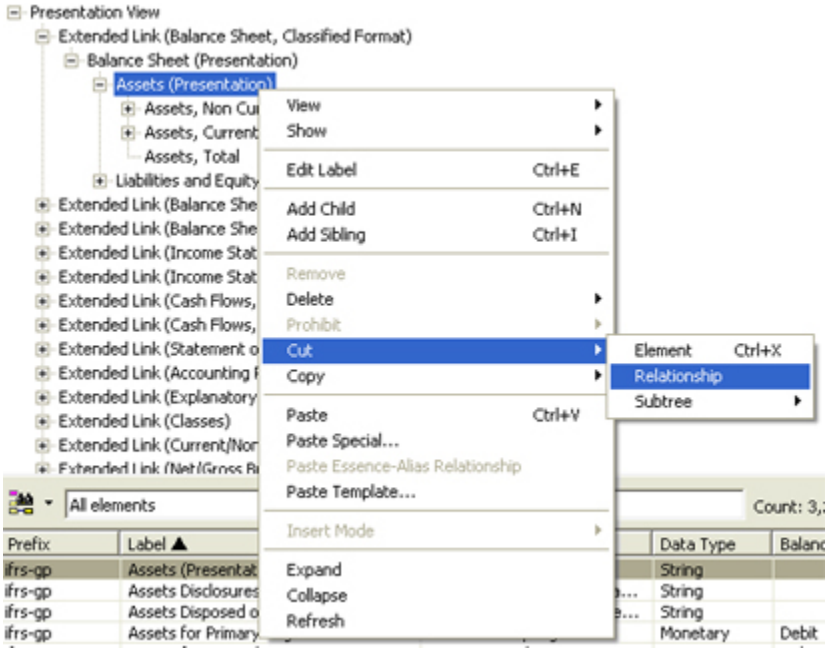


Table 4 Cut, Copy, Delete, Remove, and Paste Actions and Results

Situation	Action	Result
Concept with three label roles: standard, verbose and custom	Copy element from the tree view and paste into a taxonomy.	Message is displayed in the message window indicating that the custom label was created in the taxonomy.
Concept with documentation	Copy element from the tree view and paste into a taxonomy.	Element is copied to a taxonomy with the documentation intact.

Situation	Action	Result
Concept with reference part	Copy element from the tree view and paste into a taxonomy.	If the reference part is not in the taxonomy, a warning message is displayed. To solve the problem, define or remove the data in the reference part before copying the data.
Concept with AppInfoXML	Copy element from the tree view and paste into a taxonomy.	Element copies with a name space prefix. AppInfoXML is also copied.
Concept with custom data type	Copy element from the tree view and paste into a taxonomy.	Warning message is displayed in the message collection window if the taxonomy does not contain the custom data type. The message indicates that a data type was created, but not with the same name as in the original taxonomy.
Concept with formula that refers to a different concept in original taxonomy	Copy element from the tree view and paste into a taxonomy.	Error message is displayed indicating that there is a missing element in the taxonomy.
Concept with a child and three label roles-standard, verbose and custom	Copy parent element with a child from the tree view and paste into a taxonomy.	Element is copied and labeled. This is a valid action.
Concept with a child and three label roles-standard, verbose and custom	Copy parent element relationship with a child from the tree view and paste into a taxonomy.	Element is copied with relationship intact. The child element is copied also. This is a valid action.
Concept with a child that has custom label role	Copy parent element with a child from the tree view and paste into a taxonomy.	Parent and child are copied. The child's custom label role changes to the URI.
Concept with a custom reference part and a child element	Copy parent element with a child from the tree view and paste into a taxonomy.	If ReferencePartDefinition does not contain a value, then concept is copied without the ReferencePartDefinition. If it contains a value, the action is invalid and an error message is displayed.
Concept with a child that has a custom Reference Part	Copy parent element relationship with a child from the tree view and paste into a taxonomy.	If ReferencePartDefinition does not contain a value, the concept is copied without the ReferencePartDefinition. If it contains a value, the action is invalid and an error message is displayed.
Concept with documentation and a child element	Copy element that has a child element from the tree view and paste into a taxonomy.	Element is copied to the taxonomy with the documentation.
Concept with a child that has documentation	Copy element relationship that has a child element from the tree view and paste into a taxonomy.	Element is copied to a taxonomy with the documentation.
Concept with a child that has a custom data type.	Copy element with a child element from the tree view and paste it into a taxonomy.	A message is displayed in the message collection window indicating that the taxonomy does not contain the custom data type, and that a data type was created using a different name.
Base concept with child element from referenced taxonomy.	Copy element into a taxonomy.	Element is copied into a taxonomy.
Base concept with child element from referenced taxonomy.	Copy element with a relationship and child elements.	The element, its relationship, and the child elements are copied and the namespace prefix changes to the taxonomy name.

Situation	Action	Result
Base concept with child element from referenced taxonomy. The child element has a reference part definition.	Copy element with the child element and the child element's relationship that includes a reference part definition.	A message is displayed in the message collection window indicating that the taxonomy does not contain the custom data type, and that a data type was created using a different name.
Elements with a relationship that does not exist in another taxonomy.	Paste relationship into a taxonomy that does not contain the element relationship.	Although this is a valid operation, a message is displayed indicating that the element does not exist in a taxonomy.

About Taxonomy Coloring

For every concept displayed in the tree view, the schema color determines the concept color. In the example below, this color schema is used:

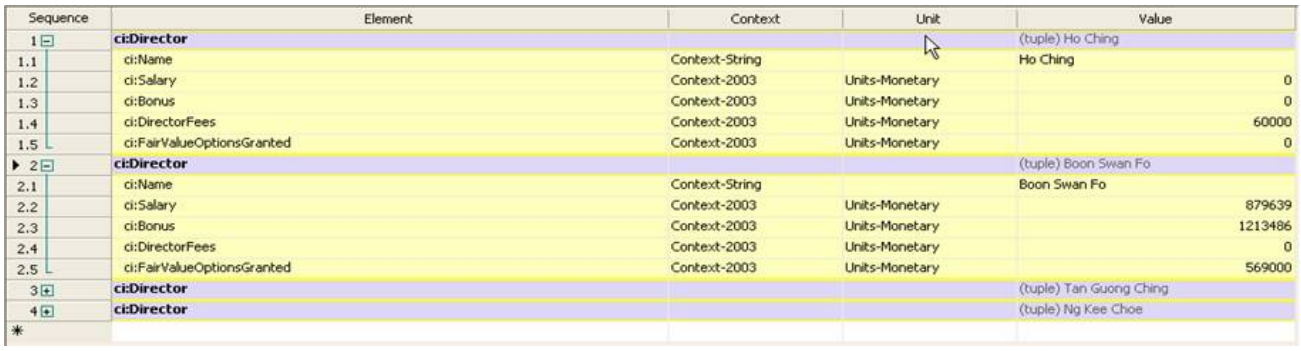
- Concepts from an extension taxonomy is displayed in black
- A base or referenced taxonomy is displayed in blue

To change the color settings, select **Tools**, then **Options**, and then **Taxonomy Coloring**. See "Using the Referenced Taxonomy Collection Editor" on page 58.

Using Instance Coloring

When reporting facts while building an instance document, it can be difficult to determine which facts are grouped together by a tuple and which facts are reported within tuples. Using instance coloring to indicate fact levels helps identify fact groupings.

You can also use color to indicate grouping control, helping to navigate an instance document more effectively. You can also change the instance grid color to adapt to different media such as projectors, or laptop LCD screens.



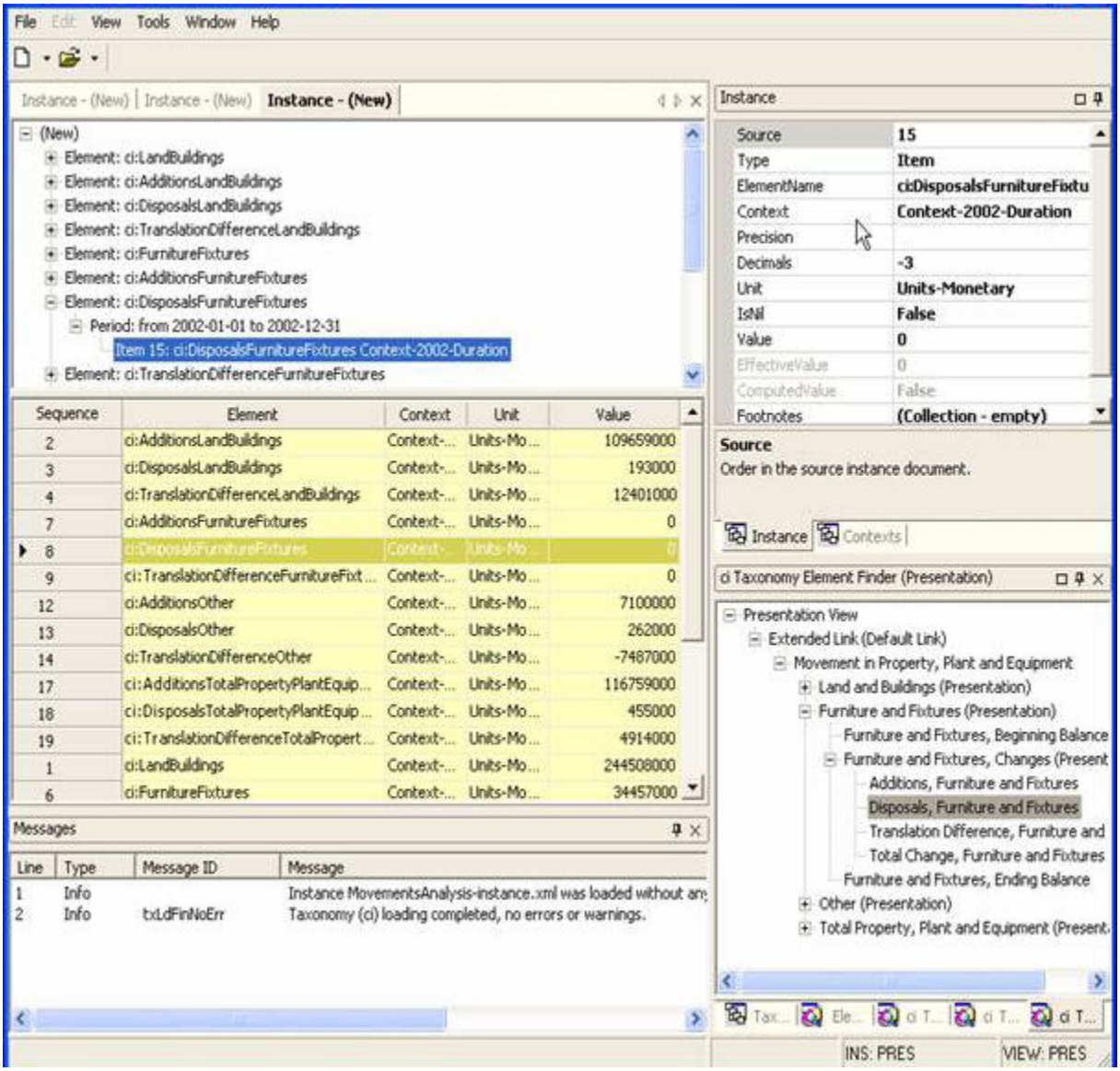
Sequence	Element	Context	Unit	Value
1	**ci:Director**			(tuple) Ho Ching
1.1	ci:Name	Context-String		Ho Ching
1.2	ci:Salary	Context-2003	Units-Monetary	0
1.3	ci:Bonus	Context-2003	Units-Monetary	0
1.4	ci:DirectorFees	Context-2003	Units-Monetary	60000
1.5	ci:FairValueOptionsGranted	Context-2003	Units-Monetary	0
2	**ci:Director**			(tuple) Boon Swan Fo
2.1	ci:Name	Context-String		Boon Swan Fo
2.2	ci:Salary	Context-2003	Units-Monetary	879639
2.3	ci:Bonus	Context-2003	Units-Monetary	1213486
2.4	ci:DirectorFees	Context-2003	Units-Monetary	0
2.5	ci:FairValueOptionsGranted	Context-2003	Units-Monetary	569000
3	**ci:Director**			(tuple) Tan Guong Ching
4	**ci:Director**			(tuple) Ng Kee Choe

In the example above, the tuple is colored in one color, and the tuple facts contained within are displayed in a different color. The grouping control on the side of the grid control assists you in expanding and collapsing tuple facts.

Synchronizing Instances

To assist users navigating an instance or editing a taxonomy within an instance, select a fact in the instance tree view or instance grid and it is synchronized with the tree view property display.

If the instance window does not synchronize with the taxonomy window, ensure that the correct view is selected (Presentation, Calculation, Definition), and that Sync to Panes is set for the correct view.



Enabling Editing

If you open locked or restricted taxonomies or instance documents, XBRL Taxonomy Designer prevents the application from modifying them.

Utility Information	
ReadOnly	True
CheckedOut	False
Encoding	utf-8
WriteExtendedAppinfo	False
NestingDepth	0

➤ To enable a taxonomy or instance document for editing:

1 Select **Open**.

2 Select **Enable Edting**.

Printing Instance Documents

The print functionality provided helps debug instance documents as you create them. A key print function is the ability to provide an audit trail of source information imported under mapping control.

The first time in any session that you select Page Setup, Print, or Print Preview, Windows management functions search for a printer. It is not possible to use these options unless a printer driver is installed and usable.

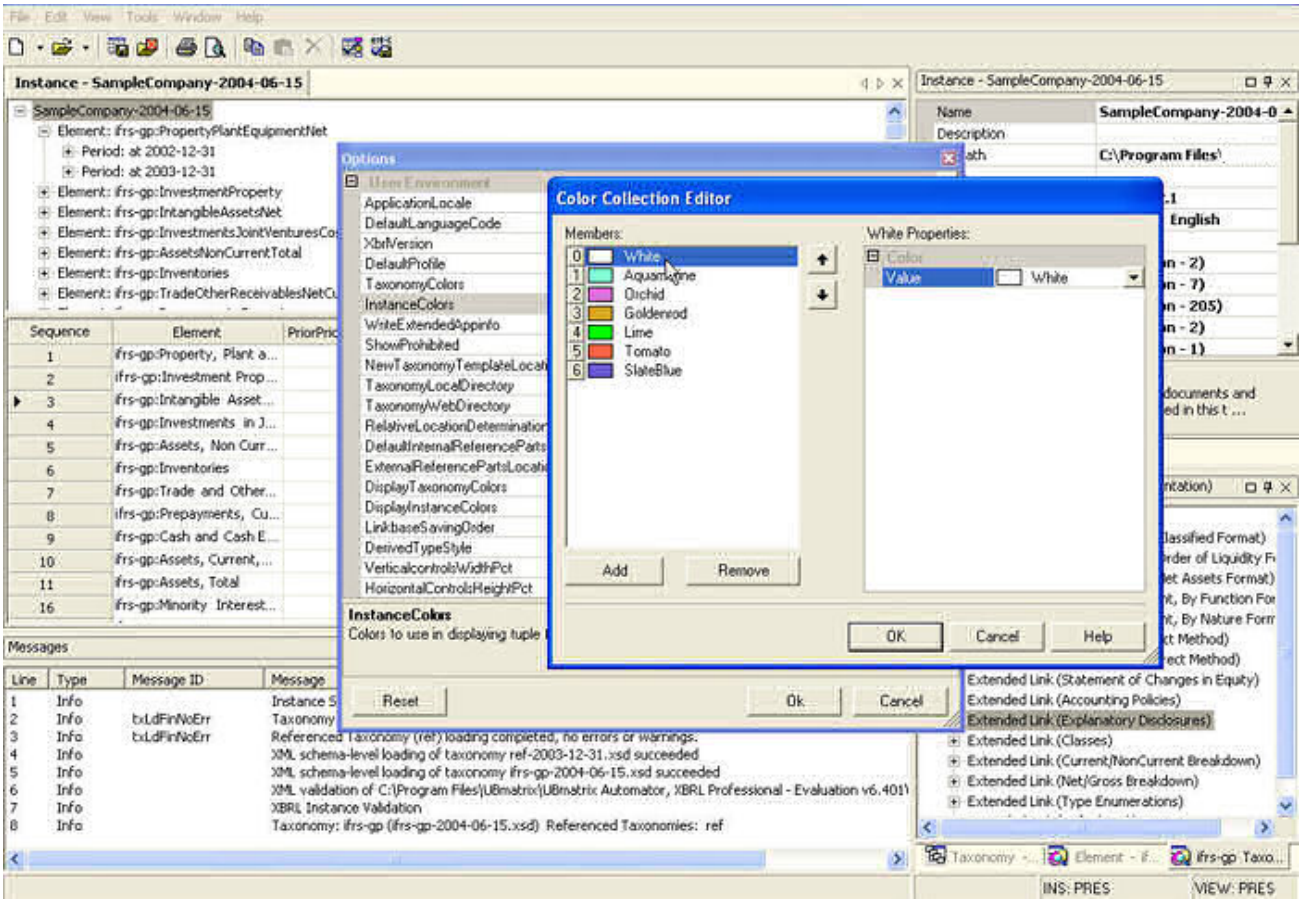
Note: A newly installed system with no printer or a laptop computer with a networked printer operated from an airplane cannot use the print options. To solve the laptop problem, install a phantom local printer on the laptop local printer port.

Configuring Color Options

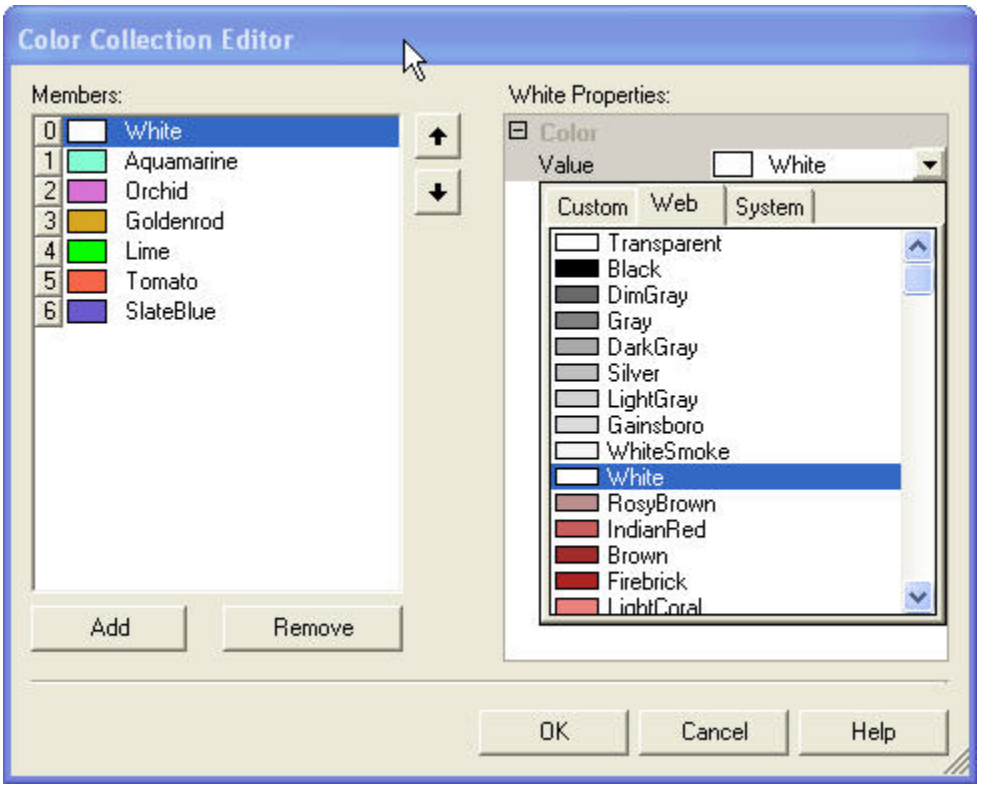
➤ To change the instance coloring:

1 Select **Tools**, and then **Options**.

2 Double-click **InstanceColors**.



3 Select the sequence to change.



4 Alter the sequence in the Properties page, and then click **OK**.

About Native Document Format Files

Whenever a taxonomy is saved, XBRL Taxonomy Designer saves it in both native document format and the document's XBRL format.

The native document format is an XML format that contains taxonomy information for a DTS. The file has `_native.xml` appended to the file name and is saved in the same directory as the taxonomy `.xsd` file. For example, a file named `MyTaxonomy_native.xml` exists in the same subdirectory as `MyTaxonomy.xsd`.

The native document file contains information used within the application that is not stored in XBRL files. The native document saves application and metadata settings that are lost when saving taxonomies to XBRL. This file also aggregates the base edited taxonomy with referenced taxonomies.

The XBRL specification does not include the types of information stored in the native document file. For example, notes, taxonomy-specific metadata such as data types, roles, arc roles; and other options specific to the taxonomy are saved in this file.

3 Using Taxonomies

In This Chapter

About Taxonomies

An XBRL taxonomy is a collection of elements, linkbases, role definitions, reference part definitions, and data type definitions. Linkbases contain extended links and, depending on linkbase type relationships, labels, formulas, or references.

- Relation-type links include presentation, calculation, and definition. Definition relations are further specified by several role types, such as general-special and essence-alias.
- There are three kinds of documentation-type links: labels, references, and formulas.
- The collection of all taxonomies and linkbases is called a DTS. XBRL allows discovery by encounter of an undeclared reference, such as by relationship end, to a taxonomy not identified by importing reference.

An XBRL taxonomy incorporates referenced taxonomies, which contain all taxonomy parts, as well as reference XML schemas (for example, type definitions and other definitions). The referencing user of a standard taxonomy cannot edit standard referenced taxonomies, but can prohibit the traversal of relationship-type links created in those taxonomies and override labels and references.

When you create a group of taxonomies, you can edit them all. You also can modify the referenced taxonomies directly if you create the referencing taxonomies.

Using Extended Links

A taxonomy is a collection of extended links that contains arcs (relationships) between elements or a resource . Each relationship view has one or more extended links. These extended links contain relationships between elements. Extended links are independent networks of relationships.

Taxonomy creators can create Extended Link Roles that help describe and organize their extended links.

Differentiating Between Elements and Links

The Relationships View pane shows a tree view of the links or relationships between elements. The Elements List pane shows elements in a dictionary style.

You can add, delete, copy or move:

- Elements
- Relationships
- Elements and their child elements
- Relationships and their child relationships

About Multiple Taxonomy DTS

A loaded taxonomy may reference other taxonomies explicitly through import constructs in its `.xsd` file, or implicitly by finding references to roles, linkbase arcs, or reference part definitions. Referenced taxonomies are loaded as identified and discovered and form a DTS. The DTS is represented as a tree. You can view and edit the tree as the Referenced Taxonomies Collection of the opened (base) taxonomy.

Taxonomies other than the open taxonomy may be read-only (for example, when you use standard taxonomies through an extension taxonomy). You can edit these taxonomies if you own the referenced taxonomies. You can specify whether the Open Taxonomy window and the referenced taxonomy (after loading its DTS tree) can be edited or are read-only.

About Tuples

A tuple is a structure of items in an instance document. It is defined in XBRL 2.1 as complex data types (definition links in XBRL 2.0). For XBRL 2.1, one XML schema element is saved in the `taxonomy .xsd` file specifying the tuple structure. Tuples can contain items or other tuples, and can be nested to any depth required.

V2 tuples (they do not exist in V1) are usually defined in the definition linkbase. However, many V2 taxonomies do not provide such definition links, and use presentation links to identify tuple item collections. The taxonomy loader recognizes definition or presentation links from tuple-to-tuple items as a tuple structure.

The conversion process maintains the tuple composition as loaded by the V2 taxonomy loader, only changing the schema compositional element to all or sequence as indicated.

Using the Taxonomy Editor

These topics describe the Taxonomy Editor components:

- "Taxonomy Document Editor Menu Options" on page 31
- "Taxonomy Editor Toolbar Buttons" on page 35
- "Taxonomy Editor Messages Pane" on page 36

Taxonomy Document Editor Menu Options

The menu options and toolbars change between the instance document and the taxonomy. Select the taxonomy of the instance document controls (for example, Taxonomy Relationship View or Taxonomy Elements List) to view the taxonomy editing toolbar and menus. Select any instance control (for example, Instance Relationship View, Instance Fact Values Grid, Instance Dockable Controls or Property Grids) to view the instance editing menu and toolbar.

Table 5 Taxonomy Editor File Menu Options

Taxonomy Editor File Menu Options	Descriptions
New	Creates an instance document or taxonomy.
Open	Opens a taxonomy or instance document.
Close	Closes an open taxonomy or instance document.
Save	Saves the selected taxonomy.
Save as File	Saves the selected taxonomy to the location you specify.
Save to Server	Saves the selected taxonomy to the Web location you specify.
Load reference taxonomy	Loads a referenced taxonomy into the instance document's namespaces collection.
Import	Imports a taxonomy or portion of a taxonomy from Microsoft Excel, Access, OLE DB Provider databases, Microsoft Word, CSV files, or XML.
Export	Exports a taxonomy or portion of a taxonomy to Microsoft Excel, Access, OLE DB Provider databases, CSV files, or XML.

Taxonomy Editor File Menu Options	Descriptions
Page Setup	Displays the page setup window for specifying print options.
Print	Enables you to print a report of instance document data, contexts, and the source data mapping audit trail.
Print Preview	Displays a report of instance document data, contexts, and the source data mapping audit trail for you to check before printing.
Exit	Exits XBRL Taxonomy Designer.

Table 6 Taxonomy Editor Edit Menu Options

Taxonomy Editor Edit Menu Options	Descriptions
Undo	Reverses one or more changes made to the taxonomy. After exiting XBRL Taxonomy Designer, Undo is not available. However, if the log file option is enabled, you can manually reverse changes using the logged steps.
Edit Label	Edits the label of the taxonomy element using the current relationship tree view. Note that you can also edit the element by selecting and clicking the element again (do not double-click because this expands or collapses the tree) from within the relationship tree view.
Add Child See "Adding Child Elements" on page 78.	Adds an element as a child of the current relationship view selection.
Add Sibling .	Adds an element as a sibling of the current relationship view selection. "Adding Sibling Elements" on page 79
Remove	Removes an element with no parent or child relationships from its immediate parent extended link of the relationship tree view. This element is not deleted, or removed from the element list view or from any other extended link. Removed elements are displayed in gray.
Delete	Select **Delete** and then **Element** to delete the currently selected element. Select **Delete**, and then **Relationship** to delete the relationship of the currently selected element. The element remains in the elements list. If no other relationships in this extended link exists, the element becomes a root element to the current extended link. Select **Delete** and then **Sub tree** to delete elements and relationships of the subtree. Select **Delete**, then **Sub tree**, and then **Relationships** to delete the relationships of the selected subtree. The elements and all other relationships of elements in the subtree remain.
Cut	Removes elements, relationships, subtree elements and subtree relationships, and makes them available to paste in a different location from the clipboard. You can paste a cut object using the clipboard to Automator's taxonomy editor or other tools compatible with clipboard-contained XBRL.
Copy	Copies elements, relationships, subtree elements and subtree relationships, and makes them available to paste in a different location from the clipboard. You can paste a copied object using the clipboard to Automator's taxonomy editor or other tools compatible with clipboard-contained XBRL.

Taxonomy Editor Edit Menu Options	Descriptions
Paste	Adds previously cut or copied elements, relationships, or subtrees of elements and relationships at the selected position in the taxonomy tree. The shape of cursor at emplacement determines whether the paste results are as child, sibling before, or sibling after the emplacement. If the names of pasted objects are the same as the names of similar objects in the receiving taxonomy, the pasted elements' names are appended with Copy, Copy2, etc., to prevent duplicate names.
Paste Special	Provides explicit control of pasting into multiple relationship tree views, including choice of language and role of labels controlled, and other options discussed.
Paste Essence-Alias Link	For version 2.0 taxonomies, pastes a same-as (Essence-Alias definition link) between the element selected, using copy, that becomes the essence part of the relationship, and the paste which becomes the alias side of the relationship. For version 2.1, the definition tree view, essence-alias role, is used instead.
Paste Template	Displays the Patterns Taxonomy dialog box. The template you select is inserted as a child of the current tree view extended link or element. This option helps build complex taxonomy sections. The templates act as a guide to creating types of taxonomy components.

Table 7 Taxonomy Editor View Menu Options

Taxonomy Editor View Menu Options	Descriptions
Presentation	Displays Presentation links relationships in the tree view.
Calculation	Displays Calculation links relationships in the tree view.
Definition	Displays in the tree view Definition links relationships for the arc role (relation type) indicated in the toolbar (general-special, essence-alias, etc.).
Labels	Displays labels of the default language and default label role in the taxonomy tree view, instead of element names.
Element Names	Displays element names in the taxonomy tree view rather than the label for the selected language and label role.
Namespace Prefixes	Prepends the namespace prefix before the label or tree view element name.
Prohibited Relationships	Displays prohibited links in the tree view, which are displayed in strikethrough font.
Show	Displays the Elements List pane, changes the orientation of window panes and controls, and enables you to add Relationships View panes.
Expand	Expands elements of the selected subtree.
Collapse	Collapses elements of the selected subtree.
Refresh	Refreshes the Relationships tree view display.

Table 8 Taxonomy Editor Tools Menu Options

Taxonomy Editor Tools Menu Options	Descriptions
Validate Taxonomy	Validate the current taxonomy, for XML or XBRL checks, and to specify saving a report file. See "Validating Taxonomies" on page 51
Renaming Wizard	Displays the element renaming wizard to rename elements of a taxonomy.
Flatten Taxonomy	Elements, labels, references, formulas, and relationships of referenced taxonomies are copied into the base taxonomy, and the referenced taxonomies are removed from the top taxonomy referenced taxonomies collection. (At this time, Flatten does not copy referenced custom data types, reference part definitions, or other taxonomy metadata.)
Rebuild Tuple Definition	For a selected tuple element, this option uses the presentation subtree of relationships to rebuild a tuple definition (for example, the complex type structure of the tuple content items). This structure can be viewed graphically from the element property grid, Tuple Definition property.
XBRL Mapper	Initiates the XBRL mapper setup dialog, and subsequent mapping control. See "XBRL Mapper" on page 170.
Options	Opens the Options form where you can set XBRL Taxonomy Designer options.
Element Label Spell Checker (Plug-In)	Spell checks specified labels by language and role. Dictionaries are derived from http://lingucomponent.openoffice.org/spell_dic.html dictionaries and are user-modifiable. (For more on this, see the Plug-In Tools chapter in this guide.)
Element Label Renamer (Plug-In)	This plug-in replaces substrings of characters in selected taxonomy labels. Selected elements, subtrees, or all taxonomy labels substrings can be replaced. Options are available to change the element name when the default role/language label is changed.
Linkbase Tidier (Plug-In)	This plug-in cleans up linkbases. Current options are to clear titles and reorder. See "Linkbase Tidier" on page 98.
XBRL V2 To V2.1 Taxonomy Converter (Plug-In)	Converts V2 taxonomies to V2.1 taxonomies, allowing specified options for linkbase and element changes for V2.1. Can also be applied to taxonomies in V2.1 format that require the same conversion actions. See "V2 to V2.1 Conversion" on page 98.

Table 9 Taxonomy Editor Windows Menu Options

Taxonomy Editor Windows Menu Options	Descriptions
Show All Dockable Controls	Shows all hidden dockable controls including docked controls displayed on the right side of the XBRL Taxonomy Designer window, and free-floating controls (for example, those users drag to a second monitor).
Hide All Dockable Controls	Hides all dockable controls.
Taxonomy Properties	Enables you to select the taxonomy properties form to display for all taxonomies loaded into the application.
View Taxonomy Tab	Enables you to navigate to the tabbed window of a taxonomy or instance document. Select a numbered tab to which to navigate.

Table 10 Taxonomy Editor Help Menu Options

Taxonomy Editor Help Menu Options	Descriptions
Contents	Select this option or press F1 to access XBRL Taxonomy Designer online help.
License agreement	Displays the XBRL Taxonomy Designer End-User License Agreement (EULA).
About	Displays the product and license information for XBRL Taxonomy Designer.

Taxonomy Editor Toolbar Buttons

This section explains the Taxonomy Editor toolbar.

Table 11 Taxonomy Editor Toolbar Buttons

Button	Description
	Creates a new file
	Opens a file
	Saves a file
	Closes a file
	Adds a sibling
	Adds a child
	Moves an element up in the editable taxonomies
	Moves an element down in the editable taxonomies
	Same functionality as selecting **Edit**, and then **Delete**.
	Undoes the last actions. You can specify how actions to undo. Entries in the undo list are also saved in the log file if the option is enabled (select **Tools**, and then **Options**).

Taxonomy Editor Messages Pane

The Messages List pane displays error messages, warnings, and other informational messages generated when loading and validating a taxonomy, or importing information. Double-click a message to navigate to the element associated with the message.

As a process proceeds, you can quit if errors occur, and then allow the application to correct the errors, or correct them manually.

Opening a Taxonomy

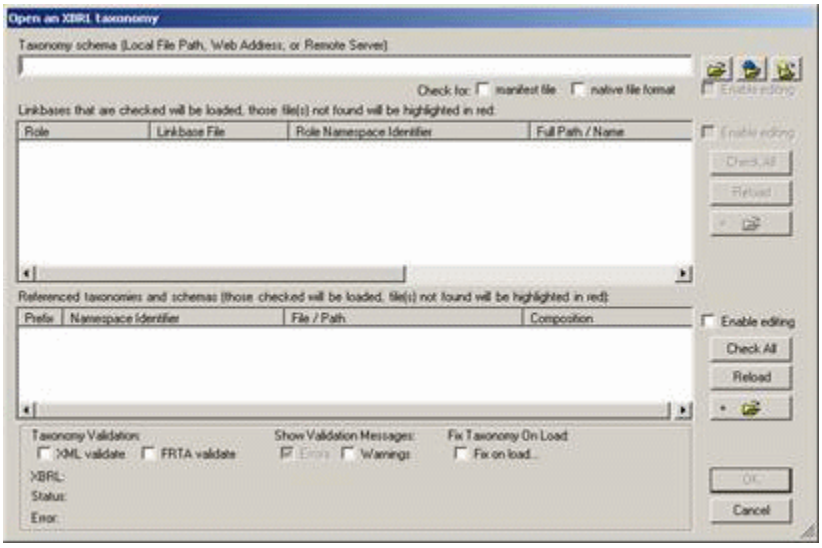
A taxonomy can be opened from a:

- local file system
- local or remote directory listing
- Web-based XBRL Taxonomy
- remote server location
- a manifest

Open an XBRL Taxonomy from the Local File System

➤ To open a taxonomy from your local system using the XBRL Taxonomy Designer main screen:

1 **Select File, then Open, and then Taxonomy.**



There are several methods of loading taxonomies, and the file types listed within the browse dialog. Each file type discovers the taxonomy using a different method.

Table 12 Taxonomy File Types

File Type	Discovery Result
XBRL Taxonomy (`.xsd`)	Selects an XBRL `.xsd` schema
Native file format (`_native.xml`)	Opens the taxonomy native file format file
XBRL Linkbase (`.xml`)	Opens the linkbase and discovers the taxonomy
DTS Manifest (`.xmf`)	Opens the manifest and discovers the taxonomy

Three icons at the top right of the window open taxonomies. At the bottom of the window several load taxonomy options are displayed.

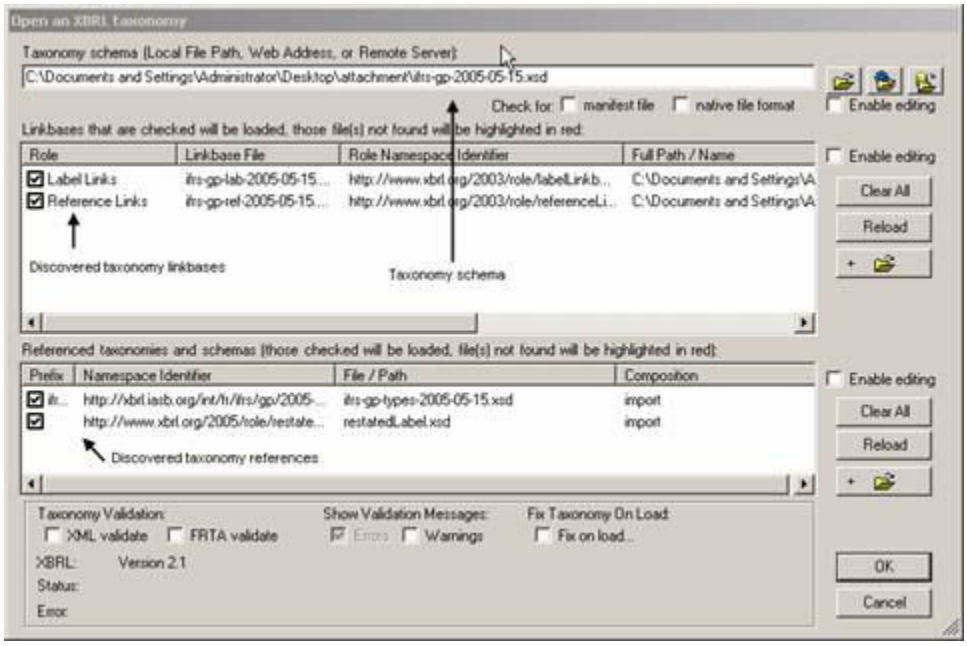
-  Browse to any accessible location on your computer and select a taxonomy file.
-  Access taxonomies stored in a Taxonomy Directory on a user-defined Web location.
-  Access taxonomies stored in a local Taxonomy Directory.

2 Click  to select **Select an XBRL Taxonomy**, and then navigate to a taxonomy file.

3 Browse to the taxonomy.

4 **Double-click the `.xsd` taxonomy file, and then click Open.**

Linkbase and Referenced Taxonomy information for the taxonomy is displayed in the Open Taxonomy window.

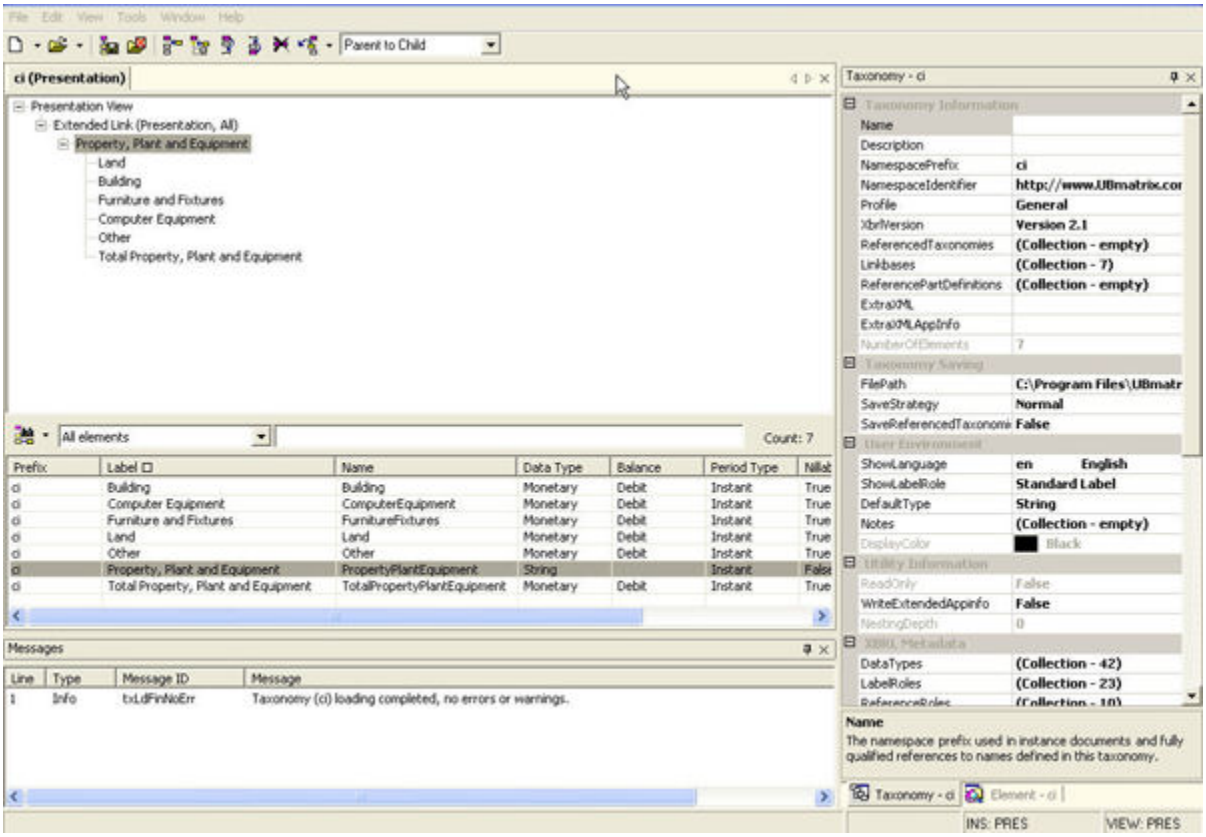


5 **Select the appropriate check box to complete XML validation.**

6 **Select load options, and then click OK to load the selected taxonomy.**

See "Using Taxonomy Load Options" on page 43.

The Taxonomy Editor screen is displayed with the selected taxonomy loaded and ready for editing.



Opening an XBRL Taxonomy from a Local or Remote Directory Listing

Local and Remote directory lists enable you to define a set of common files you open regularly. Selecting a file from these lists, accesses the selected file without typing the Web URL or navigating to the directory.

The local directory `XML` file is installed in this location:

```
Oracle\Disclosure Management\XBRL Taxonomy Designer\templates
```

This is a sample entry:

```
<TaxonomyDirectoryEntry>
 <ID>20</ID>
<NamespaceIdentifier>http://www.Oracle.com/Patterns/BasicCalculation</
NamespaceIdentifier>
 <Category>Financial Pattern Samples</Category>
 <Title>Basic Calculation</Title>
 <PreferredNamespacePrefix>ci</PreferredNamespacePrefix>
 <Description>Very basic calculation.</Description>
 <DateReleased>2005-02-01</DateReleased>
 <XBRLVersion>XBRL 2.1</XBRLVersion>
 <Location>BasicCalculation.xsd</Location>
 <Status>Draft</Status>
 <SortOrder>20</SortOrder>
 <PrintOut>None</PrintOut>
 <Contact>someone@someemail.net</Contact>
 <HasLabels>True</HasLabels>
 <HasPresentation>True</HasPresentation>
```

```
<HasCalculation>True</HasCalculation>
<HasDefinition>False</HasDefinition>
</TaxonomyDirectoryEntry>
```

➤ To open the taxonomy:

1 **Configure the directory**`XML` **file.**

2 Select **File**, then **Open**, then **Taxonomy**, and then select the taxonomy.

Opening a Web-based XBRL Taxonomy

➤ To open a Web-based taxonomy:

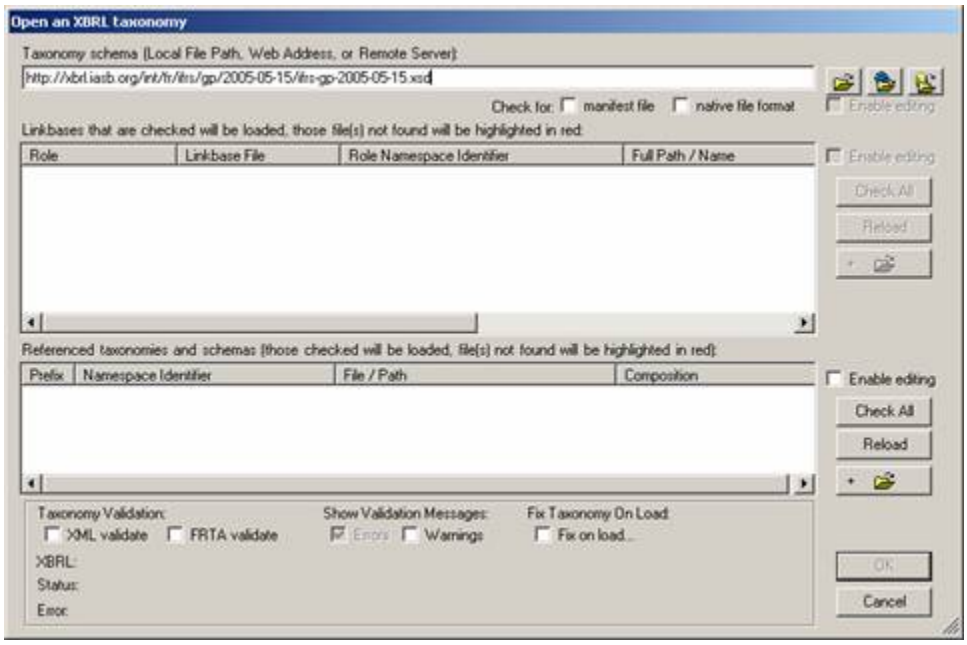
1 Select **File**, then **Open**, and then **Taxonomy**.

2 Enter the taxonomy's Web address in the **Taxonomy schema** field.

For example:

http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/ifrs-gp-2005-05-15.xsd

3 Click **Enter**.



The loading process is displayed when the files are requested and the progress of the downloaded.

Taxonomies opened from a Web-based location are cached automatically within the `\cache` install directory. See "Caching XBRL Taxonomies" on page 51.

Select 

to end the loading before it completes. Oracle recommends that you clear your local cache after stopping the load operation because the file may have been partially loaded.

If you open particular Web-based taxonomies regularly, you can create a Web-based directory listing for them.

Opening an XBRL Taxonomy from a Remote Server Location

Taxonomy Management Server (TMS) users maintain taxonomies in a remote taxonomy document repository.

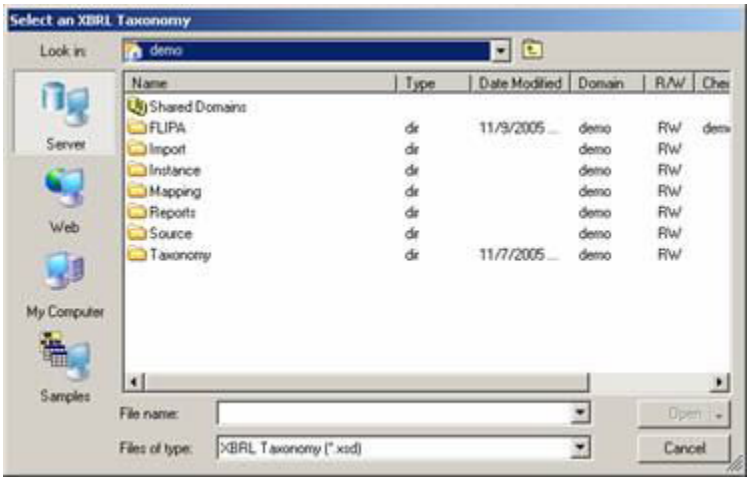
➤ To open an XBRL taxonomy from a remote repository:

1 Select **File**, then **Open**, and then **Taxonomy**.

2 Select the remote server.

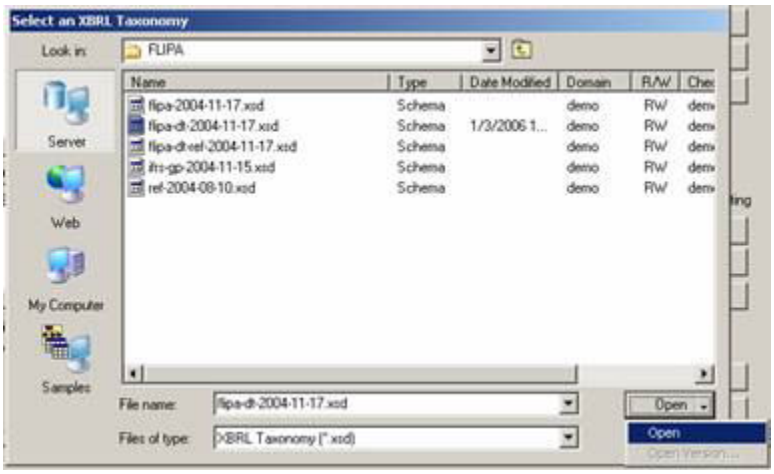
3 Select **Server**, and then log on to the TMS server.

4 Navigate to the repository folder, and then select the taxonomy to load.

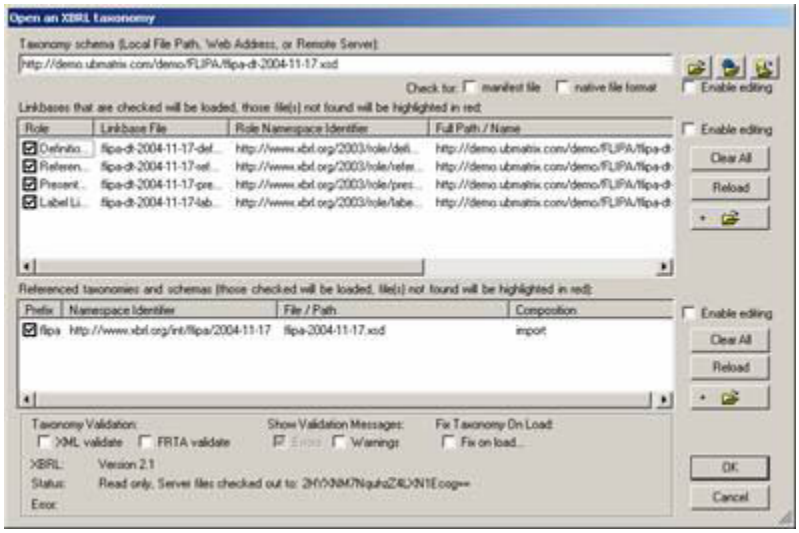


See "Creating DTS Manifest Files" on page 45.

5 Browse to and select the taxonomy.



6 Select **Open** or open a version of the taxonomy.

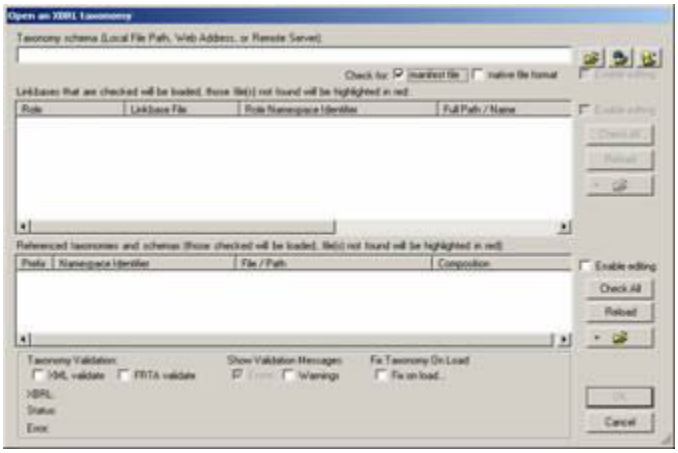


Opening an XBRL Taxonomy Using a Manifest

Manifest (`.xmf`) files contain an inventory of the schemas and linkbases included in a Taxonomy DTS. Parsing the taxonomy to discover the DTS during loading can be time consuming with larger taxonomies. Manifests shorten the discovery process because they includes the paths of all files that are loaded. You can create manifests when you save taxonomies. By default, this option is not eanbled.

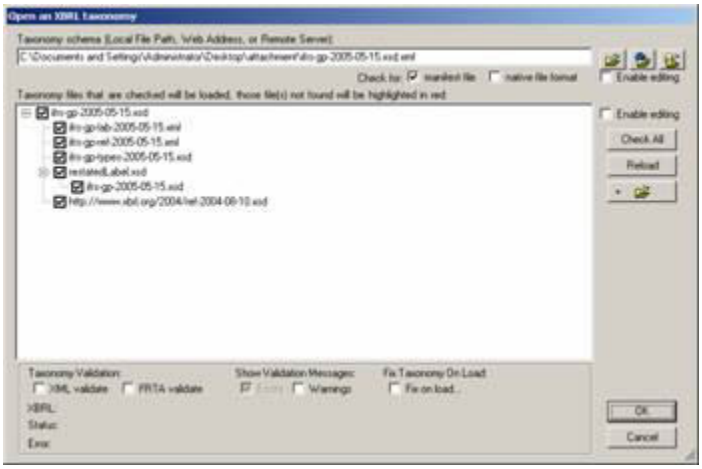
➤ To open a manifest:

1 **Select File, then Open, then Taxonomy, and then Manifest.**



2 **Browse to the manifest.**

See "Creating DTS Manifest Files" on page 45.



Using Taxonomy Load Options

There are several Taxonomy load options in the Open Taxonomy window.

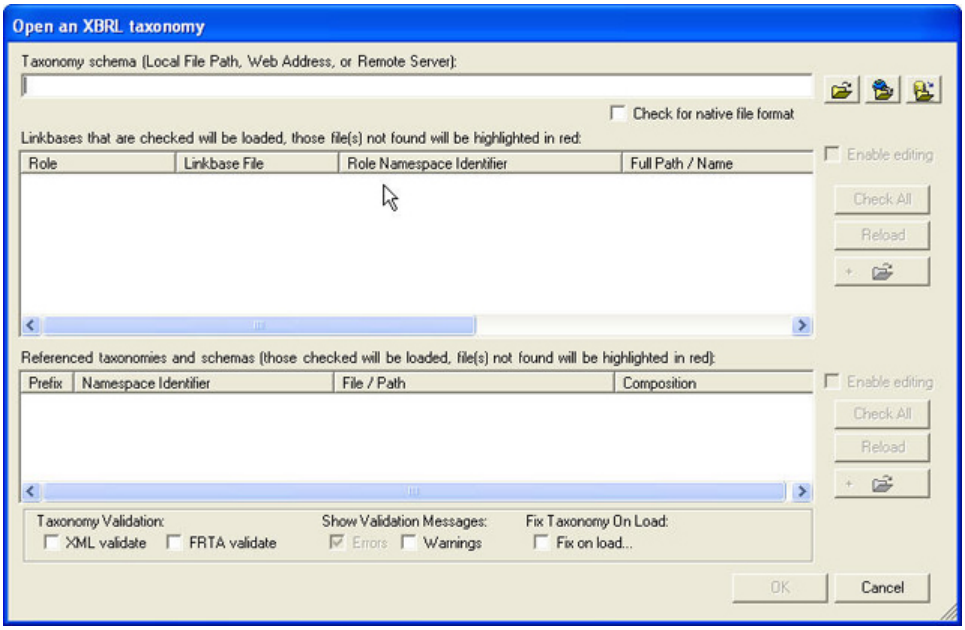


Table 13 Taxonomy Load Options

Taxonomy Load Options	Description
Check for native	Filters for taxonomies saved in the Oracle Hyperion Disclosure Management proprietary native format. See "About Native Document Format Files" on page 28.
Fix on load	XBRL Taxonomy Designer attempts to fix minor errors in the taxonomies (for example, duplicate arcs).
XML validate	Validates the Instance Document during load.

Taxonomy Load Options	Description
FRTA validate	Performs FRTA Validation of the Instance Document during load. XBRL.org publishes Financial Reporting Taxonomies Architecture (FRTA) taxonomy design and implementation guidelines. The FRTA document is available at: http://www.xbrl.org/TaxonomyGuidance/. FRTA establishes rules and conventions that make taxonomies more usable and efficient. Financial reporting taxonomies must comply with FRTA to obtain formal XBRL approval.
Show Messages: Errors	Specifies the level of errors that are displayed in the messages dialog. Select this option to display nonfatal errors in Instance or Taxonomy validation.
Show Messages: Warnings	Specifies the level of warnings that are displayed in the messages dialog. Select this option to see nonfatal errors in Instance or Taxonomy validation.
Enable Editing	Unlocks read only taxonomies. Set this option to a taxonomy extension (for example, the US-GAAP).

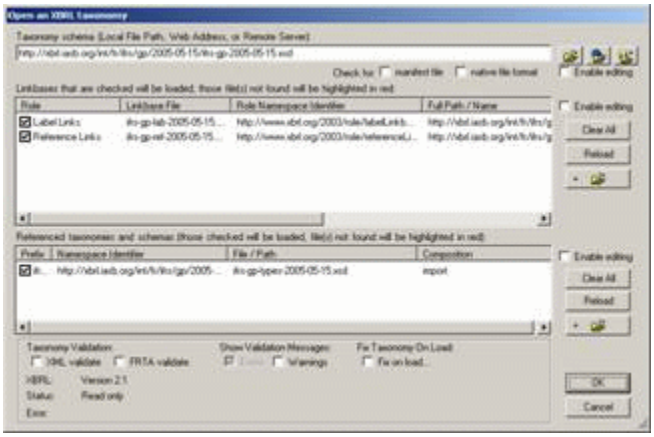
Adding and Removing References When Loading a Taxonomy

When you open a taxonomy you can add reference taxonomies.

➤ To add or remove reference taxonomies:

1 **Select File, then Open, and then Taxonomy.**

2 **Select the taxonomy.**



The reference taxonomies that were discovered with the selected schema are displayed.

3 **Select the reference taxonomy schema.**

Adding and Removing Linkbases When Loading a Taxonomy

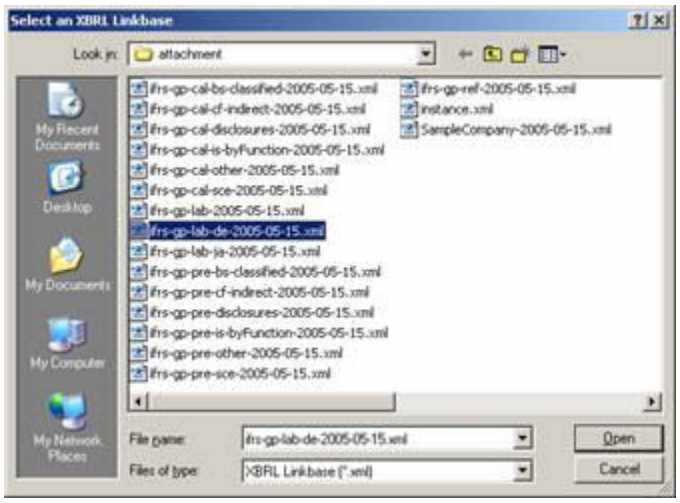
After opening a taxonomy, you can add linkbase resource files (for example, a linkbase that contains translated labels for the taxonomy).

➤ To add or remove linkbases:

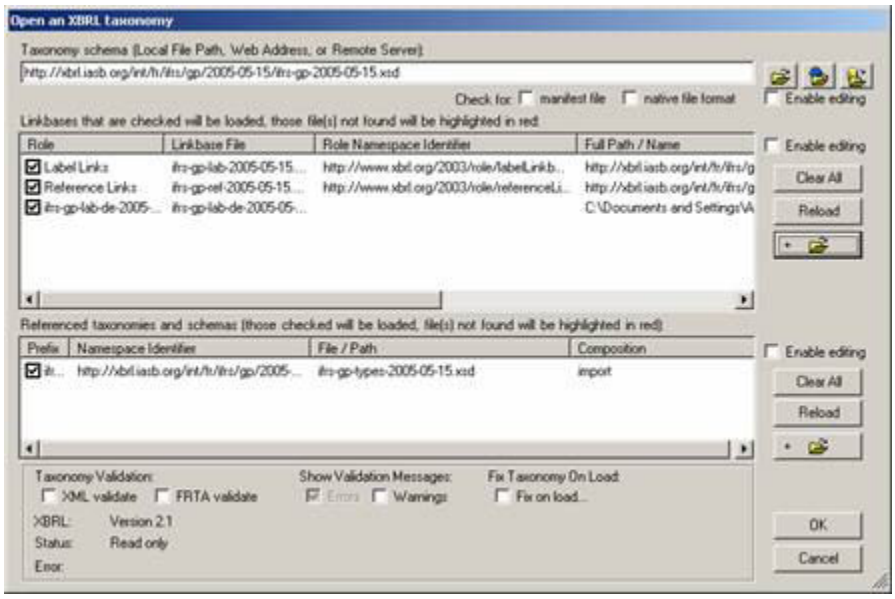
1 **Select File, then Open, and then Taxonomy.**

2 Browse to and select the taxonomy.

3 Select and browse to the linkbase to add, and then select **Open**.



4 Select the linkbase, and then select **Open**.



Note the added linkbase is appended to the linkbase list.

See "Detached and Linked Linkbases" on page 141.

5 Select or clear the checkbox associated with each linkbase, as appropriate, or select Clear All to remove all linkbases.

Creating DTS Manifest Files

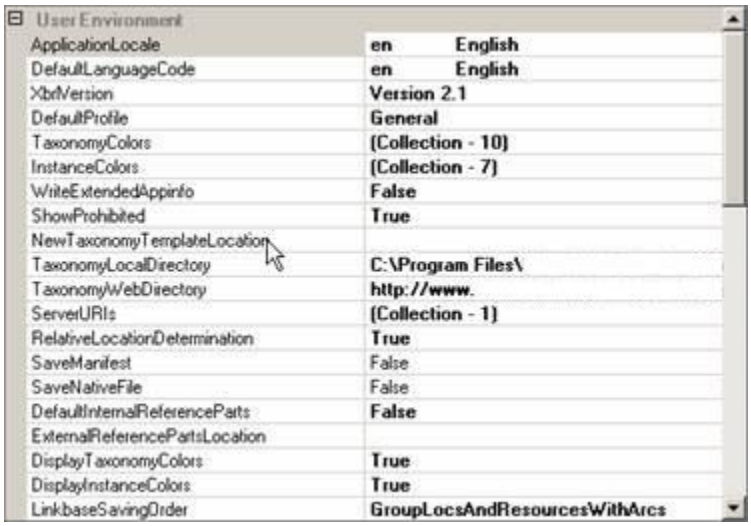
Manifest files maintain an inventory or record of all schemas, and linkbase versions that are part of the DTS of the taxonomy.

➤ To create a manifest (`.xmf`) file:

1 Select **Tools**, and then **Options**.

2 Select **Save Manifest**, and then **True**.

When you save the taxonomy, an additional file is saved with the extension `.xmf`.



➤ To save an XBRL taxonomy and create a manifest:

1 Select **File**, then **Open**, and then **Taxonomy**.

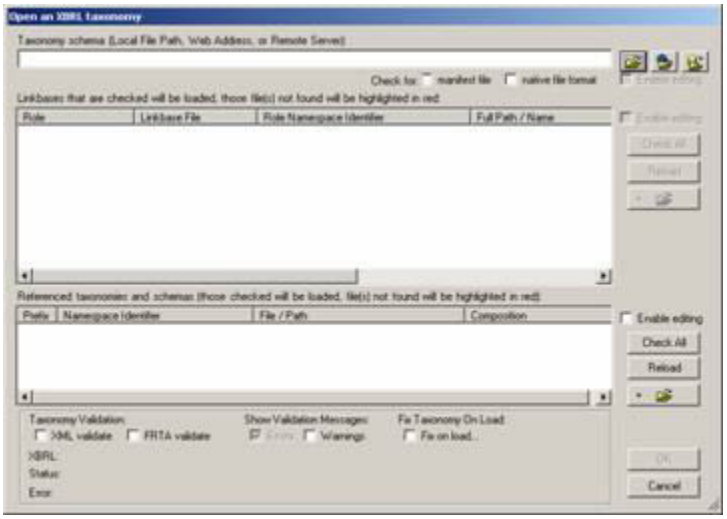
2 Select from the server the taxonomy to load.

3 Select **File**, and then **Save to Server**.

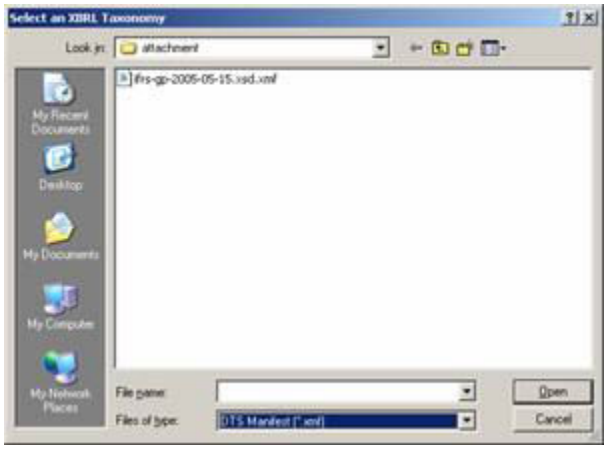
An `XMF` file is created.

➤ To open an XBRL taxonomy using the manifest file:

1 Select **File**, then **Open**, and then **Taxonomy**.



2 Select the **Manifest** check box, or select **DTS manifest** from the file type list using Browse.



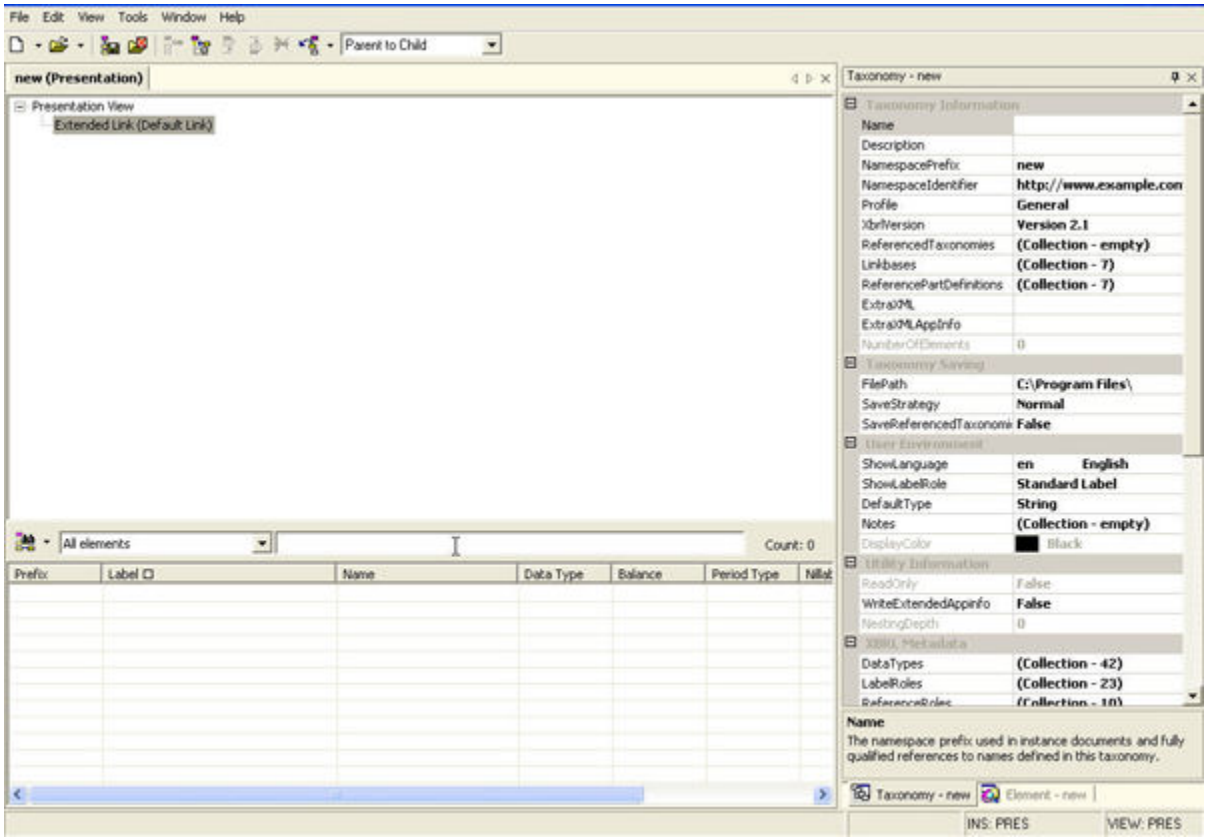
3 **Browse to the manifest (`.xmf`) file and select it to load the manifest and the DTS tree.**

Creating a Taxonomy

You can create a taxonomy that is not based on another one.

➤ To create a taxonomy from the Taxonomy Editor, select **File**, then **New**, and then **Taxonomy**.

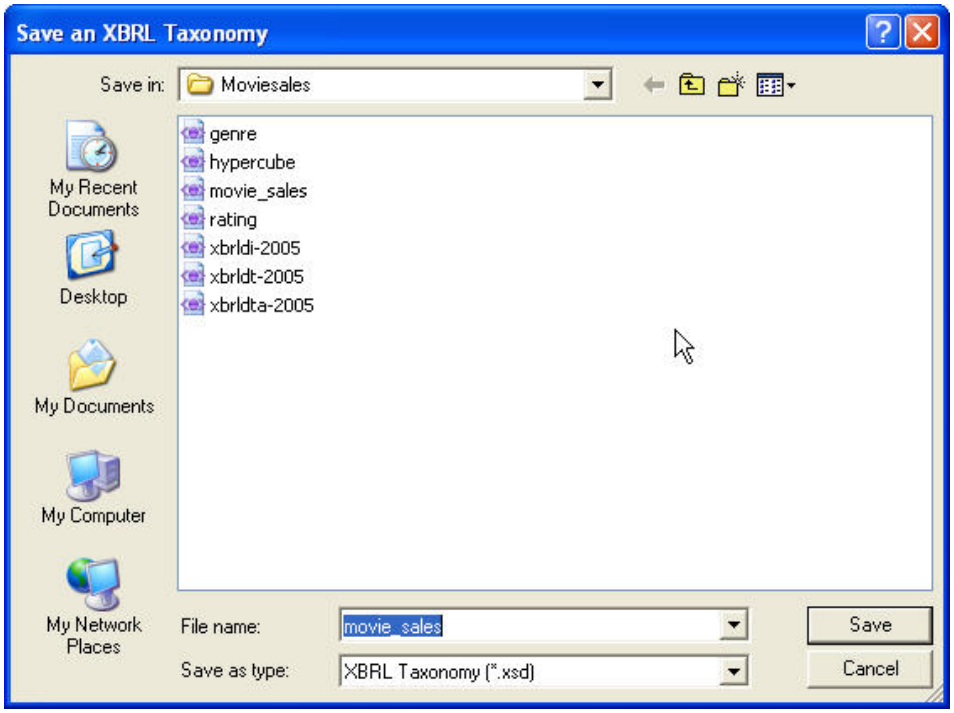
Using the taxonomy editor, you can add elements and relationships to this taxonomy (see "Adding Child Elements" on page 78).



Saving a Taxonomy

➤ To save a taxonomy:

1 Select **File**, and then **Save**.



2 In the **File Name** field, enter the taxonomy name.

3 In the **Save In** field, browse to the appropriate location.

4 Click **Save**.

5 Select **File**, and then **Close** to close the taxonomy.

Saving a Taxonomy to a Server

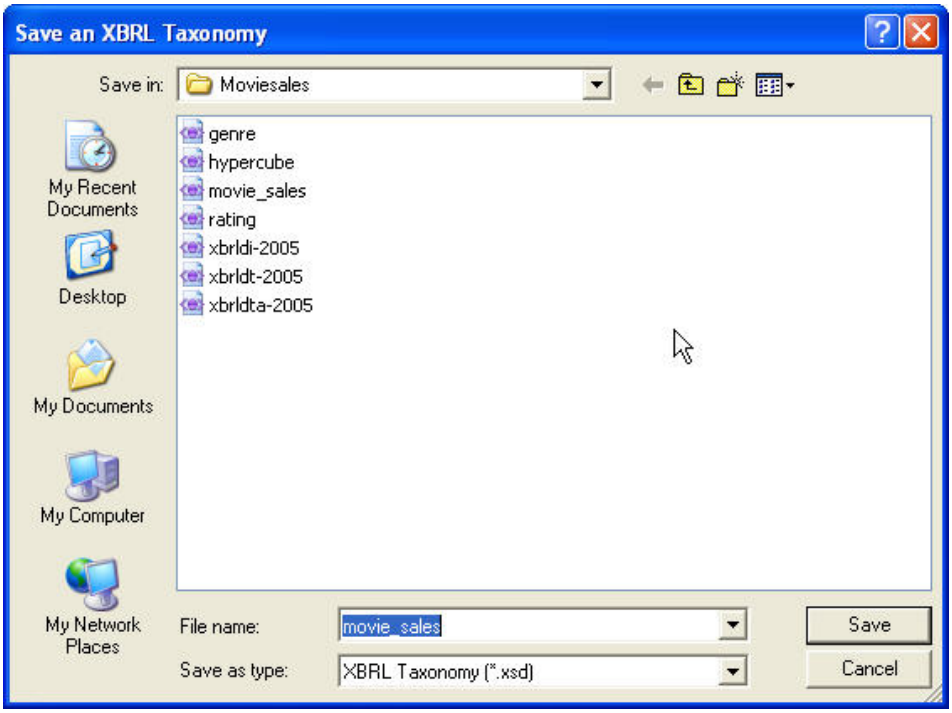
➤ To save a taxonomy to a server:

1 Select **File**, and then **Save To Server**.

2 Enter the **Server URI** in the Server field.

3 Enter the User, Password, and Domain.

4 Click **Log in**.



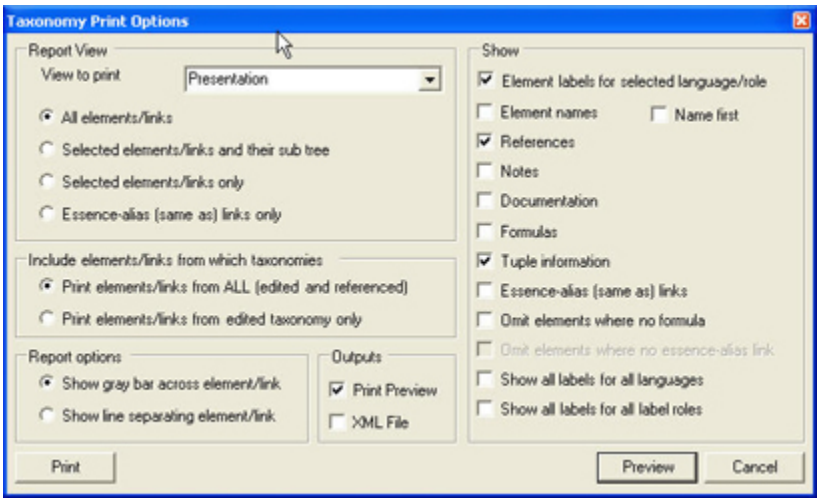
5 **Browse to the folder, and then click Save.**

Printing Taxonomies

➤ To print taxonomies:

1 **Select File, and then Print or Print Preview.**

The Taxonomy Print Options window is displayed.



The XBRL Taxonomy Designer Taxonomy Print Options window enables you to select which reports, elements, notes, and other taxonomy aspects to print.

2 **Select print and output options.**

3 To preview the information, click **Print Preview**.

4 Click **Print**.

Validating Taxonomies

Validating taxonomies ensures consistency with XBRL specifications and basic business rules. For accounting taxonomies, basic accounting rules apply.

➤ To validate taxonomies:

1 Select **Tools**, and then **Validate Taxonomy**.

XML Schema Validation, XBRL Validation, and FRTA Validation are selected by default. You can change the settings.

2 Click **Start**.

When complete, the status and details of the validation are displayed in the Messages pane.

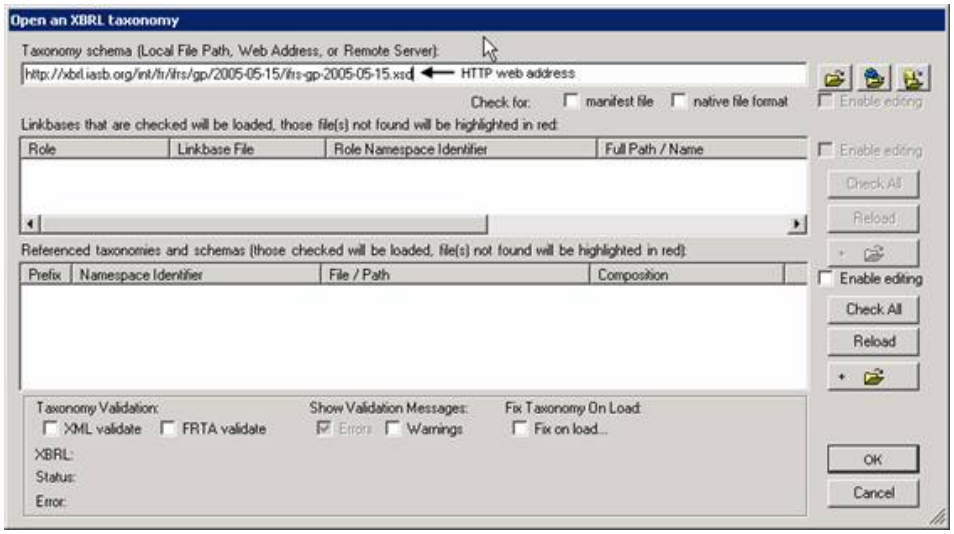
Caching XBRL Taxonomies

Loading XBRL taxonomies such as IFRS, US GAAP, or references to these taxonomies from an HTTP location can take a significant amount of time due to connection speed or taxonomy size. If you enter an HTTP address to an XBRL taxonomy, its respective schemas and linkbases are cached locally. Cached taxonomies are loaded from your local cache of taxonomies rather than from the Web.

➤ To cache XBRL taxonomies;

1 Select **File**, then **Open**, and then **Taxonomy**.

2 Enter the HTTP address and press **Return**.

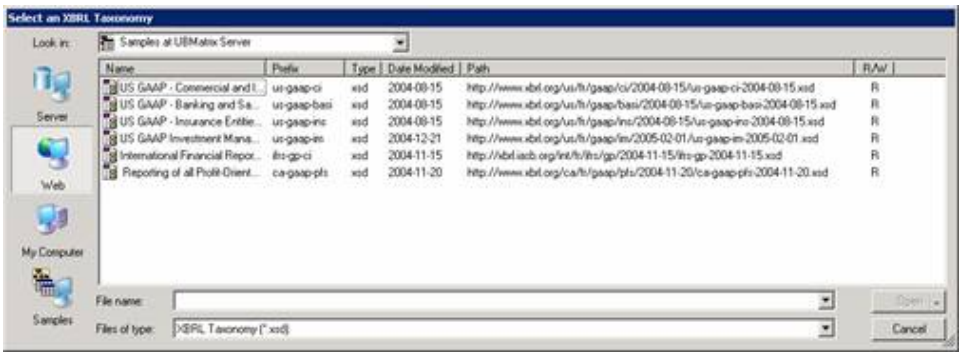


➤ To load Web-based taxonomies using an alternative approach:

1 Select **File**, then **Load**, then **Taxonomy**, and then select the remote or Web-based location  button.

2 Select the **Web** tab.

3 Enter the Web-based address, and then press **Enter** or select **Open**.



The status of each file downloaded from the HTTP address is displayed in the application status bar. The download can be cancelled at any stage.

The default local cache directory is:

```
C:\Program Files\Oracle\Disclosure Management\XBRL Taxonomy Designer\cache
```

Caution! If the download of the taxonomy is interrupted for any reason, the files may be partially cached. Oracle recommends removing these cached files.

When you enter a Web address, the local cache is checked for a taxonomy file with the same name, date and time. If a matching cached file exists, the cached file is loaded instead of the Web file. If the cached file is older than the current Web file, the cached file is replaced with the Web file. Delete the cache files to load the files from the Web location.

To preserve the full HTTP filename within the local cache, these conventions are used:

- ':' changed to ';'
- '/' changed to '!'
- '?' changed to '@'
- '*` <> |' changed to '~'.

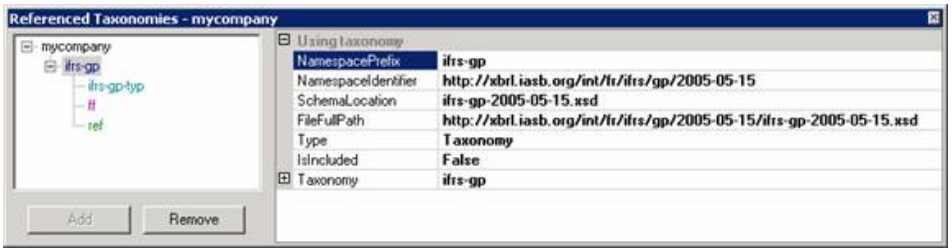
For example, suppose the Web-based file name is:

```
http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/ifrs-gp-2005-05-15.xsd
```

Using the conventions, the local cache file name is:

```
http;!!xbrl.iasb.org!int!fr!ifrs!gp!2004-11-15!ifrs-gp-2004-11-15.xsd
```

Taxonomies can also contain referenced taxonomies imported from a Web-based address. These files are also cached.



If you regularly load taxonomies from a Web location, you can create a Web-based directory listing for them.

About the Properties Grid

This section includes information about:

- Using the Properties Grid
- the Properties Grid pane
- Using the Elements Properties Grid

Using the Properties Grid

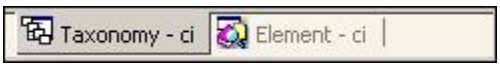
You can edit taxonomy and element concept properties using the XBRL Taxonomy Designer Professional Properties Grid.

The Properties Grid is a dockable pane located by default on the right side of the main screen. Use the tabs at the bottom of the Properties Grid to switch between taxonomy and element properties. See "Adding a Referenced Taxonomy Using the Properties Grid" on page 92 and "Creating a Business Rule Using the Properties Grid" on page 232.

The Properties Grid Pane

You can edit the taxonomy and element concept properties using the Properties Grid pane of the Taxonomy Editor screen.

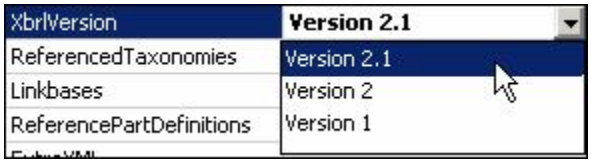
Use the tabs at the bottom of the Properties Grid to select which properties are displayed—taxonomy or element.



To view element properties, an element must be selected in the Relationships View tree or within the Elements List.

Taxonomy properties are displayed for the opened base taxonomy. Element properties are displayed for the selected object selected in the Relationships View tree or the Elements List. Element properties for element concepts are those of the concept, and when an extended link is selected, those of the extended link.

Property choices are displayed on a drop-down list.



Properties with a dialog entry display an ellipsis button ... Click the ellipsis button to display the associated dialog (for example, a referenced taxonomies editor, a linkbases editor, or file selector).

Each property that is a collection or complex object has an editor. These are described in "Collection Editors" on page 126.

Using the collection editor for the Referenced Taxonomies property, you can:

- Edit referenced taxonomies, reference parts definitions, auxiliary schemas, and import files
- Edit the status of the referenced taxonomies.
- Access referenced taxonomy properties, including collections and linkbases
- Change the status of referenced taxonomies to editable or read-only

The Linkbase collection editor organizes the taxonomy labels, references, formulas, and relations into extended links, and allocates them to linkbases for storage in files or the `.xsd` file.

You can edit metadata collections for data types and role collections. Role collections include label, reference, linkbase, extended link, arc, and formula roles. Schema properties specify namespace and schema locations for XBRL and formula definitions.

In the Element Properties Grid, you can change the namespace prefix only for referenced taxonomies that are editable. The name, balance type, period type, and nillable properties are supplemented by collection editors for labels, references, documentation, and parent and child relations.

Using the Element Properties Grid

The figure shows the Element Properties Grid.

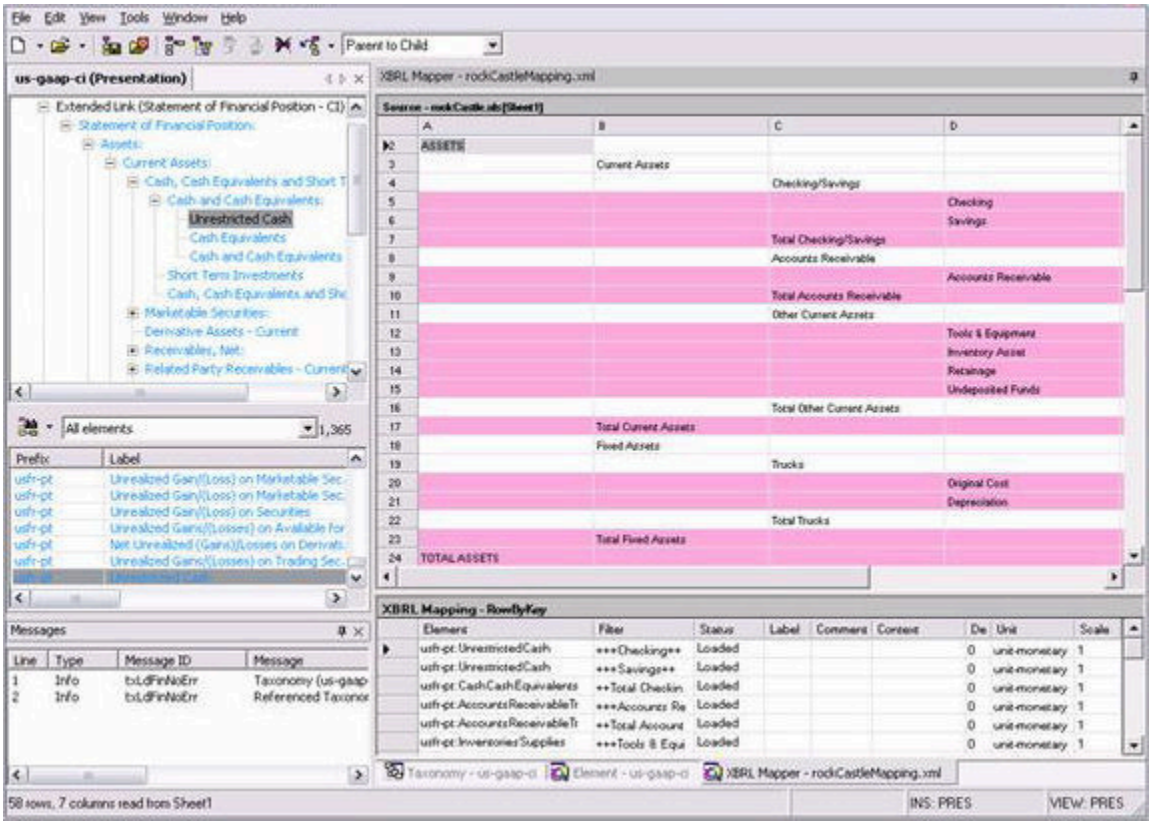


Table 14 Elements Properties Grid Fields

Property	Description
Namespace Prefix	The namespace prefix of the taxonomy that contains this element. For example, “ci” as an abbreviation for USGAAP Commercial and Industrial taxonomy. If the DTS contains multiple editable taxonomies, then this property includes a drop-down list of editable taxonomies to which this element can be moved. Moving an element to another taxonomy moves its labels, references, formulas, and relationships to that taxonomy. If there is only an editable taxonomy, this is a read only field. Using the referenced taxonomies (DTS) collection editor tree view, you can change the edit status of any taxonomy in the DTS.
Name	Element name (for example, CashAndCashEquivalents). See Auto-Naming of Elements, for information about automatically assigning element names based on labels, see . Although this property can be edited, if automatic naming options are active for the taxonomy of the element, any changes to the default label of the element (for the taxonomy’s shown language and shown role) are reflected by automatic name generation.
Labels	Collection of the element's labels. For example, Cash and Cash Equivalents for the English label and the Standard label role. Selecting the associated ellipsis button activates the Labels collection editor (see “Collection Editors” on page 126). You can also click the plus box to expand the labels collection in the property grid, and then edit or enter labels. After entering a label’s text and choosing its role, to add a label, double-click the plus or minus expansion box of the Add new property.

Property	Description
Data Type	The selected element's data type (for example, Monetary). Changing the data type redisplays the property grid with the new data type's properties.
Tuple Definition	The Tuple Definition property, which is displayed only for tuple elements. Click the ellipsis button to activate the Tuple Definition collection editor (see "Collection Editors" on page 126).
Balance Type	A monetary element property that can be: Unspecified (blank), Debit, or Credit.
Period Type	A required value for all V2.1 taxonomy elements except tuples. Its value can be Instant or Duration.
Nillable	An optional attribute that you can assign to non-abstract elements. The default value for nillable is False. Change the value to True to indicate that in instance documents, the element can have an attribute of nil="true" and no content.
Documentation	The selected element's documentation collection. The behavior of this property (in terms of collection editor and expandable subproperties) matches that of the Labels property. For XBRL 2.0 and 1.0, each element's documentation (one per element only) is the documentation supplement of the taxonomy concept element included in the `.xsd` file. For XBRL 2.1, each element can contain any number of documentation items, each a label with a role of documentation (instead of label), and specified language and resource role. Roles that are identified as documentation include documentation, guidance, and user-defined roles. An XBRL 2.1 element can contain any number of documentation entries.
References	Citations of standard and literature references for the taxonomy concept. An element can contain any number of references. Each reference has multiple reference parts, based on the linkbase's taxonomy reference parts collection. The behavior of this property in terms of collection editor and expandable subproperties is similar to the Labels property (see "Collection Editors" on page 126).
Formula	Formula assigned to the selected taxonomy element. Note that the formula implementation is an extension to XBRL 2.1, and is one that the consortium is specifying. It is currently one entry, though it may be replaced with a formulas collection in the future as the standard evolves (see "Collection Editors" on page 126).
Selected Parent	The element name of the selected element's parent. Expands to list the editable properties of the relationship to the selected parent.
Substitution Group	This property is defined for every element. By default, it must be an item or a tuple. It can also contain the value of other elements that are derived from items or tuples. An item is the default substitution group for a concept that is reported in an instance document with a value. A tuple groups instance document items.
Parents	Collection of relationships to parent elements assigned to the selected taxonomy element. You can display all relationships, or only the relationships to the selected relationships' view relationships (see "Collection Editors" on page 126).
Children	Collection of relationships to child elements assigned to the selected taxonomy element. You can display all relationships, or only the relationships to the selected relationships' view relationships (see "Collection Editors" on page 126).

Table 15 Elements Properties Grid Fields

Property	Description
Flag	Arbitrary flag you can assign to a taxonomy element. You can search on all flagged elements. When editing a taxonomy, you can use flags to indicate, for example, that an element requires additional work.
Read Only	Indicates whether the selected taxonomy element is read only or can be edited. This property is the same as the taxonomy's read-only setting, which can be changed using the Referenced Taxonomies (DTS) collection editor.
Date Last Changed	Indicates the date and time the element was last changed.
Element ID	Shows the selected element's ID. You cannot change this property.
Appinfo XML	Enables you to paste or read any XML fragment that is added to the AppInfo element as a taxonomy element child.

About Collection Editors

This section includes information about using the:

- Labels Collection Editor
- Referenced Taxonomy Collection Editor
- Formulas Collection Editor
- Documentation Collection Editor
- Tuple Definition Collection Editor
- Relations Collection Editor
- Reference Part Definitions Collection Editor
- Notes Collection Editor
- Display Colors Collection Editor
- XBRL Metadata Collection Editors
- Data Types Collection Editors
- Label Roles Collection Editor
- Reference Roles Collection Editor
- Linkbase Roles Collection Editor
- Extended Linkbase Roles Collection Editor
- Arc Roles Collection Editor
- Language Codes Collection Editor
- Word Substitution Collection Editor
- Word Removal Collection Editor

Using the Labels Collection Editor

An element can have labels associated with it. Labels can be grouped by language code or by label role.

For XBRL Version 2.1, labels and documentation are stored in, and loaded from, label linkbases. However the Labels collection editor only displays roles with type set to label. Labels with type set to documentation are displayed in the documentation collections.

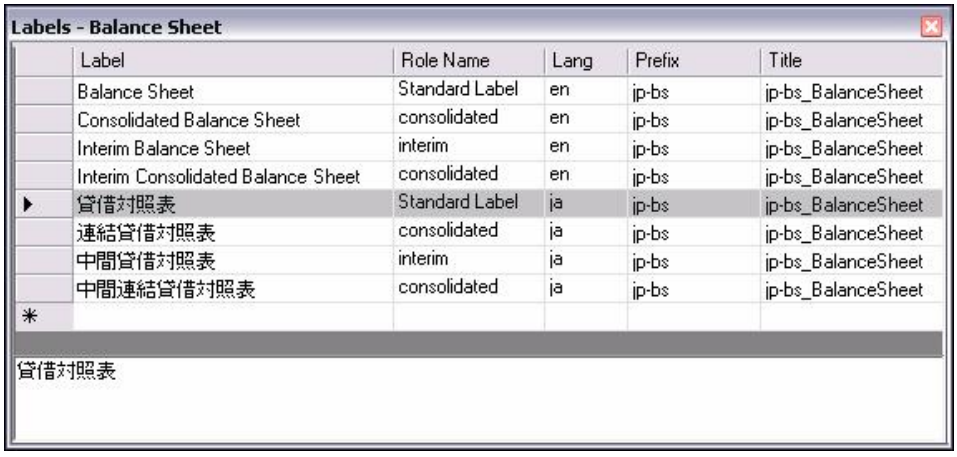
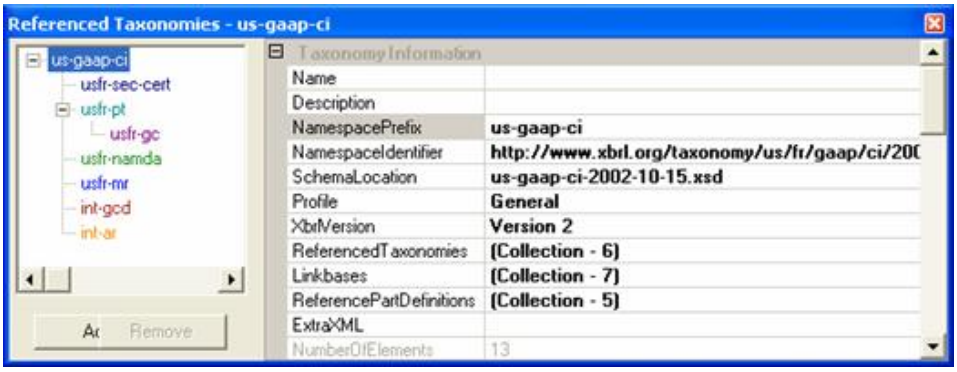


Table 16 Labels Collection Editor Fields

Column	Description
Label Value	The element's label value, which is also displayed in the text box below or to the right of the data grid.
Role	Assigned label role, which is based on the collection of label roles available within this taxonomy (for example, Standard or Verbose).
Language Code	Language code assigned to this label, based on the list of language codes available to this taxonomy (for example, “en” for English).
Prefix	The label taxonomy prefix, which is not the same as the taxonomy of the element being labeled. This field is only displayed when the DTS has multiple taxonomies.
Title	Optional title that you can assign to a label.
Entry text box	A text box for entering label names that are longer than the space provided in the table cell version of the label.

Using the Referenced Taxonomy Collection Editor

The Referenced Taxonomies collection editor shows the base taxonomy’s DTS in a tree view. Use this editor to view and edit the taxonomy properties form of each referenced taxonomy. When the Taxonomy Properties Grid displays an ellipsis button for this editor, select the ellipsis button to activate the editor.



Right-click the taxonomies within the tree view to display the shortcut menu (see Table 17, "Shortcut Menu Options ," on page 59).

Table 17 Shortcut Menu Options

Menu Option	Description
Save	Saves the selected referenced taxonomy. This option is useful for nested referenced taxonomies.
Save as	Saves the referenced taxonomy's `.xsd` file, using the name you enter. This option is useful for nested referenced taxonomies. This option also renames the linkbases of the selected referenced taxonomy that contain the taxonomy name within the linkbase name.
Set Read Only / Enable Editing	Change the taxonomy from read only to editable, or editable to read-only. Note that the `.xsd` file or save-as location must have local disk drive write permission to change the setting. If a taxonomy is set to editable, relationship tree view editing changes are also made in the taxonomy's relationship linkbases, If a taxonomy is set to read-only, relationship changes to the referenced taxonomy insert prohibit arcs in the base taxonomy.
Add	For referenced taxonomies, adds a referenced taxonomy to the selected taxonomy on the DTS tree. The Open Taxonomy dialog is displayed. This dialog includes a check box for enabling editing of the added subreferenced taxonomy. This check box is not available when you open the base taxonomy. For reference parts definitions, adds a taxonomy containing only reference part definitions to the selected taxonomy. The reference part definitions are merged into the collection of the selected taxonomy, and can be used in the selected taxonomy's reference linkbase. For referenced schemas, adds a non-XBRL `.xsd` file as an imported schema of the selected taxonomy.
Remove	Removes the selected taxonomy from the DTS. Do not remove taxonomies that have relationships with other DTS taxonomies' elements (tuples, reference parts, for example), or their linkbases (label, formula, reference, or relationships).

The Referenced Taxonomies collection editor includes a properties grid. After selecting the base taxonomy in the DTS tree, the properties grid displays the taxonomy properties as they are displayed in the Taxonomy Editor. After a referenced taxonomy is selected in the DTS tree, the properties grid displays the referenced taxonomy's properties, including a property that controls the display of the complete set of the referenced taxonomy properties.

Table 18 Referenced Taxonomies Properties Displayed in the DTS Tree

Property	Description
Namespace Prefix	The namespace prefix of the referenced taxonomy within the <schema> (xmlns of the referencing taxonomy). Often, but not necessarily, this is the same prefix as that of the referenced taxonomy.
Namespace Identifier	The namespace identifier that uniquely identifies the referenced taxonomy. Always the same as that of the referenced taxonomy.
Schema Location	The file name for the referenced taxonomy's `.xsd` file, which is generally relative to the referencing taxonomy.
File full path	The full path on the local disk drive to the referenced taxonomy, or the referenced taxonomy's URL. The path used in this DTS is not necessarily the standard location of the published version of the referenced taxonomy.
Type	For taxonomies, this is the usual taxonomy of elements, linkbases, and so on. For part and type definitions, this is a taxonomy of only part or type definitions. Part definitions are merged with the referencing taxonomy's part definitions collection. Contains no elements, relationships or other linkbases, or subreferenced taxonomies. For schema namespaces this is a definition only for xmlns purposes in the referencing taxonomy. For schema imports, this is a non-XBRL `.xsd` file that has an import in the referencing taxonomy. For schema include: (Deferred feature) A non-XBRL .xsd which has an include in the referencing taxonomy.
Taxonomy	Expands to display the selected referenced taxonomy's complete taxonomy properties grid. From this referenced taxonomy's properties, you can access the referenced taxonomy's linkbase collection, type definitions, and so on.

Using the Formulas Collection Editor

Formulas use a proprietary, Disclosure Management format because there is currently no XBRL International specification for defining business rules. Formulas are implemented as a resource within a dedicated linkbase that is similar to the XBRL label or references linkbases.

You can enter formulas for taxonomy elements, and enable users to run calculations (for example, a calculation might test the validity of data for an element, or for an element relative to another element).

Select the formulas from the Elements Properties Grid, to enter a formula.

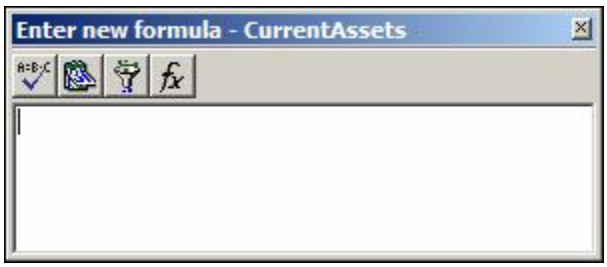


Table 19 Formula Entry Controls

Control	Description
	Checks the syntax of the formula, displaying any parser messages in the message collection, and displaying the decompiled formula as it would be saved in a formula linkbase.
	Enter a named relative context
	Enter a named absolute context
	Select standard functions from a menu
Formula	Type the formula into the text box. You can drag element names from the relationships tree or elements list view to the text box, or you can enter values into the text box. For example, `"ci:TotalAssets = ci:TotalLiabilitiesAndEquity"` indicates that total assets must equal total liabilities plus equity in instance documents.

The formula syntax is similar to Microsoft Excel syntax for formulas. Table 20 lists examples of formula expressions.

Table 20 Formula Expression Examples

Formula	Description
ci:ElementA = ci:ElementB	A Boolean result that is true if ElementA equals ElementB.
ci:ElementA > ci:ElementA[-P1Y]	A Boolean result that is true if ElementA is greater than the corresponding element of the previous year's context.
ci:ElementA < 100000	A Boolean result that is true if ElementA is less than 100000.
ci:ElementA + ci:ElementB	An addition result of the same type as the two terms, using the same units and context as the terms

Using the Documentation Collection Editor

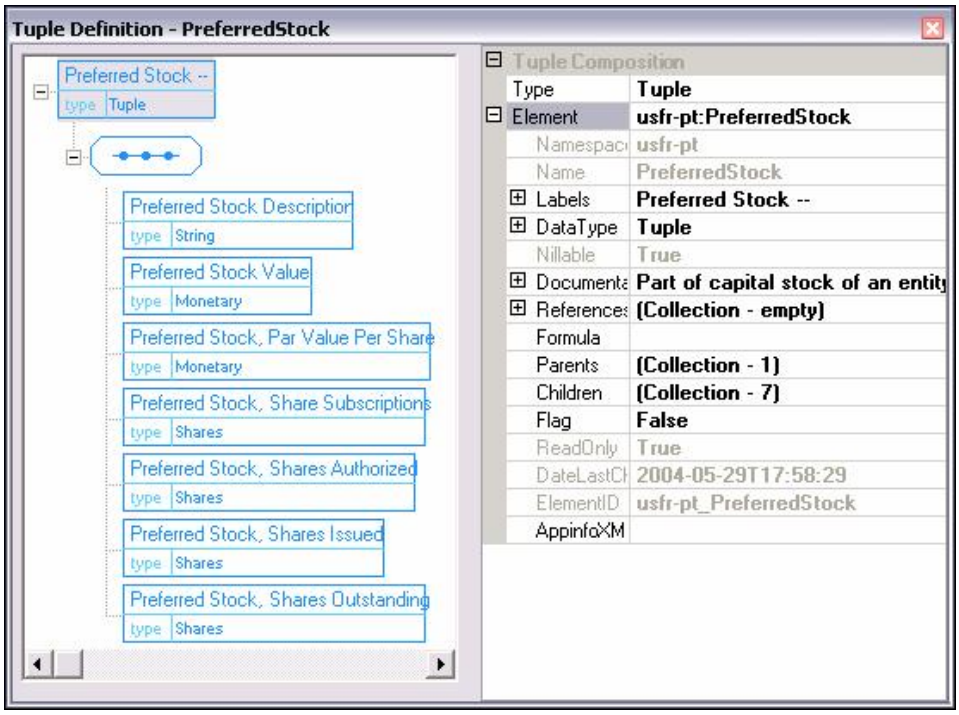
Element documentation explains the selected element. For XBRL V2 taxonomies, the documentation is stored in the <appInfo><documentation> element of the XML Schema file of the taxonomy. For XBRL Version 2.1, this documentation is stored in the label linkbase with a role that you specified. Roles with type set to "documentation" are shown in the documentation collections; roles with type set to "label" are shown in label collections. Label and documentation types are stored in, and loaded from, label linkbases.

Table 21 Fields in the Documentation Collection Editor

Column	Description
Name	The element to which the comment applies
Role	Label role assigned to this documentation based on the collection of label roles available to this taxonomy (for example, Documentation or Guidance)
Language Code	Language code assigned to this documentation based on the list of language codes available to this taxonomy (for example, “en” for English)
Prefix	The prefix of the taxonomy for which the documentation is defined. This taxonomy may differ from that of the concept element documented. This field is displayed only when the DTS has multiple taxonomies
Title	An optional title assigned to the label. A label title is not required by XBRL
Entry text box	The documentation entry field

Using the Tuple Definition Collection Editor

Tuples are a structure of items in an instance document, defined in XBRL 2.1 as complex data types (definition links in XBRL 2.0). For XBRL 2.1, one XML Schema element is saved in the taxonomy `.xsd` file specifying the tuple structure. Tuples can contain items or other tuples, and can be nested to any depth required.



Use the ellipsis button associated with the Tuple Definition property in the Taxonomy Property Grid to activate the Tuple Definition collection editor. In XBRL 2.0, the tuple is defined by the definition relations view. For XBRL 2.1, it is synchronized with the presentation relations view. When you change an element to data type tuple, its presentation relationship children are interpreted to be the tuple elements, and not the child elements. When editing the presentation

relations tree or the tuple definition tree, the changes are synchronized. Each selection in the taxonomy definition tree has appurtenant properties displayed in the right property grid. (See "Adding a Referenced Taxonomy Using the Properties Grid" on page 92.)

A shortcut menu is available in the tuple definition tree view.

Table 22 Tuple Definition Tree View Shortcut Menu

Menu Option	Description
Add Child	For sequences, a sequence construct is added as a child of the selected tree element. For choices, a choice construct is added as the child of the selected tree element. For taxonomies, a taxonomy element is created and added as the child of the selected tree element.
Add Sibling	Same as Add Child, but the element is added as a sibling of selected tree element.
Move Node	Moves the selected node up or down in the taxonomy definition tree.
Delete Relationship	The parent relationship is deleted, and the selected element is no longer a member of the tuple's element collection.
Paste	The copied element is added as a child of the selected tuple section.
Refresh	The tuple tree view is redrawn.

When you select the sequence type within a tuple definition tree view, the property types for tuples are:

- **Sequence**—Elements must appear in order. A sequence construct can contain taxonomy elements, sequence, or choice constructs.
- **Choice**—One element or group of elements are used within a tuple, or a different element or group of elements are used within a tuple. A choice construct can contain taxonomy elements, sequence, or choice constructs.
- **All**—The order of the elements in the tuple is unconstrained. You can specify the All property for tuples only (for example, there is no add-child All option). An all construct can only contain taxonomy elements.

When you select an item that is a taxonomy element within a tuple definition tree view, the options in Table 23 are available:

Table 23 Tuple Definition Tree View Taxonomy Element Item Options

Property	Description
Type	Indicates a tuple child's type. Available types are element and tuple.
Min Occurs	Indicates the minimum number of times the child element or tuple must appear within a tuple when the tuple is used in an instance document. A value of zero indicates the element is optional. A value of one, the default indicates that the element must appear within the tuple.
Max Occurs	Indicates the maximum number of times the child element or tuple can appear within a tuple when the tuple is used in an instance document, or "unbounded"
Element Name	Indicates the element name of the item within a tuple.

Using the Relations Collection Editor

Taxonomy elements contain links to elements within the base taxonomy, or within a referenced taxonomy. The Parent Elements and Child Elements form displays a comprehensive list of all arc-type links. The links displayed depend on the current Relationships View mode. These links can be filtered and expanded to show all links or other types of links, depending on your needs.

Note that resource type links such as labels and references are not displayed on this list, but can be viewed in other ways. The links displayed include: Presentation, Calculation, and the various types of Definition links.

There are properties for relations. The Parents Collection Editor shows a list of elements (parents) from which the selected element is related.

The Children Collection Editor shows a list of elements (children) to which the element is related.

Table 24 Collection Editor Fields

Menu Option	Description
To or From Element	Label of the element linked To (parents) or From (children) the selected element
Relation Type	Indicates whether the relationship is a presentation arc role (P), a calculation arc role (C), or a definition arc role (D). Definition links also display the type of definition arc role. Note that this field has no column heading.
Prohibited	Indicates whether the relation is a prohibiting relation, basically overriding a relation that exists in another taxonomy. **Note:** Relations between elements in one taxonomy cannot be prohibited within that taxonomy.
Required	Indicates whether the relation is required, as defined by the XBRL Specification. False, the default value, indicates the relation is not required.
Order	Indicates the order of the link relative to other links of the parent. The application maintains the value of this option, and it is not editable.
Priority	Indicates the priority of the link, as defined by the XBRL Specification. For example, 0, 1, 2, and so on. The default value is 0.
Weight	Weight is only available for calculation links and indicates the weight assigned to the link. It is used as defined in the XBRL Specification (for example, 1 indicates add, and -1 indicates subtract).
Preferred Label Role	The role used if no role is defined explicitly.
Title	An optional title assigned to the link (for example, “This is my title”).

Using the Reference Part Definitions Collection Editor

The Reference Part Definitions collection editor manages taxonomy reference part definitions. Using this editor, you can edit or reference from other taxonomies, and reference part definitions used in element references.

The Reference Part Definitions collection editor defines the possible citation fields of each reference instance in the taxonomy's references extended link. Reference part definitions identified in this collection are the fields provided for editing the taxonomy's extended link reference. It does not matter which taxonomy contains the element.

These are the fields that are column headings when editing references for the pertinent element:

Figure 5 Reference Part Definitions Collection Editor

Table 25 Collection Editor Members

Option	Description
Name	The reference part name displayed in combo boxes throughout the application (for example, Name or Subparagraph).
XML Tag	The reference part's XML element name (for example, Name or Subparagraph). **Note:** XML element names cannot contain spaces.
Externally Defined	True when the part definition is from a referenced `.xsd` taxonomy of reference part definitions, and not saved in the taxonomy's `.xsd` schema file.
Taxonomy Prefix	The prefix of the referenced taxonomy providing the part definition. The prefix is displayed in reference part collection editor column headers to distinguish ambiguous use of reference part definition local names.
Taxonomy Namespace Identifier	The namespace identifier of the referenced taxonomy providing the part definition.

Using the Notes Collection Editor

Notes provide a method of assigning arbitrary text to an element within a taxonomy. The notes collection is a list of tasks for taxonomy elements. Notes are saved in the native document format taxonomy file.

Figure 6 Notes Collection Editor

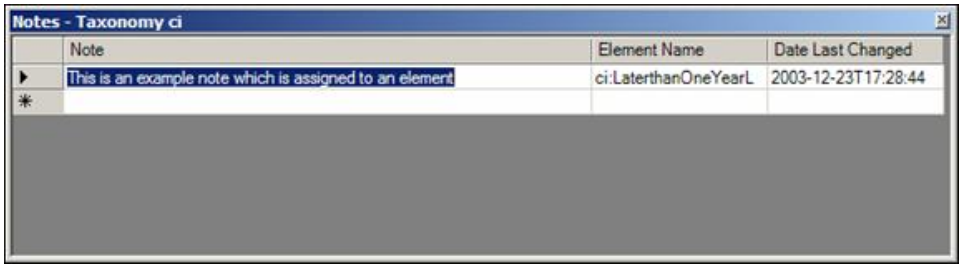


Table 26 Notes Collection Editor Fields

Menu Option	Description
Note	An arbitrary note a user can assign to a taxonomy element.
Element Name	The element name of the taxonomy element to which the note is assigned.
Date Last Changed	The date and time the note was last changed.

➤ To use notes:

1 **Open the Notes Collection Editor, and then enter a note in the Notes field.**

2 **Drag an element from the Relationships View or Elements List to the Element Name field within the collection editor.**

Note: Date Last Changed is assigned automatically.

While viewing the note collection, double-click a note to locate its element in the Relationships View and Elements List. The element is made visible (tree and list view panes scroll as needed) and the element is highlighted in yellow.

Using the Display Colors Collection Editor

The Display Colors collection editor (accessed by selecting Tools, then Options, then Taxonomy Colors property, and then ...) provides the sequence of colors to assign to each reference taxonomy. The colors are assigned in sequence, with each DTS-encountered reference taxonomy assigned the next available color.

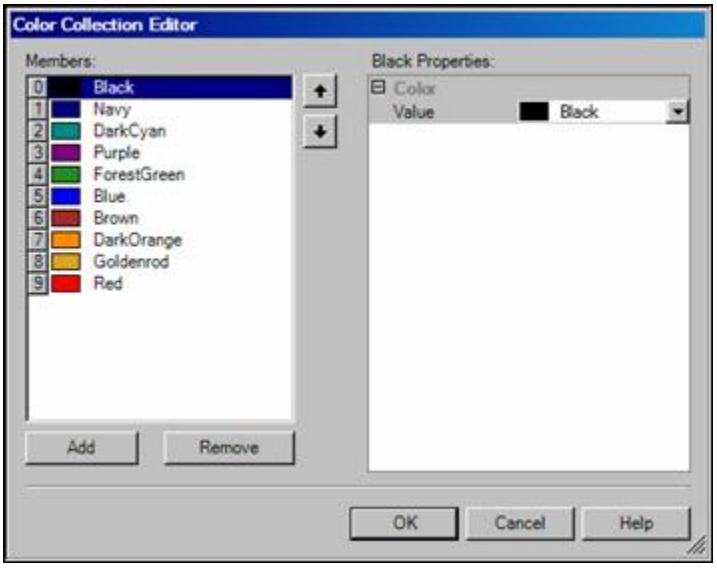


Table 27 Color Collection Editor Fields

Property	Description
Value	Drop-down list that enables you to select a color for the highlighted color sequence number.

Using the XBRL Metadata Collection Editors

The XBRL Metadata collection editors include the Data Types collection editor and the Roles collections editors.

The Data Types collection editor defines built-in and taxonomy extensions to data types usable for taxonomy concept definitions, and in limited cases, for general XML constructs.

Roles collection editors are available for extended links' roles, relationship arcs, labels and documentation, references, and formulas.

Using the Data Types Collection Editor

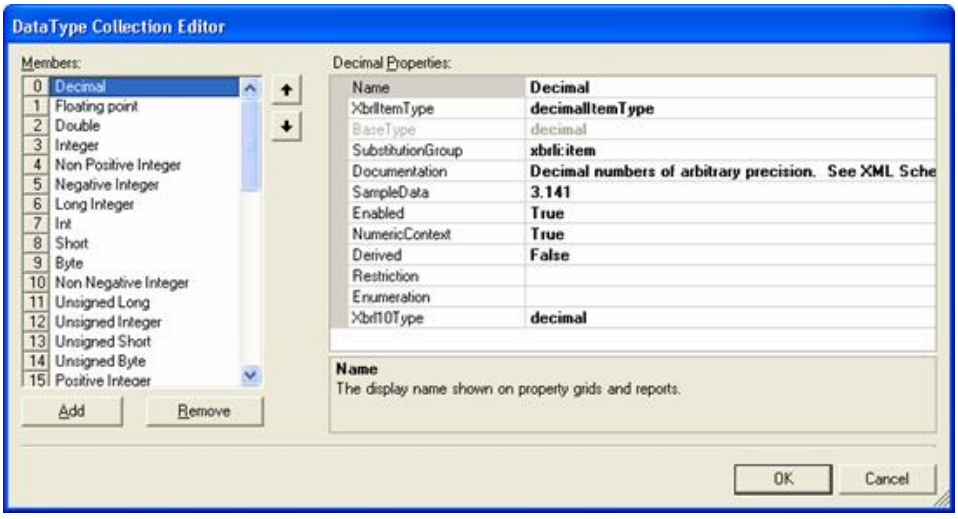


Table 28 Data Types Collection Editor Fields

Property	Description
Name	The date type name (for example, “Decimal”) displayed throughout the application in combo boxes and other places the data type is used.
XBRL Item Type	XBRL International data type name (for example, `decimalItemType`), used in the XBRL core schema files). **Note:** All XBRL data types must be defined by the XBRL core schemas, or derived from a data type defined by the XBRL core schemas used.
Base Type	XML Schema base type from which the XBRL data type is derived (for example, “Decimal”).
Substitution Group	XBRL substitution group to which this data type belongs (for example, `xbrli:item`).
Documentation	Decimal numbers of arbitrary precision. See XML Schema Part 2, Section 3.2.3
Sample Data	Sample data for this data type.
Enabled	Indicates whether this data type is enabled for the current taxonomy.
Numeric Context	Indicates whether this data type is numeric context, which requires a `unitRef` for instance document fact values.
Derived	Indicates whether the data type is a derived data type or a base data type. For example, monetaryItemType is a base data type. A data type based on monetaryItemType that limits the values using XML Schema restrictions is a derived data type.
Restriction	The custom XML restriction for this data type.
Enumeration	Restricted value choices for this data type.

Using the Label Roles Collection Editor

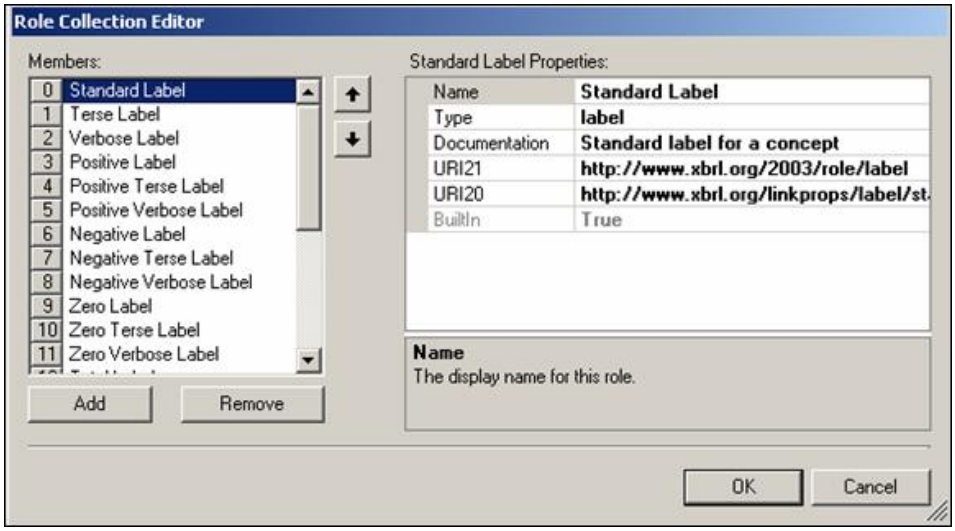


Table 29 Label Role Collection Editor Fields

Option	Description
Type	Indicates the type of label role. In this case, all label roles have the value “label”.

Option	Description
Name	Internal name of the label role (for example, Standard Label).
Description	The label description.
URI21	XBRL International assigned label role for XBRL version 2.1 for this label (for example: http://www.xbrl.org/2003/role/label, the identifier for the standard label role).
URI20	XBRL International assigned label role for XBRL version 2.0. **Note:** Many of these label roles did not exist in XBRL 2.0.
Built In	Indicates whether this is a built in label role.

Using the Reference Roles Collection Editor

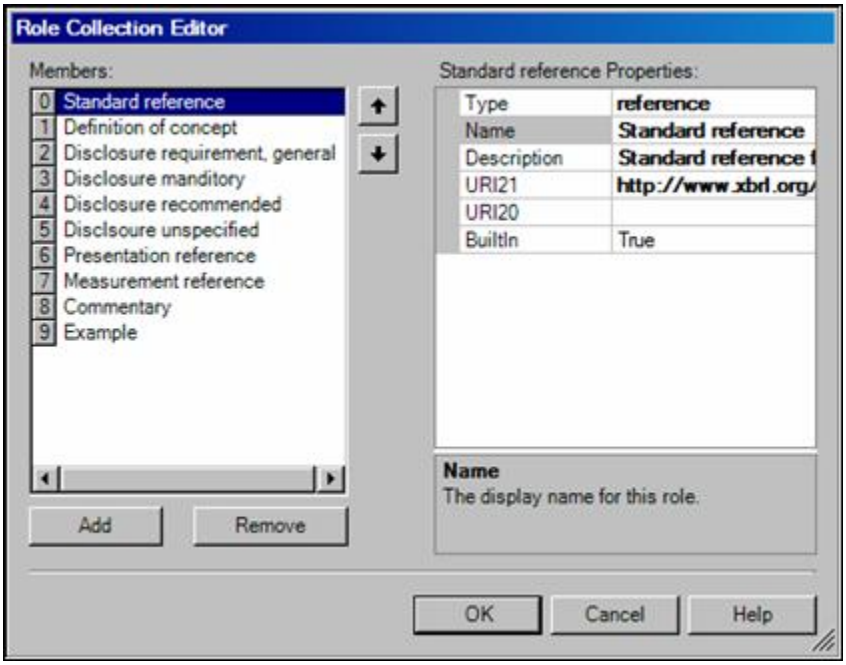


Table 30 Reference Role Collection Editor Fields

Option	Description
Type	Type of role. In this case, the value is always “references” for references.
Name	The reference role name (for example “Standard reference”)
Description	Reference role description.
URI21	XBRL International assigned reference role for XBRL v2.1 for this type of reference (for example, www.xbrl.org/2003/role/reference, the identifier for the standard reference role
URI20	XBRL International assigned reference role for XBRL version 2.0 **Note:** Many of these reference roles do not exist in XBRL 2.0.

Option	Description
Built In	Indicates whether this is a built in reference role

Using the Linkbase Roles Collection Editor

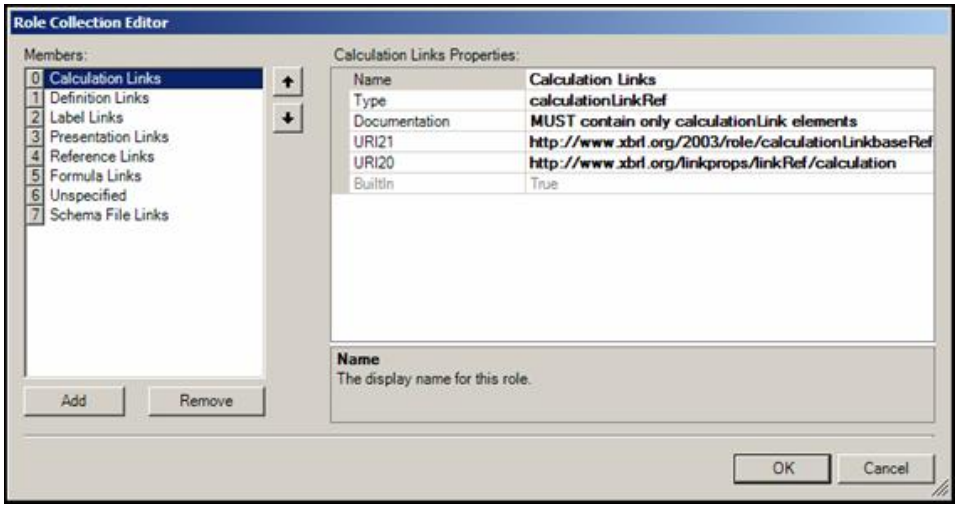


Table 31 Linkbase Role Collection Editor Fields

Option	Description
Name	The internal linkbase type name
Type	The internal linkbase role type
Documentation	Linkbase role description
URI21	XBRL International assigned linkbaseRef XLink:role for XBRL v2.1 for this linkbase type: www.xbrl.org/2003/role/calculationLinkbaseRef (the calculation linkbase identifier)
URI20	XBRL International assigned linkbaseRef role for XBRL version 2.0
Built In	Indicates whether this is a built in linkbase role (read only)

Using the Extended Link Roles Collection Editor

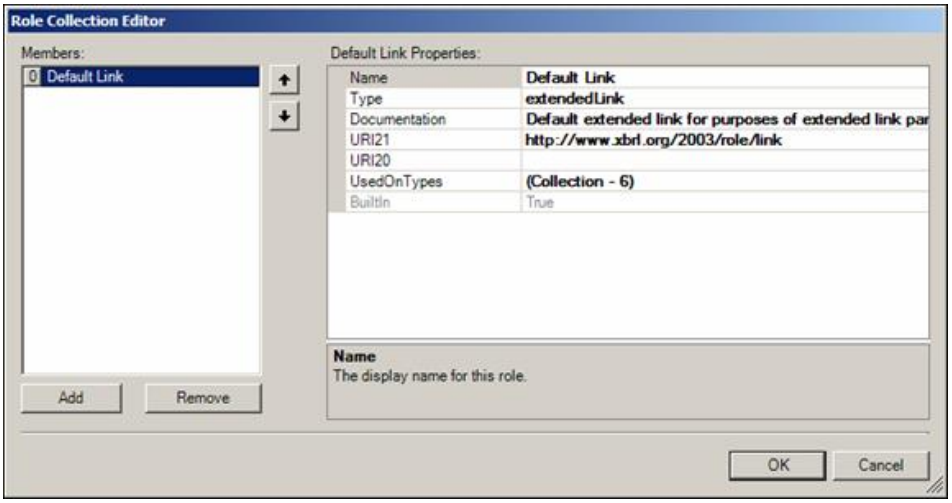


Table 32 Extended Link Roles Collection Editor Fields

Option	Description
Name	Internal extended link role name (for example “Default Link”)
Type	Internal extended link role type (for example, “extendedLink”)
Documentation	Extended link role description
URI21	URI assigned to XBRL 2.1 taxonomies for this extended link role for XBRL v2.1 for this type of extended link, which is the standard extended link: www.xbrl.org/2003/role/link
URI20	XBRL 2.0 taxonomy URI
UsedOn	Lists the linkbaseRef XLink:role types that you can select for this extended link
Built In	Indicates whether this is a built in extended link role (read only)

Using the Arc Roles Collection Editor

The figure below shows the Arc Roles Collection Editor.

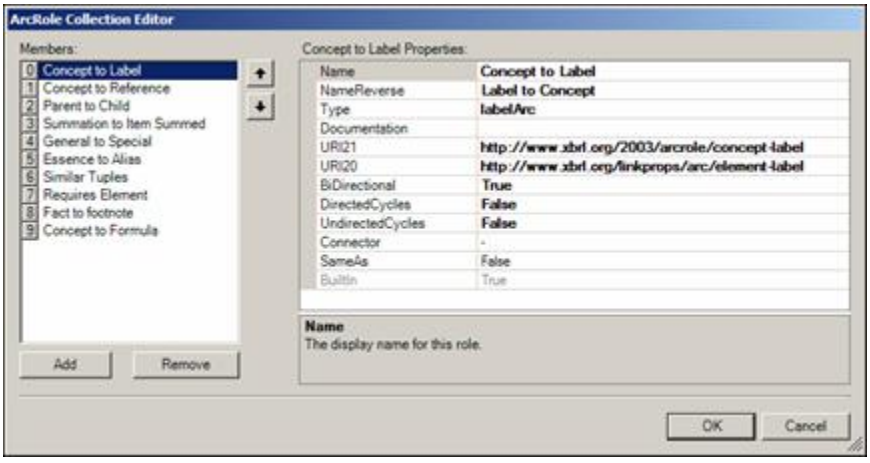


Table 33 Arc Roles Collection Editor Fields

Property	Description
Name	Internal arc role name (for example “Concept to Label”)
Name Reverse	Internal arc role reverse name (for example “Label to Concept”)
Type	Internal arc role type (for example “labelArc”)
Documentation	Arc role description
URI21	URI assigned to XBRL 2.1 taxonomies for this arc role for XBRL v2.1 for this type of arc, which is the standard extended link: www.xbrl.org/2003/role/arcrole/concept-label
URI20	URI assigned to XBRL 2.0 taxonomies
BiDirectional	Indicates whether the arc is bidirectional (for example “True” indicates the arc is bidirectional)
Directed Cycles	Indicates whether directed cycles are allowed for this arc type (for example, “True” indicates that directed cycles are allowed)
Undirected Cycles	Indicates whether undirected cycles are allowed for this arc type (for example, “True” indicates that undirected cycles are allowed)
Connector	Internally used connector between the arc role parts
Same as	Indicates whether the arc is a “same as” type arc
Built In	Indicates if this is a built in extended link role (read only)

Using the Language Codes Collection Editor

The following figure shows the Language Codes Collection Editor (Standard ISO 639 language codes used by XML).

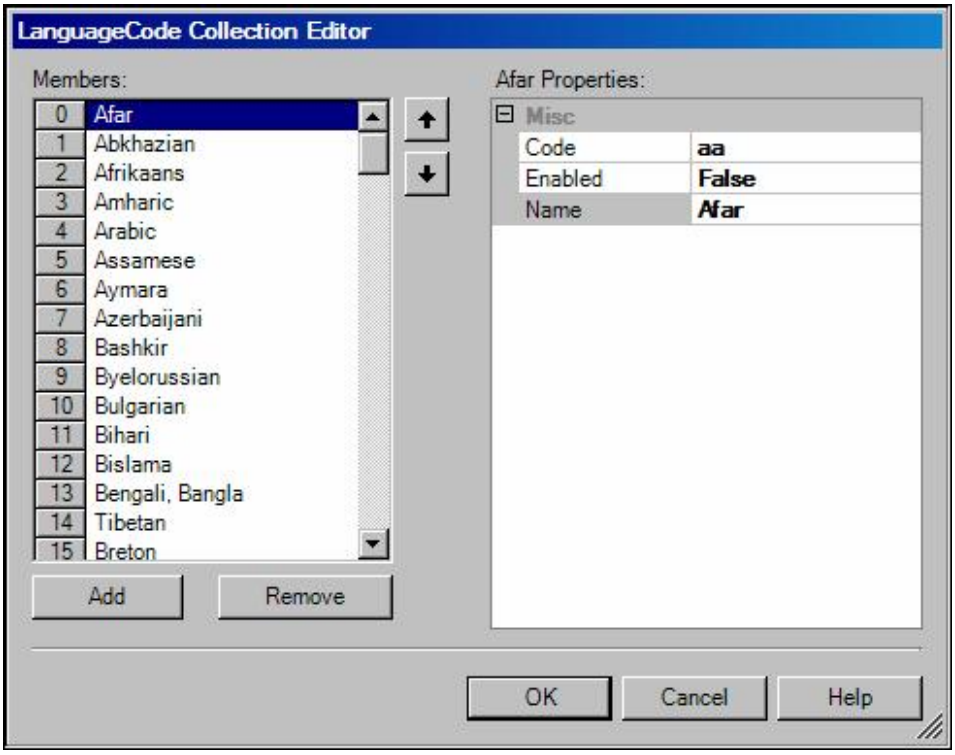


Table 34 Language Code Collection Editor Fields

Property	Description
Code	ISO 639 language code value
Enabled	Indicates whether this language code is enabled for this taxonomy
Name	Language code name

Using the Word Substitution Collection Editor

The figure below shows the Word Substitution Collection Editor.

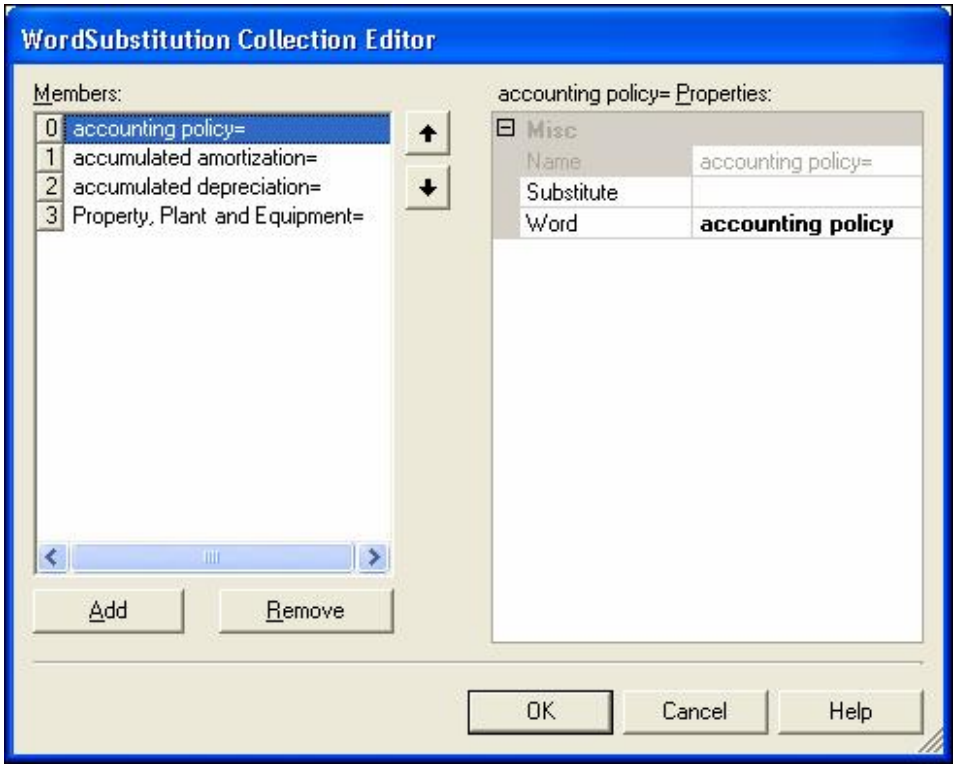


Table 35 Word Substitution Collection Editor Fields

Property	Description
Text	The current taxonomy label text entered in the taxonomy label field (for example, “accounting policy”)
Replace With	Text replacing the current text used in the generated element name. For example, if the current text is “accounting policy”, specify “apy” in this field to replace the current text with the shorter version. In this example, the label “Accounting Policy” changes to “apy” if this option is enabled.

Word Removal Collection Editor

The following figure shows the Word Removal Collection Editor.

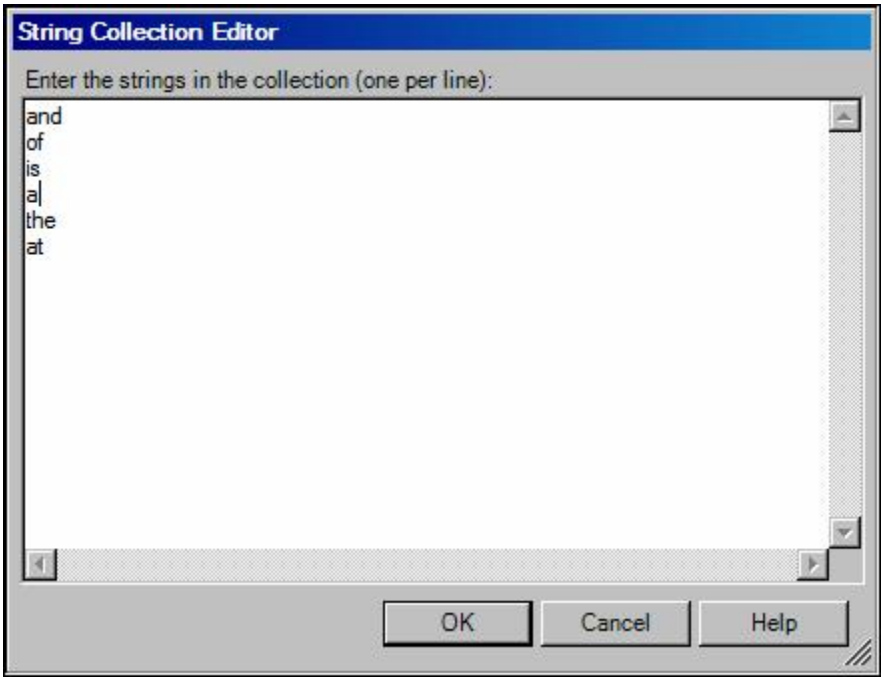


Table 36 Word Removal Collection Editor Fields

Property	Description
WordToRemove	The specified word is removed if found when generating an element name if this option is enabled. For example, entering "and" in this field removes the word from the element label even if users enter "and" as part of the element name ("Property and Equipment" changes to "PropertyEquipment" in the label).

About Elements

This section explains how to work with elements.

Working with Elements

An element is the most basic concept in taxonomy building. XBRL elements are facts or pieces of information described by an XBRL taxonomy. For example, an element with the name Total NonInterestIncome is the USFR-FST taxonomy's XBRL element name for the financial statement disclosure fact for Total Noninterest Income.

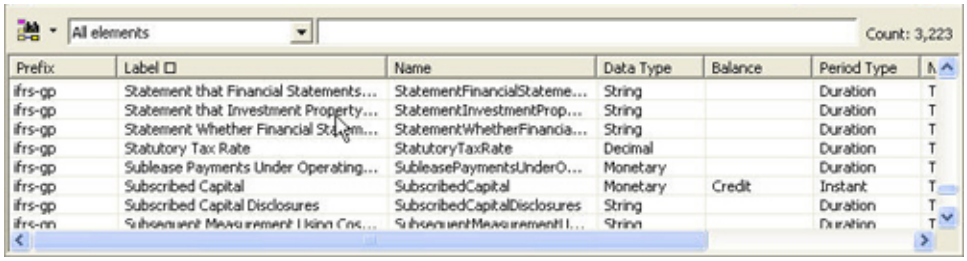
XBRL Taxonomy Designer provides an easy-to-use interface for working with taxonomy elements. The basics of taxonomy building are:

- Adding an element (see "Adding an Element" on page 76)
- Adding child elements (see "Adding Child Elements" on page 78)
- Adding sibling elements (see "Adding Sibling Elements" on page 79)
- Editing taxonomy elements (see "Editing Taxonomy Elements" on page 86)

- Moving elements within a taxonomy (see "Moving Elements Within a Taxonomy" on page 86)

About The Elements List View

The Elements List view is a list of all elements within the base taxonomy and all referenced taxonomies (referred to as the Discovered Taxonomy Set or DTS). An element is listed in the Elements List view only once. Even those elements that do not participate in a relationship are in the element list view. Note the current Count is displayed above the list.



Elements (concepts) are connected by links or relationships. The Relationships View pane shows a tree view of the links or relationships between elements. The Elements List pane shows elements in a dictionary style.

You can add, delete, copy or move:

- An element
- A relationship
- An element and all of its child elements
- A relationship and all of its child relationships

Adding an Element

You add elements using the menu options and toolbar buttons.

➤ To add elements:

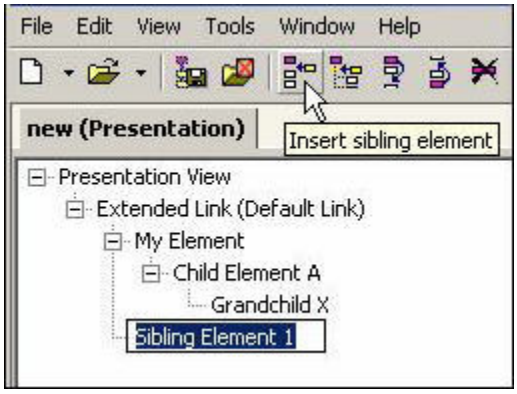
1 In the **Relationships** pane, select **Child Element A**.

2 From **Taxonomy Editor**, select **Edit**, and then **Add Child**.

3 Name the child element.

4 In the **Relationships** pane, select **Sibling Element 1**.

5 In the **Taxonomy Editor** toolbar, click the  **Add Sibling** icon.



6 Name the sibling element.

➤ To use shortcut keys to create elements:

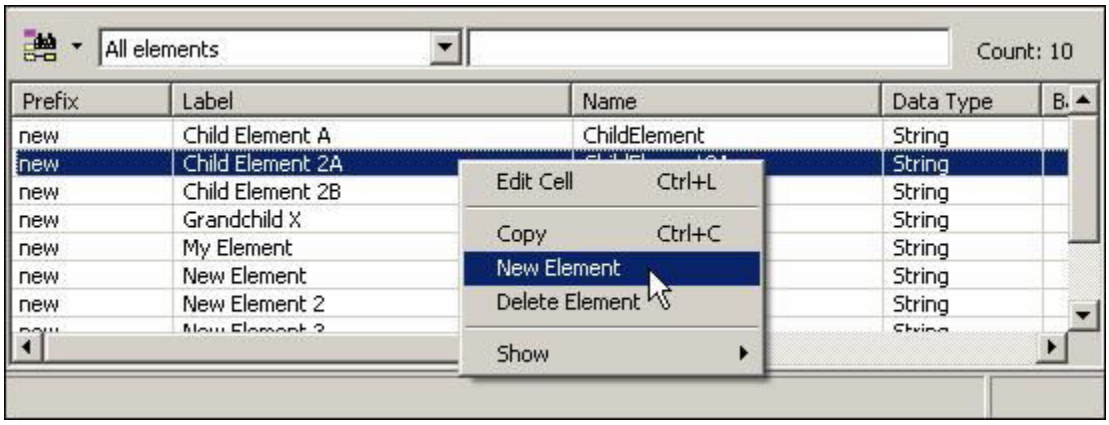
1 Select any element, and then press **Ctrl+N** to create a child element.

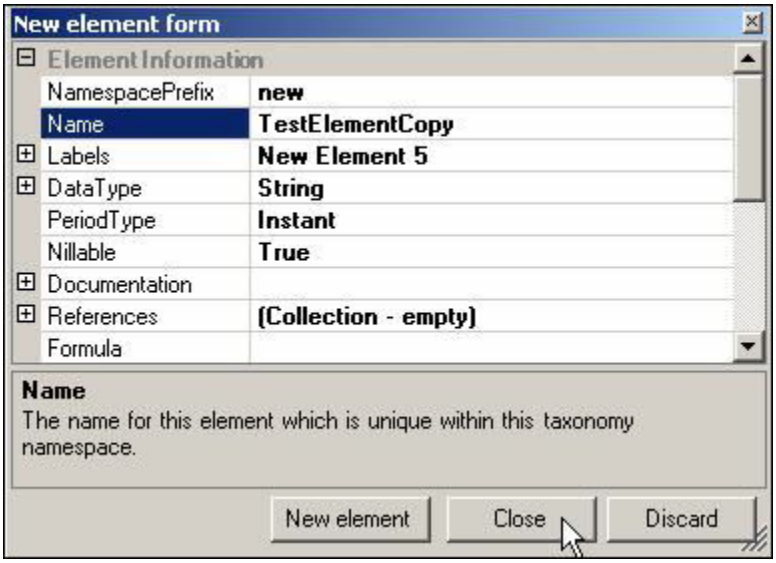
2 Select another element, and then press **Ctrl+I** to create a sibling element.

3 Press **Ctrl+N** several times to create multiple children.

➤ To add an element directly to the Elements List, which is just below the Relationships pane:

1 From anywhere within the **Elements List**, right-click and select **New Element**.





2 **Optional:** Enter an element name.

3 When finished, click **Close**.

4 To form a relationship for the element, hold the left-click button and drag the element from the Elements List to the appropriate location on the relationship tree.

Adding Child Elements

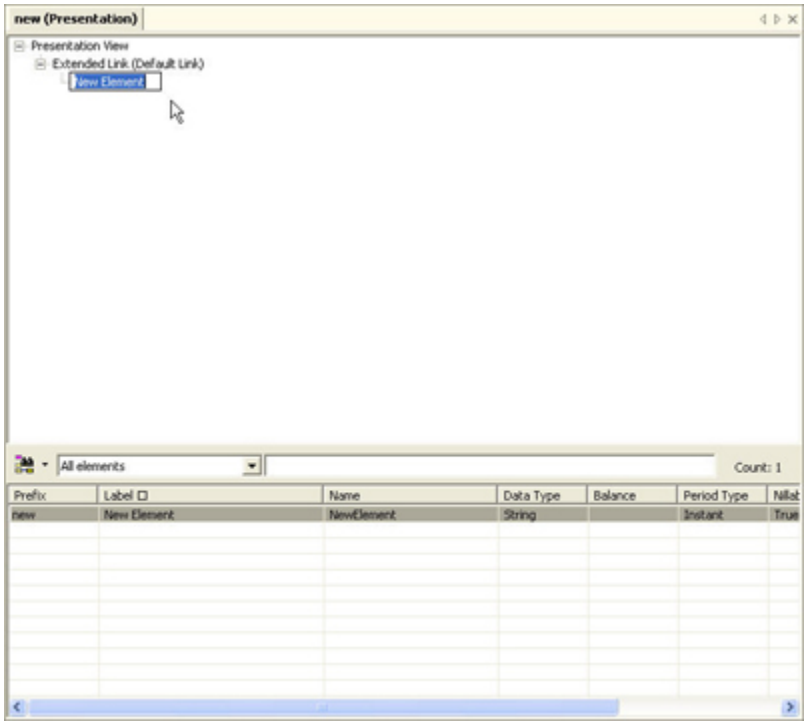
Relationships between elements are important. XBRL allows the creation of parent-child elements. The value of a parent element is the sum total of all its child elements.

➤ To add an child element:

1 Open the taxonomy in the Taxonomy Editor.

2 In the **Relationships** pane, right-click **My Element** and then select **Add Child**.

3 Enter an element name, and then click **Enter**.

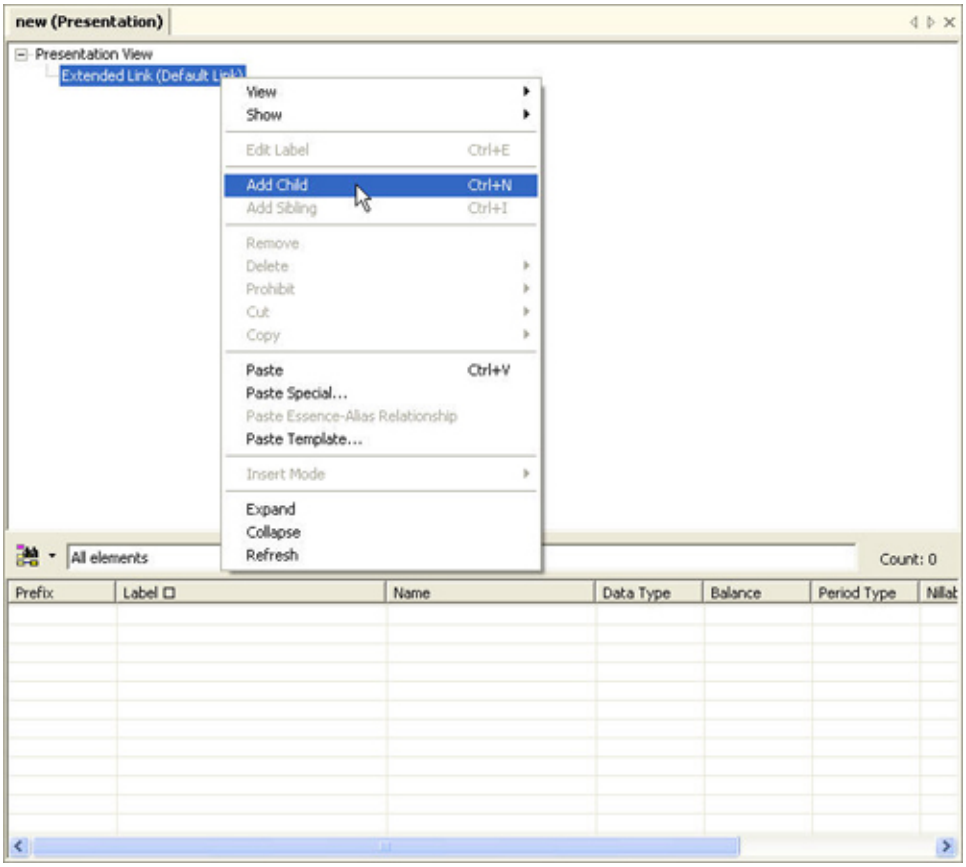


Adding Sibling Elements

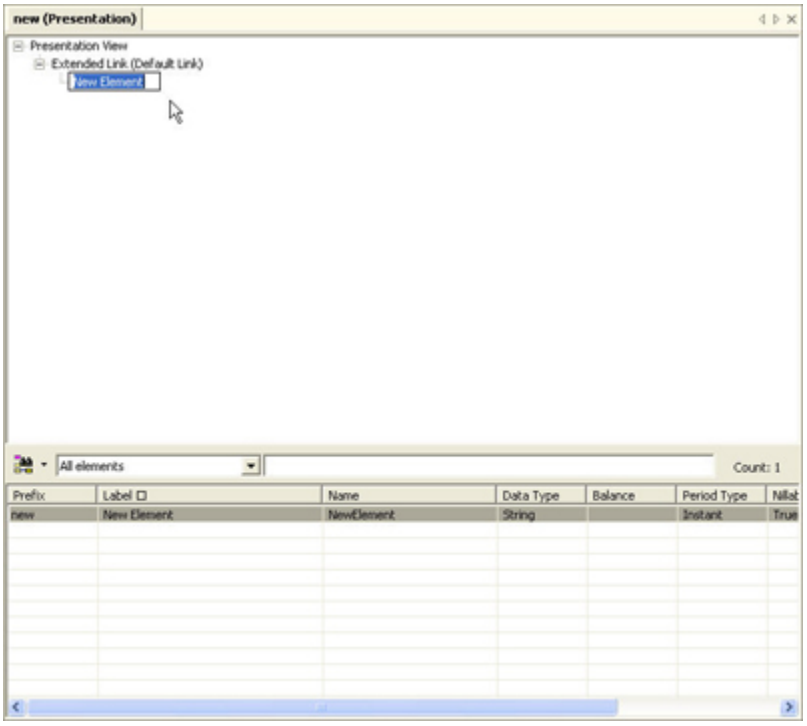
➤ To add a sibling element:

1 Open the taxonomy in the Taxonomy Editor.

2 In the **Relationships** pane, right-click **My Element**, and then select **Add Sibling**.



3 Enter an element name, and then click **Enter**.



4 Right-click **My Element**, and then select **Add Sibling**.

The new element is created at the same level as My Element within the tree view.

5 Name this element **Sibling Element 1**.

➤ To add elements using the menu options and toolbar buttons:

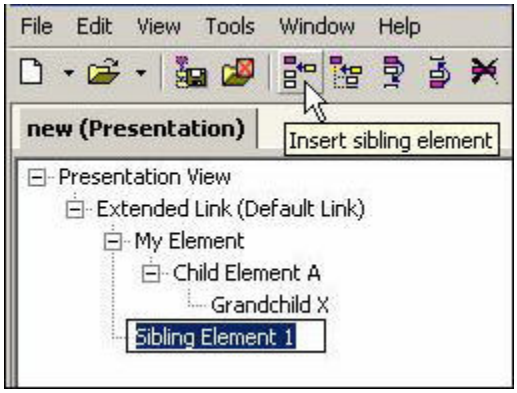
1 In the **Relationships** pane, select **Child Element A**.

2 From the Taxonomy Editor, select **Edit**, and then **Add Child**.

3 Name the child element.

4 In the **Relationships** pane, select **Sibling Element 1**.

5 On the **Taxonomy Editor** toolbar, click the **Add Sibling** icon.



6 Enter a name for the sibling element.

➤ To add elements using the shortcut keys:

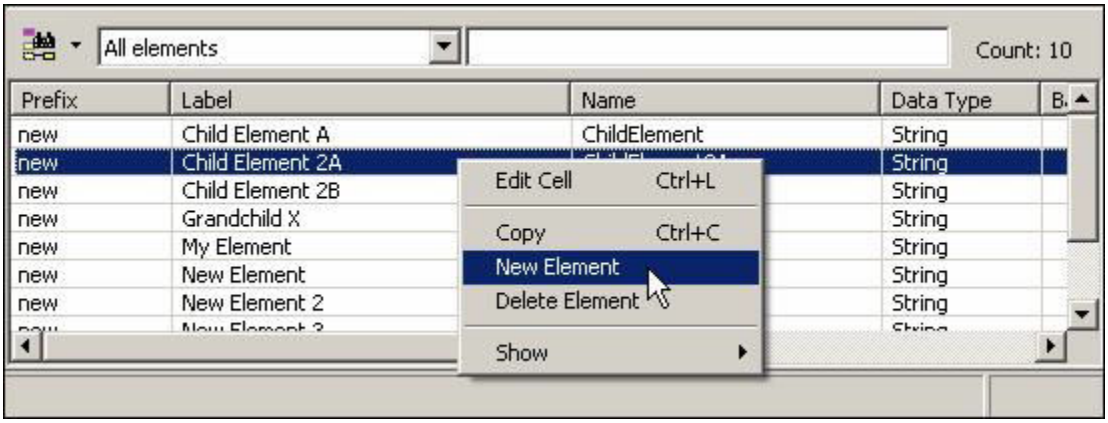
1 Select any element, and then type **Ctrl+N** to add a child element.

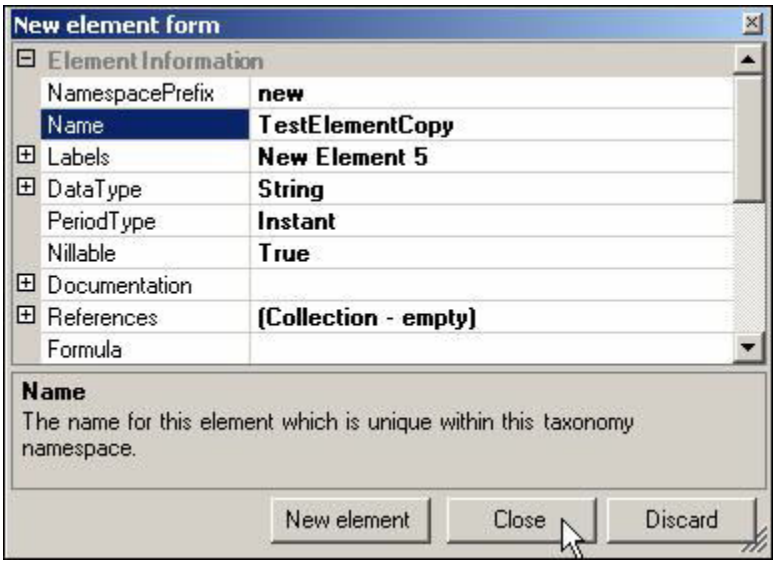
2 Select another element, and then type **Ctrl+I** to add a sibling element.

3 Press **Ctrl+N**, several times to add multiple children.

➤ To add an element directly to the Elements List (located just below the Relationships pane):

1 From anywhere within the **Elements List**, right-click, and then select **New Element**.

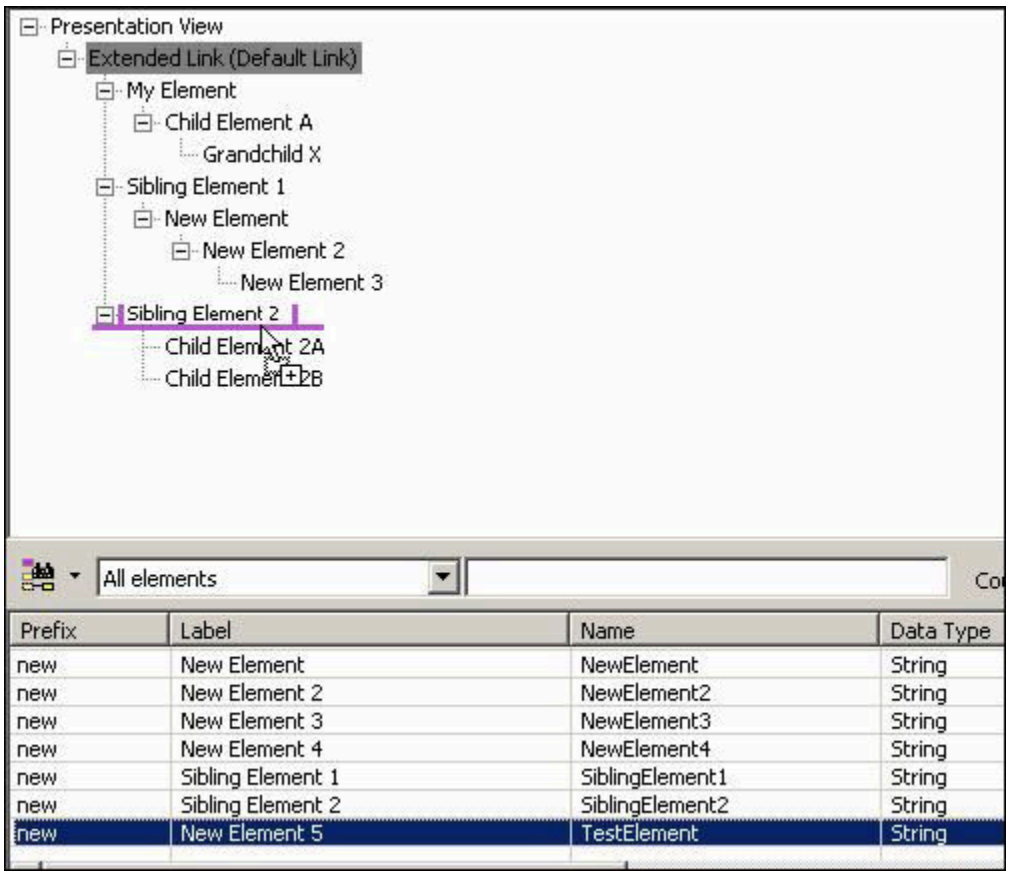




2 **Optional:** Add a name.

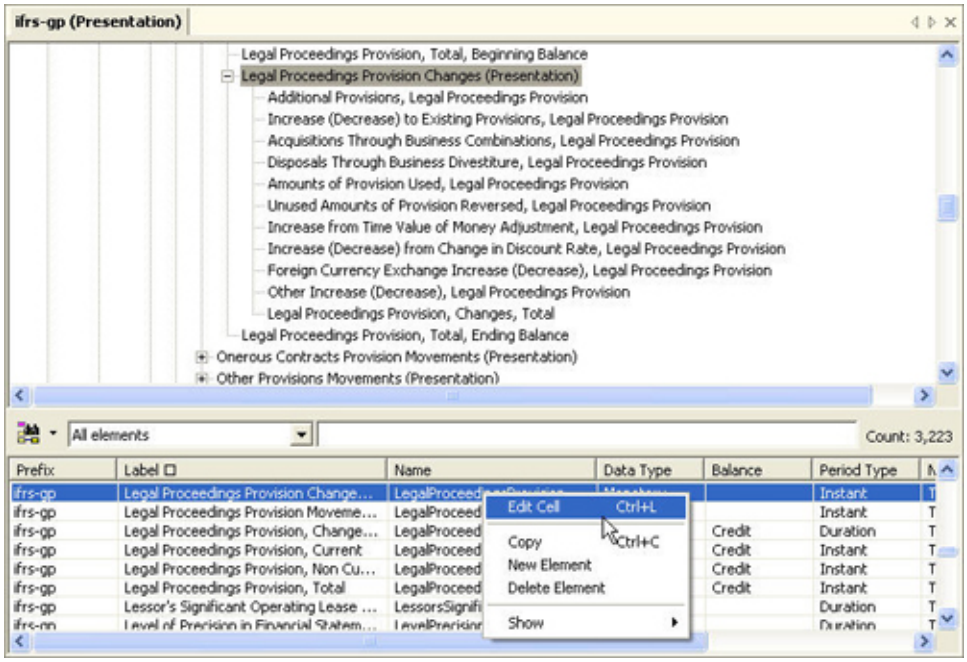
3 Click **Close** to add the element to the Elements list.

4 To form a relationship for the element, drag the element from the **Elements List** to the appropriate location on the relationship tree.



Editing Element Properties Within the Elements List

You can edit element properties in the Elements List view (as well as in the Element Properties Grid). Right-click the element to edit in the Elements List, and then select Edit Cell to begin editing element properties.



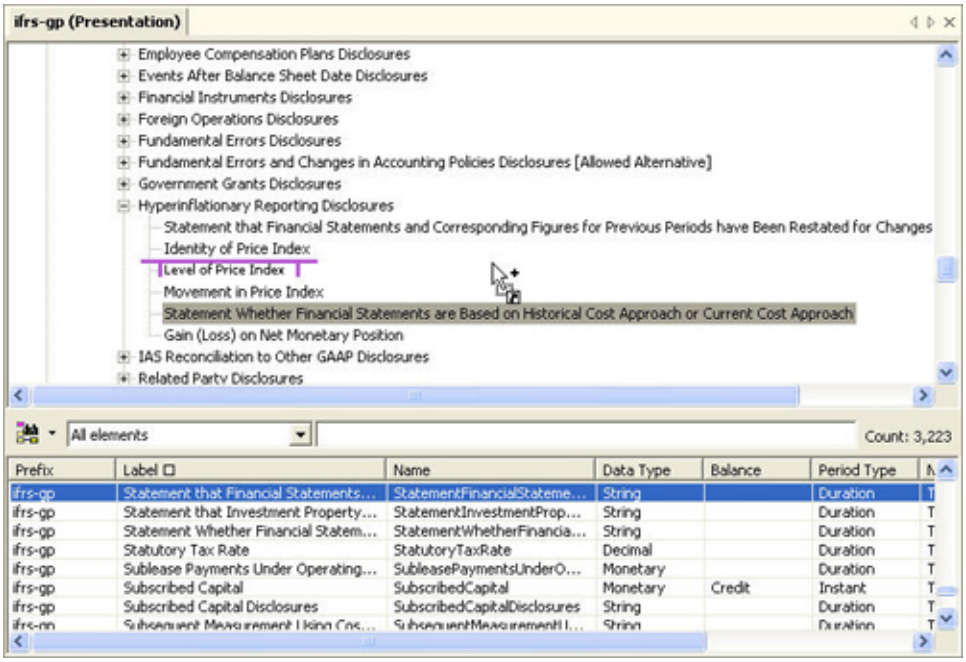
If you right-click a cell in the Elements List, a shortcut menu is displayed. The following table explains the shortcut menu options.

Table 37 Shortcut Menu Options

Option	Description
Edit Cell (Ctrl+L)	Edit element information directly from within the element list view.
Copy Element	Select an element from the element list view, and subsequently paste a link to that element in the relationship view.
New Element	Creates an element without any relations.
Delete Element	Deletes an element from the dictionary and the relations view.
Show	Same as View, and then Show.

Dragging Elements from the Elements List

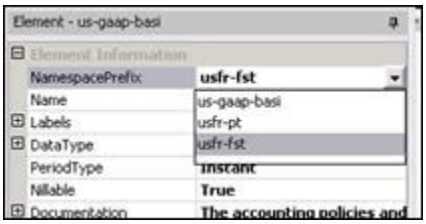
Elements can be dragged from the Elements List pane to the Relationships tree, or copied and pasted. The purple line in the screenshot indicates where you are dragging the element within the taxonomy.



As you drag the element, information is displayed in the status bar indicating where the element is placed if you stop dragging it.

Moving Elements Between Taxonomies

You can move elements between editable taxonomies by changing the element's NamespacePrefix.



Selecting a different prefix moves the element from the original taxonomy's collection of elements to the elements of the newly-selected taxonomy. This also moves any element linkbase entries that were in the original taxonomy's linkbase to the nearest equivalent linkbases of the new taxonomy. Each linkbase entry moved retains its extended link role in the parent taxonomy linkbases.

Selecting new prefixes for elements that are members of a tuple is problematic. Tuples are complex constructs of the owning taxonomy. In general, move all tuple contents to the taxonomy, then select **Tools**, and then **Rebuild Tuple Definition** to be sure the tuple definitions are correctly reconstituted in the new taxonomy.

Editing Taxonomy Elements

➤ To edit the name and label of an element in the Relationships pane:

1. Select any element in the **Relationships** pane or in the **Elements List**.
2. Press **Ctrl+E** to edit the element name and label in the Relationships pane.
3. Change the element label, and then press **Enter** to update the element.

➤ To edit other element aspects:

1. From within the **Elements List**, right-click any cell and select **Edit Cell**.
2. Make any necessary changes, and then click **Enter**.
3. Select another cell within the **Elements List**.
4. Press **Ctrl+L** to begin editing the cell.
5. Make the necessary changes, and then click **Enter**.

Moving Elements Within a Taxonomy

➤ To move elements within the Relationships pane:

1. Select an element in the taxonomy tree view.
2. Press and hold the left mouse button.

   Note how the cursor changes as you move over elements of the taxonomy tree. Also note that the status bar at the bottom of the screen describes the action that would occur if you released the left mouse button at that time.
3. Release the left-click button when done.

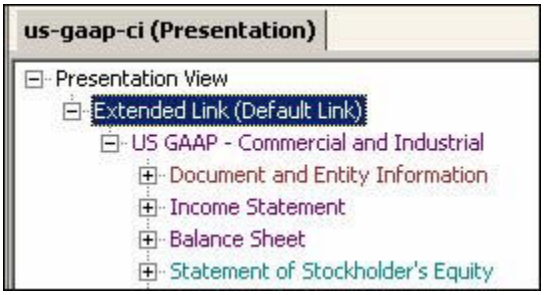
Relationship Views

This section explains how to work with relationship views.

Understanding Relationship Views

Using XBRL Taxonomy Designer, you can add and change relationship views and perspectives using the Relationships pane. This capability enables you to view two portions of a taxonomy simultaneously, or view taxonomy elements using different perspectives, for example, Presentation and Calculation.

Taxonomies can be viewed from three perspectives or views: Presentation, Calculation, and Definition. In the Taxonomy Editor Relationships pane, the view perspective is displayed on the taxonomy tab at the top of the pane.



The following sections about using the relationship views use the US GAAP CI taxonomy as an example. To follow these procedures, open the US GAAP CI taxonomy in the Taxonomy Editor.

➤ To open the US GAAP CI taxonomy:

1 Select **File**, then **Open**, and then **Taxonomy**.

2 From the **Open Taxonomy** window, click the icon.

3 Select the `us-gaap-ci-2004-06-15.xsd` file from the **Open XBRL Taxonomy** window, and then click **Open**.

4 Click **OK** to load the taxonomy and close the Open Taxonomy window.

Changing the Relationships View

➤ To change the taxonomy view:

1 With the `us-gaap-ci` taxonomy file open in the Taxonomy Editor screen, right-click in the Relationships pane.

2 Select **View**.

Three perspectives are listed in the submenu. A checkmark indicates the views currently displayed.

3 Select a view that is not selected.

For this example, select Calculation. Note how the perspective of the taxonomy changes in the Relationships pane.

4 Click to expand the **Extended Link** for the **Calculation view** of the taxonomy.

Adding a View to the Relationships Pane

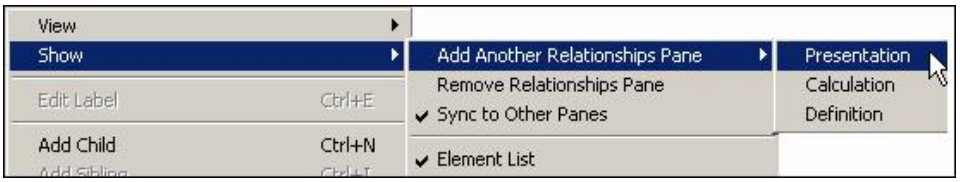
➤ To view the taxonomy from two perspectives simultaneously:

1 Right-click in the **Relationships** pane.

2 From the shortcut menu, select **Show**, and then **Add Another Relationships Pane**.

Note that in this submenu, no check marks are associated with the displayed perspectives.

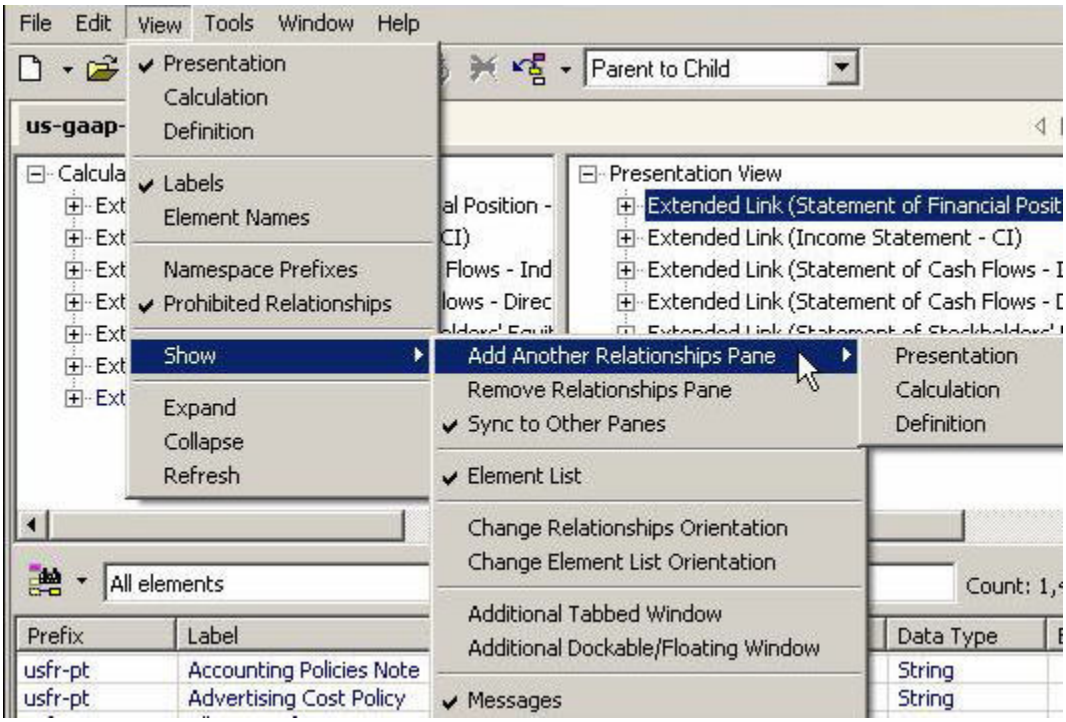
For this example, select Presentation.



A second pane is displayed in Presentation view.

➤ To add another view using the View menu:

1 **Select View, and then Show.**



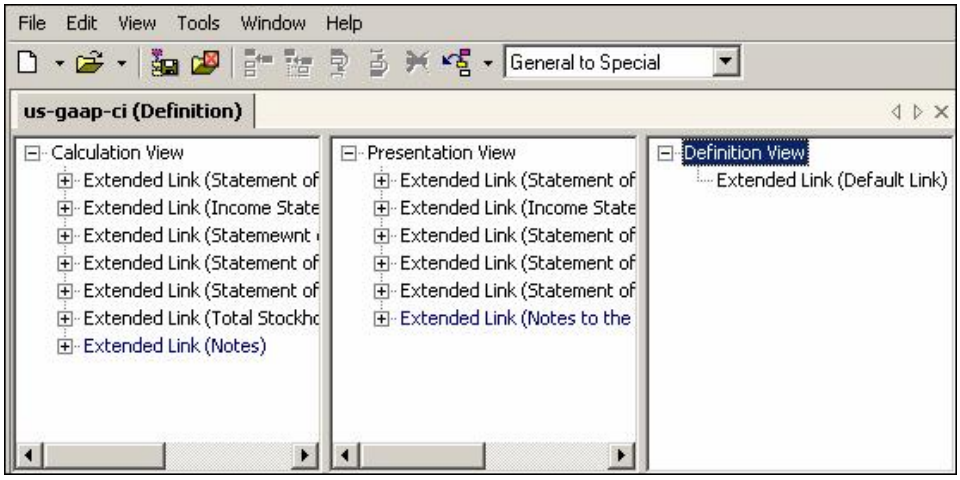
The menu options are similar to those in the shortcut menu.

2 **Select Add Another Relationships Pane, and then Definition.**

There are multiple independent roles for Definition view, including alias-essence and general-special. You can select roles from the toolbar drop-down menu.

Three views are displayed in the Relationships pane.

- Calculation
- Presentation
- Definition



Using the Sync Feature

When working with a taxonomy, you can examine elements of that taxonomy from multiple perspectives (multiple views) simultaneously. Instead of scrolling through each view to find the element, use the XBRL Taxonomy Designer sync feature to automate the process.

To begin have at least two views of the taxonomy in the Relationships pane.

The Relationships View

The relationship view shows relationships (or links) within a taxonomy. In XBRL Taxonomy Designer, you can navigate through taxonomy elements using these relationships. The relationships view can show the presentation, calculation, or definition relations between elements.

The word relationship is the instance of a relation; a relation is a class of such relationships as described by an arc role (such as parent-child, essence-alias).

Each relationship is shown in the relationships view once, but elements may participate in multiple relationships. Elements in multiple relationships can be listed in more than one place within the relationships view (see Figure 7).

Figure 7 Relationships Tree View Pane

The default view is the Presentation view. From the Presentation view, you can switch to a Calculation or Definition relation for the currently selected element. For the definition relation, the default arc role shown is general-special. For example, when Property, Plant and Equipment is selected in the Presentation view, right-clicking and selecting the appropriate relation view enables you to see Calculation links or Definition links in which Property, Plant and Equipment participates.

XBRL Taxonomy Designer opens with the view displayed when it is closed. For example, if you exit the application with a Calculation view displayed, when you restart the application, calculation relationships are displayed.

Referenced Taxonomies

This section explains how to work with referenced taxonomies.

Working with Referenced Taxonomies

A taxonomy can reference a number of other taxonomies. The base taxonomy combined with its referenced taxonomies is referred to as the DTS. After you select a base taxonomy, referenced taxonomies are listed in the Open Taxonomy window and are loaded into the Taxonomy Editor with the base taxonomy. XBRL Taxonomy Designer provides full support for working with an unlimited depth of referenced taxonomies.

In the previous tutorial, a referenced taxonomy was added to demonstrate editing taxonomy properties using the properties grid. See "Adding a Referenced Taxonomy Using the Properties Grid" on page 92.

This section covers these topics:

Load Referenced Taxonomy

➤ To load a referenced taxonomy:

1 Select **Open**, then **File**, then **Load**, and then **Referenced Taxonomy**.

2 Open the **Taxonomy** in the same fashion as you would any other taxonomy.

Viewing Referenced Taxonomies

To see a large taxonomy with multiple reference taxonomies, use the US-GAAP taxonomy.

Editing Referenced Taxonomies

To edit elements or referenced taxonomy structures of the base taxonomy you are editing, select one of the following methods:

- If a user is an extender (consumer) of the referenced (standard) taxonomy, but does not have editing rights for that taxonomy, the editor of the base taxonomy can change the referenced taxonomies' structures by prohibiting relationships between their elements in the referenced taxonomies and patch them with relations that are defined as part of the namespace of the top-level taxonomy.
- If a user is the producer (author) of the referencing and referenced taxonomies, then checking the Enable editing check box to the right of the referenced taxonomies within the Open Taxonomy window enables direct editing of referenced taxonomies.

If a user is the consumer of a standard taxonomy by an extending taxonomy, prohibit arcs block the referenced taxonomy's relationship arcs to prohibit the traversal of a relationship in referenced (standard) taxonomies. Prohibition conceptually puts a "do not traverse" sign on prohibited relationship arcs. The author is free to define a relationship between the elements in the edited top level taxonomy. Prohibited arcs are displayed on the tree view with their labels in a strike-out font. This type of prohibited arc is automatically created when a user moves an element of one of the referenced taxonomies to another place on the relationship tree. You cannot move the element to any relationship view tree.

For example, assume that the author wants to move a subtree in the Document and Entity Information section. The Web site entry needs to be moved under the entity contact information. Because this is not the top-level taxonomy, XBRL Taxonomy Designer cannot change anything in this taxonomy, but can prohibit the old link and create a new link.

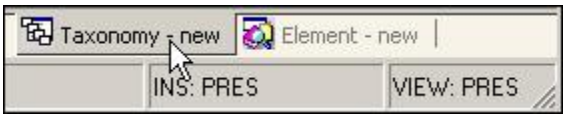
Figure 8 Moving Sub Trees

The displayed structure indicates that the Entity Web Site element is now a child of the Entity Contact element. The strikethrough on the Entity Web Site element indicates that this relationship is prohibited and not traversed. XBRL Taxonomy Designer generates XBRL relationship arcs for prohibition of the original and traversal of the new relationship.

Adding a Referenced Taxonomy Using the Properties Grid

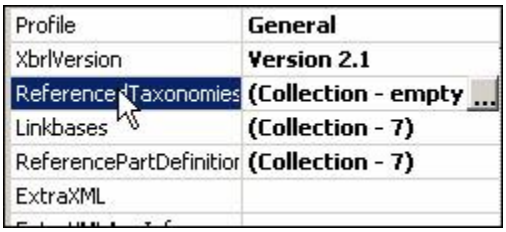
➤ To add a referenced taxonomy using the Properties Grid:

1 **With My Taxonomy loaded in the Taxonomy Editor, ensure that the Taxonomy – new tab is selected at the bottom of the Properties Grid pane.**

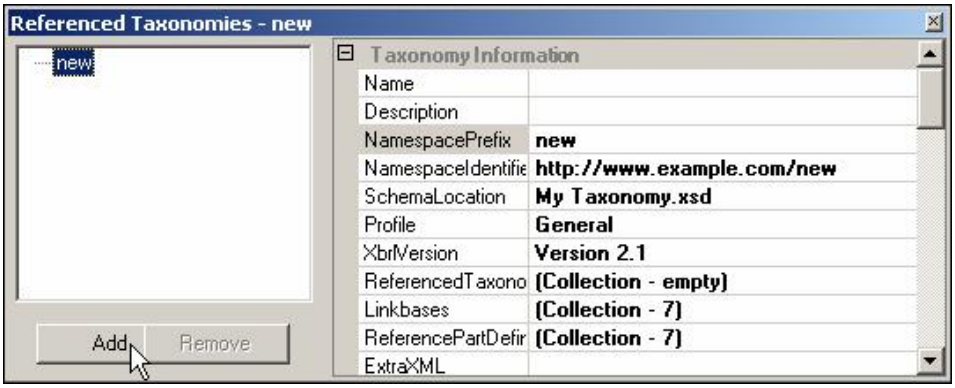


2 **Select Referenced Taxonomies.**

An ellipsis button is displayed in the associated field.



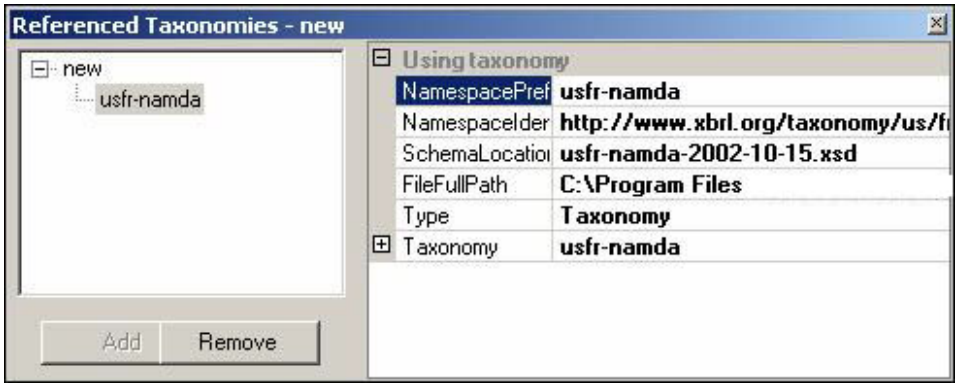
3 **Click the ellipsis button to add a References Taxonomies window.**



4 Click **Add**.

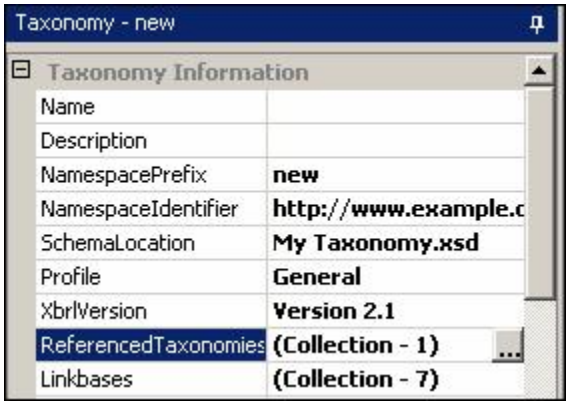
5 Select **Referenced Taxonomies**.

Use what you learned to select and load a referenced taxonomy.



6 Click the **X** to close the window.

In the Taxonomy Properties grid, the Referenced Taxonomy property indicates that there is one referenced taxonomy in the collection.



7 Select **File**, and then **Save**.

8 Click the **X** in the **Relationships View** pane to close the taxonomy.

Editing Referenced Taxonomy Element Properties

Use the Elements Property Grid to edit element properties for a set of taxonomies. You can also edit some properties from the tree view (for example, the default label), and the elements list view when shown. Using the Elements Property Grid, however, gives you control of element property editing, including changing the taxonomy to which an element belongs.

An element belongs to the taxonomy from which it was loaded, or if newly added, it belongs to the base taxonomy.

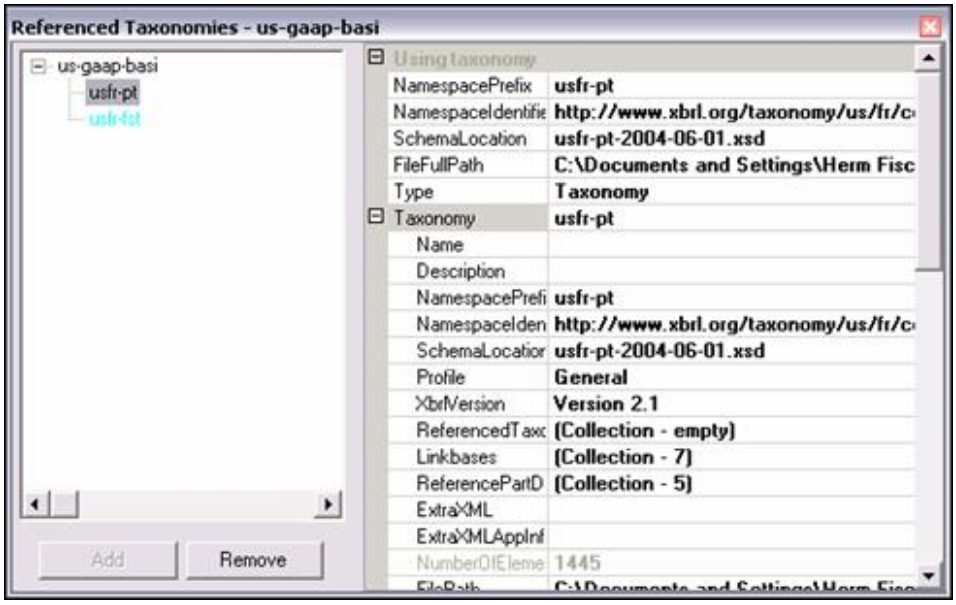
You cannot change the element properties for members of a read-only taxonomy. However, you can add the element labels and relationships of the referenced taxonomy by default in the base taxonomy. Other element attributes, such as data type, cannot be changed.

You can change the element properties in an editable taxonomy. Even if the element belongs to a referenced or DTS discovered taxonomy lower in the DTS tree, its properties can be edited. Changing a label or relationship in an element of an editable, but lower-level, taxonomy changes that element in the lower level schema file and linkbases directly. If the lower level taxonomy is read-only, however, the label and other linkbase changes are made, by default, to the open base taxonomy.

Editing Referenced Taxonomy Properties

DTS Properties are edited using the Referenced Taxonomies Collection Editor. The opened taxonomy properties are displayed in the taxonomy property panel. Click the ellipsis button to activate the Referenced Taxonomies Collection Editor.

Figure 9 Referenced Taxonomies Collection Editor



The Referenced Taxonomies Collection Editor displays a tree that lists all referenced taxonomies in the DTS. The top entry is the base taxonomy. When highlighted in the left panel tree, the opened taxonomy's properties are shown in the right panel. The right panel properties for DTS subtree entries display namespace prefix, identifier, file path, and type, reflecting referencing taxonomy usage.

Though the base taxonomy is an element and linkbase bearing taxonomy, DTS subtree entries may be more limited in usage (for example, schema files containing only reference parts definitions, or schema files containing only non-taxonomy schema definitions). The following table defines the five entry types in this subtree.

Table 38 Current Entries in the Sub Tree

Type	Content
Taxonomy	XBRL taxonomy

Type	Content
Reference Part Definitions	An XBRL taxonomy containing only reference part definitions
Schema Namespace	An xmlns entry in the opened taxonomy schema element
Schema Import	An import (and xmlns) of a non-taxonomy schema
Schema Include	An include (and xmlns) of a non-taxonomy schema

Expand the referenced taxonomies in the DTS subtree to display the internal properties of the taxonomy. These properties are not disabled even if it is a read-only taxonomy, permitting the use of the Linkbases Collection Editor in the taxonomy properties. Edits to a referenced expanded properties of a taxonomy cannot be saved if the taxonomy is read-only.

There are two places where Namespace prefix and filename are defined for subtree taxonomies: in the referenced taxonomy properties, and within the taxonomy. The referenced taxonomy properties such as namespace prefix are those of the referring taxonomy (the xmlns prefix of the higher level taxonomy). The expanded internal taxonomy properties are those stored internal to the referenced taxonomy.

To view linkbase contents and specifications of referenced taxonomies, click the plus-box to expand the internal taxonomy properties, and then select the ellipsis button on the Linkbases Collection Editor.

The DTS tree view includes a shortcut menu for the Referenced Taxonomies. Shortcut menu options enable you to:

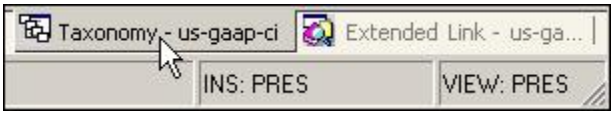
- Change an editable taxonomy. .
- Save a referenced taxonomy.
- Add referenced taxonomies, reference part definition taxonomies, or referenced schema files.
- Remove a referenced taxonomy.

Figure 10 Shortcut Menu

Removing Referenced Taxonomies

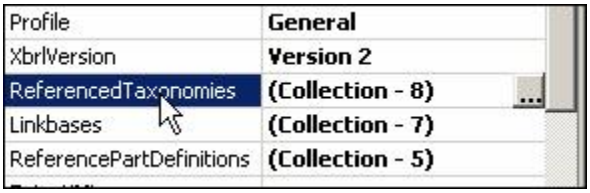
➤ To remove a referenced taxonomy from the base taxonomy:

1 With the **US GAAP CI Taxonomy** loaded in the **Taxonomy Editor**, select the **Taxonomy tab** at the bottom of the **Properties Grid** pane.



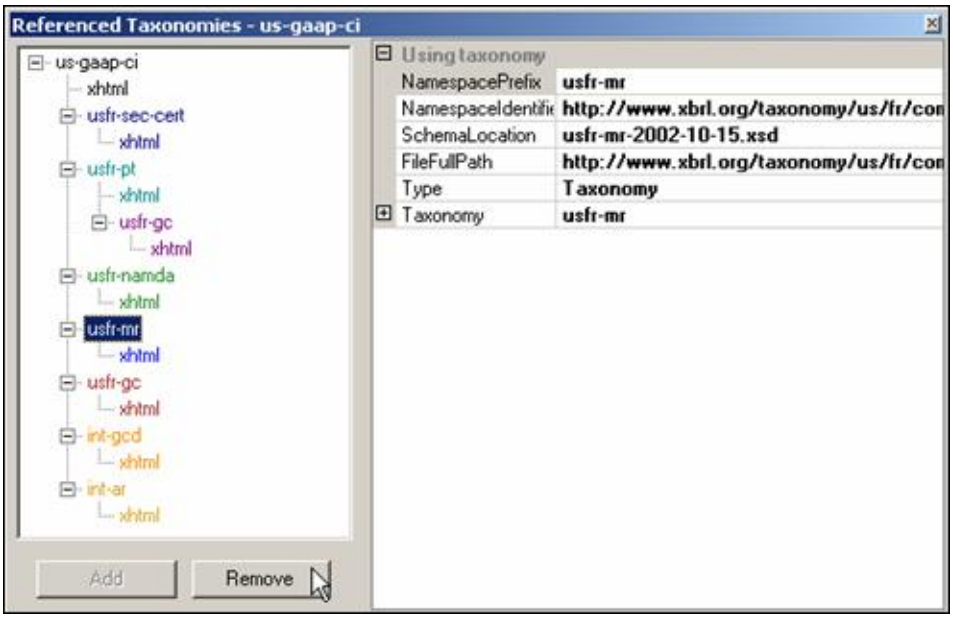
2 Select the **Referenced Taxonomies** property.

An ellipsis button is displayed in the associated field.



3 Click the  button.

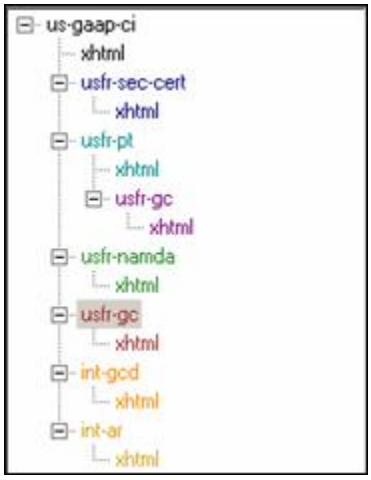
The Referenced Taxonomies window is displayed.



4 In the left tree view pane, select the referenced taxonomy to remove.

5 Click **Remove**.

The referenced taxonomy is removed from the base taxonomy . The left tree view pane in the Referenced Taxonomy window refreshes to display the following list.



6 **Click the X in the upper right of the window to close it.**

In the Taxonomy Properties Grid, note that the number of referenced taxonomies in the collection is reduced by one.

Profile	General
XbrlVersion	Version 2
ReferencedTaxonomies	(Collection - 7)
Linkbases	(Collection - 7)
ReferencePartDefinitions	(Collection - 5)

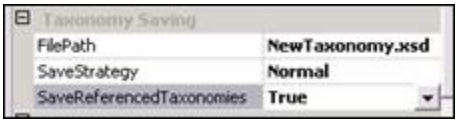
7 **Select File, and then Close to close the US GAAP CI Taxonomy.**

Do not save the changes for the purposes of this tutorial.

8 **Click No to complete this lesson.**

Saving Changes to Edited Referenced Taxonomies

When an opened taxonomy is saved, the default behavior is to save changes to the opened taxonomy. If there are editable referenced taxonomies, each of the editable referenced and DTS discovered taxonomies are also saved.



The base taxonomy SaveReferencedTaxonomies property specifies whether changes to referenced taxonomies are saved. The SaveReferencedTaxonomies setting is useful when you copy a full set of loaded referenced taxonomies (including those that are read-only) into another directory tree, or when downloading a taxonomy set from a Web accessed (http://) source.

Plug-ins

This section explains how to you use the Linkbase Tidier and V2 to V2.1 Conversion plug-ins.

Linkbase Tidier

The Linkbase Tidier plug-in window is shown below.

This tool edits linkbases according to the options described in the following table.

Table 39 Linkbase Tidier Tool Options

Option	Description
Clear titles	Remove titles from linkbase arcs (as editable on the labels, references, formulas, and relationships collection editors). Frequent exchange of linkbases between various external tools can result in excessive use of arc titles.
Reorder	Reorder the relationship order numbers of child collections for linkbase elements. Takes into consideration prohibit relationships below the top level (editable) taxonomy.

V2 to V2.1 Conversion

The following screenshot shows the V2 to V2.1 Conversion plug-in options window.

Figure 11 V2 to V2.1 Conversion Options window

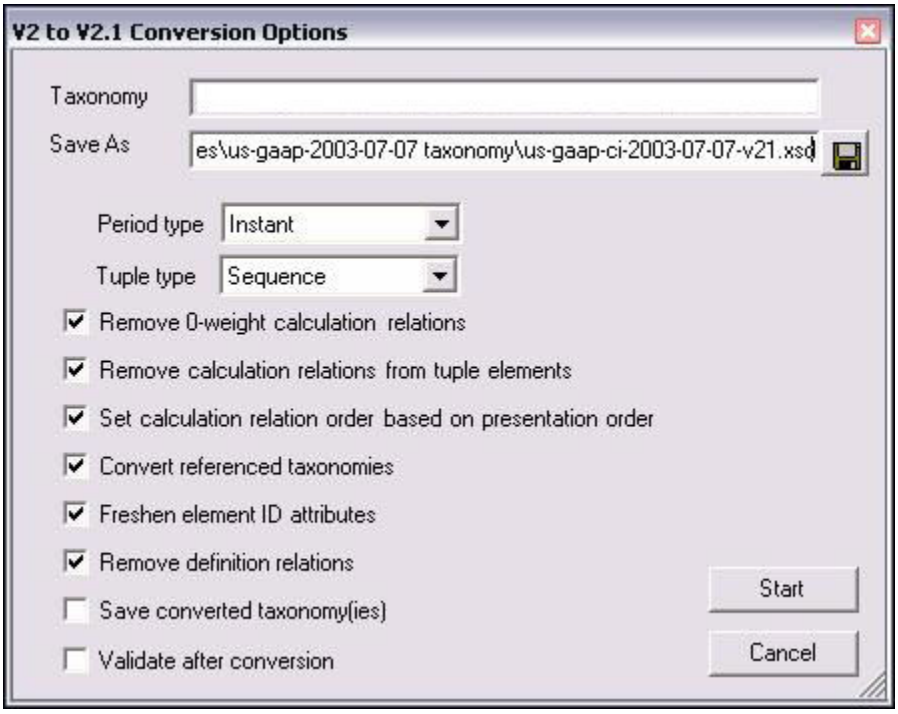


Table 40 V2 to V2.1 Conversion Options

Option	Description
Taxonomy	Indicates the source of the taxonomy that is converted. This option is displayed in gray because the file was read into memory using the taxonomy loading process.
Save As	Indicates the path and file name that replaces the original taxonomy file name after conversion. This options enables you to save the converted taxonomy with another name, or in another directory structure. Changing the file name also changes linkbase file names if any linkbase name contains the same initial characters as the initial characters of the taxonomy file name. Linkbases with names that differ from the file name must be renamed manually. Changing their directory structure must be done manually also.
Period Type	For all elements other than tuples, selects the V2.1 period type to assign during conversion.
Tuple Type	Selects the schema structure type assigned to converted tuples. After conversion, you can edit tuple definitions manually to make structure changes.
Remove 0 weight calculation relations	Zero weight calculation relations are a feature of pre-V2.1 standards, but are not permitted in V2.1. Check this option to remove these relations automatically.
Remove calculation relations from duple elements	V2.0 standard taxonomies often duplicate presentation linkbase structure in calculation linkbases, including tuple compositional links (tuple to tuple item links). This is not necessary in V2.1. Check this option to remove the duplicate structures automatically.
Set calculation order based on presentation order	V2.0 taxonomies only provided order for presentation linkbases. This option copies ordering to calculation links if corresponding parent to child relationships exist in presentation and calculation relationships.
Convert referenced taxonomies	Referenced taxonomies that are pre-V2.1 are converted in addition to the base taxonomy.

Option	Description
Freshen element ID attributes	Element ID attributes can be composed of any character strings, such as random numbers or arbitrary naming. This option provides standard format IDs for all elements.
Remove definition relations	Removes definition relations. When selected, this option erases old-style definitions, which in some cases were arbitrary copies of presentation relationships and no longer apply to V2.1 standard taxonomies.
Save converted taxonomies	Performs a “Save-as”, using the new taxonomy names for the top level and converted referenced taxonomies.
Validate after conversion	Invokes validation after conversion (same as selecting Tools, and then Validation).

Element List Filter

You can narrow the area of the taxonomy that is searched.

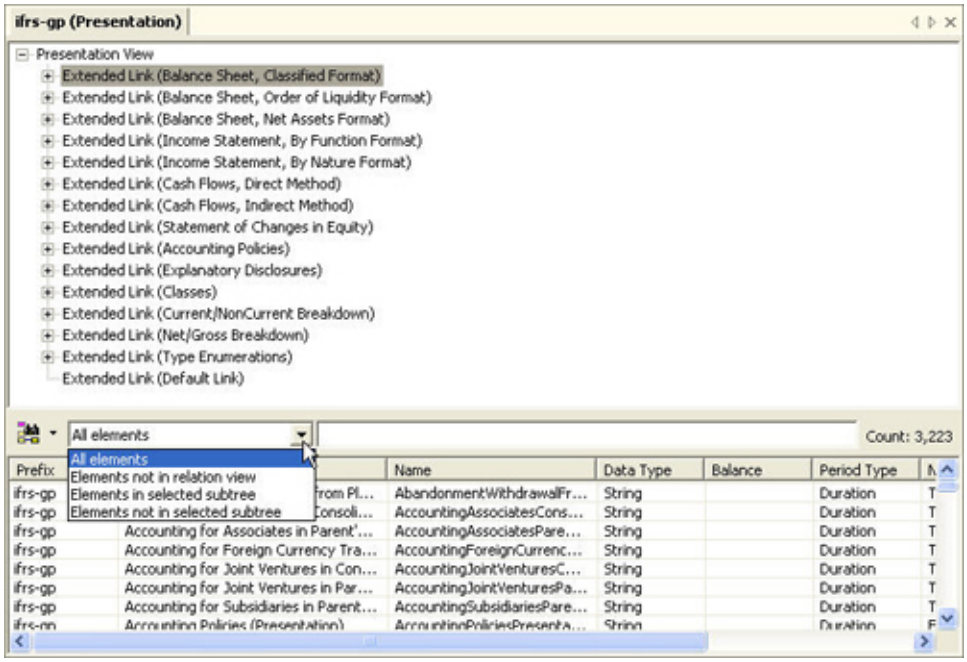


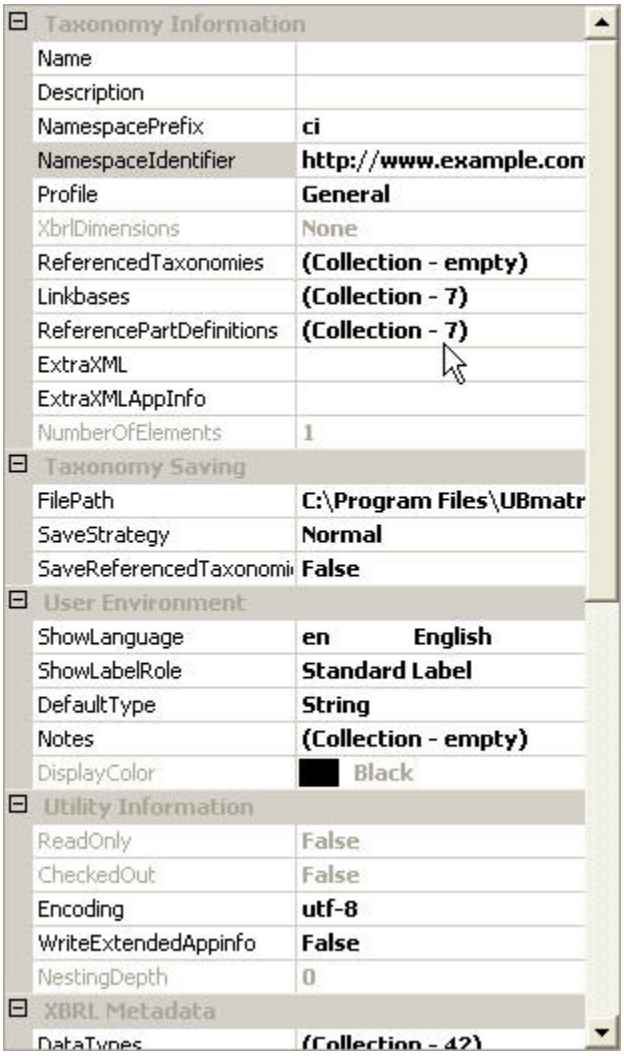
Table 41 Taxonomy Filter Options

Option	Description
All elements	The default, which displays all elements of the DTS in the Elements List pane.
Element not in the relation view	Displays all elements not in the current Relationships view in the Elements List pane. This is helpful when adding elements to the relationships tree view.
Elements in selected subtree	This option displays all elements that are in the selected subtree in the current Relationships view. This option helps you browse the tree view. The element list is sorted in the order of the selected tree view, which is similar to the printout of the taxonomy in a Relationships view.

Option	Description
Elements not in the selected subtree	Displays all elements not in the selected subtree (refer to “Elements in selected subtree” for details).

Taxonomy Properties Grid

The Taxonomy Properties Grid is the grid to the right of the main window. This grid contains a series of fields and collections that enable you to refine-the taxonomy specification.



Taxonomy Information

Table 42 Taxonomy Properties Grid Properties—Taxonomy Information

Property	Description
Name	An optional user-readable taxonomy name (for example “US GAAP Core Standard Taxonomy”). If specified, the name is stored in the local taxonomy directory.
Description	An optional user-readable taxonomy description (for example “US GAAP Taxonomy for Commercial and Industrial Companies to be used by this and that and so on....”). If specified, the description is stored in the taxonomy native file and local taxonomy directory.
Namespace Prefix	Namespace prefix of this taxonomy (for example, “ci”. The prefix is used within the taxonomy “.xsd” file constructs. It does not have to be the same in other places where this taxonomy is referenced (by other taxonomies in a DTS) or used in an instance document. References to this taxonomy and instance documents may choose different prefixes for this taxonomy. Required.
Namespace Identifier	Uniquely identifies this taxonomy (also known as the “target namespace”). By convention this is a phantom URL (no Web location need exist, only the character string must be the same wherever this taxonomy is referenced.) (Required) For example, http://www.example.com/Taxonomy/2003-01-01.
Schema Location	Physical location of the taxonomy’s .xsd file, usually a URL of a Web page, a file name if the reference to the file is relative, or a path and file name to a local file. (Required) For example, you might enter “NewTaxonomy.xsd” to indicate a relative reference to a file.
Profile	[NOT CURRENTLY IMPLEMENTED] Indicates the profile used by this taxonomy. The profile sets allowed edit parameters. For example, you could allow or disallow the addition of arc roles which is allowed under the “General” profile, but not under the “FRTA” profile. See the discussion of “Profiles” below. For example, “General” for the general XBRL profile or “FRTA” for the Financial Reporting Taxonomy Architecture profile.
XBRL Version	XBRL Version of this taxonomy. (Required) For example, “Version 2.1”, “Version 2.0”, “Version 1.0”
Referenced Taxonomies (DTS)	Tree and properties view of the DTS, the collection of taxonomies referenced by this taxonomy. Selecting this property causes an ellipsis button to appear, which activates the Referenced Taxonomies Collection Editor (see “Collection Editors” on page 126)
Linkbases	Collection of linkbases contained in this taxonomy. The ellipsis button activates the “Linkbases Collection Editor” (see “Collection Editors” on page 126s).
Reference Part Definitions	Collection of reference parts defined in, or for use by, this taxonomy. The ellipsis button activates the “Reference Part Definitions Collection Editor” (see “Collection Editors” on page 126)
Extra XML	XML fragment which is inserted within the <schema> element of the XML Schema file containing this taxonomy. For example, “<attribute name=’id’ />”. On loading a taxonomy, any unrecognized <schema> constructs are displayed in this field; likewise, on saving, they are added to the <schema> of the `.xsd` file. The drop-down button provides a multi-line text box for editing and cut/paste of text for this field.
Extra XML Appinfo	Same as Extra XML, but for the Appinfo construct.
Number of Elements	Indicates the number of elements in the taxonomy. For elements in other taxonomies of the DTS use the Referenced Taxonomies (DTS) Collection Editor.

Taxonomy Saving

Table 43 Taxonomy Properties Grid Properties—Taxonomy Saving

Property	Description
File Path	Full path of the taxonomy `.xsd` file, as a local disk drive file, or URL (http://...). (Required) For example, `c:\My Documents\NewTaxonomy.xsd`. Reset by a Save as operation. The ellipsis button activates a file chooser window, to enter an URL, type the HTTP location directly into the property field. Changing this property may affect the linkbase file path, and any referenced taxonomies and their linkbases when the taxonomy is saved.
Save Strategy	Indicates how the taxonomy information is saved. Options are: ● **Normal**: The `.xsd` file and each linkbase is a file with linkbaseRef's in the `.xsd` identifying each linkbase file. ● **Window Taxonomy**: The `.xsd` file imports a `.xsd` file that contains its taxonomy elements, and linkbase files identified by linkbaseRef.. ● **Linkbases in XSD**: The `.xsd` file contains elements and appinfo-located linkbases.
Save Referenced Taxonomies	Indicates if non-edited reference taxonomies are saved. Useful when renaming taxonomies and referenced taxonomies as a group. If false, only those taxonomies of the DTS which are editable (not read only) are saved.

User Environment

Table 44 Taxonomy Properties Grid Properties—User Environment

Property	Description
Show Language	Selects the label shown in the tree views and elements list view. The selection list includes the current default language code (tools, then options property), and any other language codes for which labels exist in this DTS.
Show Label Role	Selects the resource role of the label which is being displayed in the tree views and elements list view. For example, use "Standard" if the default labels are isplayed or "Terse" if the terse labels are displayed. The selection list includes the default role and any other roles currently in use by labels of this DTS.
Default Type	The default data type for the next element to be created in this taxonomy, for example, "String", or "Monetary".
Notes	A collection of notes contained in this taxonomy. The ellipsis button activates the "Notes Collection Editor" (see "Collection Editors" on page 126)
Display Color	Indicates the color of this taxonomy in the tree views and elements list view (see "Collection Editors" on page 126).

Utility Information

Table 45 Taxonomy Properties Grid Properties—Utility Information

Property	Description
Read Only	Indicates if this taxonomy is read only. "False" indicates that the taxonomy is editable. Changeable in the "Referenced Taxonomies Collection Editor" (by shortcut menu of its DTS tree view).

Property	Description
Write Extended Appinfo	Indicates that supplemental information normally stored in the taxonomy native file is saved within the appinfo section of the taxonomy ".xsd" file, and saved as needed in the appinfo of taxonomy elements.
Nesting Depth	Indicates the nesting depth of this taxonomy in the DTS tree. For example, if this taxonomy is not referenced by other taxonomies, the value is 0. If this is referenced by the "top level" taxonomy, the value is 1. If this is a referenced taxonomy of a referenced taxonomy, the nesting depth is 2 or more.

XBRL Metadata

Table 46 Taxonomy Properties Grid Properties—XBRL Metadata

Property	Description
Data Types	Collection of data types defined by this taxonomy (see "Collection Editors" on page 126). User-defined types is used by other DTS taxonomies by their prefixed name.
Label Roles	Collection of label roles defined by this taxonomy (see "Collection Editors" on page 126).
Reference Roles	Collection of reference roles defined by this taxonomy (see "Collection Editors" on page 126).
Linkbase Roles	Collection of linkbase roles defined for this taxonomy (see "Collection Editors" on page 126).
Extended Link Roles	Collection of extended links in the taxonomy (see "Collection Editors" on page 126).
Arc Roles	Collection of arc roles defined by this taxonomy (see "Collection Editors" on page 126).
XBRL Instance Schema	Indicates the schema location of the XBRL Instance document schema which is referred to in import statements of this taxonomy.
XBRL Linkbase Schema	Indicates the schema location of the XBRL Linkbase document schema which is referred to in import statements of this taxonomy.

XBRL Taxonomy Designer Metadata

Table 47 Taxonomy Properties Grid Properties—XBRL Taxonomy Designer Metadata

Property	Description
Formula Roles	A collection editor for formula roles (see "Collection Editors" on page 126).
Formula Linkbase Schema	Indicates the schema location of the XBRL Taxonomy Designer formula linkbase schema which is referred to by formula linkbases.
Formula Linkbase Namespace	Indicates the namespace identifier of the formula linkbase schema for formula linkbases.

Label to Name Conversion Options

These options are loaded with a taxonomy native file, or copied from Tools, then Options for taxonomies or for loading taxonomies from .xsd files.

When a user enters or modifies a default label (shown language, shown role) for an element in the relations view or in the property grid, the name of the element is generated automatically, if appropriate, (this is the default setting) based on the label entered. You may set several name generation preferences for the application as a whole, or for each taxonomy. These settings enable you to customize the name generation to, for example, capitalize the first letter of the name, drop connector words from the name such as "the" and "a", etc.

In case of languages without space characters between words (such as Kanji) the name generation algorithms may be inapplicable. In these cases it may be best to turn off the feature. This can be done using Tools, then Options, which copies into a newly-created Taxonomy Properties tab.

Table 48 Taxonomy Properties Grid Properties—Label to Name Conversion Options

Property	Description
Auto Update from Label	Specifies automated generation of names of elements when their default language, default role, labels are entered or edited.
Use Camel Case	Specifies that the first letter of each word of the label is capitalized when generating an element name from a label in this taxonomy.
Capitalize First Letter	Specifies that the first letter of an element name is to be capitalized in this taxonomy when generating an element name from a label.
Word Removal	Specifies that the "Word removal list" is used when generating element names from labels.
Word Removal List	Specifies words to be removed from element names generated from labels (see "Collection Editors" on page 126).
Word Substitution	Specifies that the "Word Substitution List" is used when generating element names from labels
Word Substitution List	Specifies text strings and replacement values to be substituted for them in generating names from labels (see "Collection Editors" on page 126).

4 Instance Documents

In This Chapter

Understanding Instance Documents

XBRL instance documents are collections of fact values for concepts defined in XBRL taxonomies. There is a tight relationship between taxonomies and instance documents, and the instance document functionality leverages the robust taxonomy functionality of the application.

When creating instance documents, ensuring data accuracy of the information expressed is also important. XBRL Taxonomy Designer offers validation features to help ensure data accuracy including XML validation and XBRL validation of calculation linkbases, formulas, and business rules.

Basic terms and definitions of associated with the XBRL Taxonomy Designer Instance Document Editor:

- **Instance Documents**—A XBRL Instance document is a collection of Facts defined by the referenced taxonomy organized by Context and Units.
- **Contexts**—Context contains information about the entity being described, the reporting period and the reporting scenario. Collectively, these give the appropriate context for understanding the reported facts.
- **Fact Values**—A fact value is an individual value for a concept defined in a taxonomy. For example if a taxonomy defines a concept called "Name," its fact value may be "Steve."

- **Business Rules**—Business rules are Excel-like formulas which can be used to build complex validation rules or calculate ratios for business analysis.
- **Tuples**—A tuple is a structure of items in an instance document, defined in XBRL 2.1 as complex data types (definition links in XBRL 2.0). For XBRL 2.1 one XML Schema element is saved in the taxonomy .xsd file specifying the tuple structure. Tuples can contain items or other tuples, and can be nested to any depth required.

  V2 tuples (they do not exist in V1) are usually defined in the definition linkbase. However, many V2 taxonomies did not provide such definition links, and instead used presentation links to identify tuple item collections. The taxonomy loader recognizes definition or presentation links from tuple-to-tuple items as composing a tuple structure.

  The conversion process maintains the tuple composition as loaded by the V2 taxonomy loader, only changing the schema compositional element to all or sequence as indicated.

Instance Editor Components

These topics discuss the components of the Instance Document Edit:

- "Instance Editor Menu Options" on page 108
- "Instance Editor Toolbar Buttons" on page 110
- "Understanding Instance Documents" on page 107
- "Instance Show View Menu" on page 113

Instance Editor Menu Options

The menu options and toolbars change between the Instance Document editor and the Taxonomy Editor, depending on which screen control is activated.

Following usual windows behavior, activating (by clicking to select) an instance document's controls for taxonomy (Taxonomy Relationship View, Taxonomy Elements List, and so on), switch the toolbar and menus to that of taxonomy editing. Clicking to select any instance control (Instance Relationship View, Instance Fact Values Grid, Instance Dockable Controls and Property Grids), switches the menu and toolbar to that of the editing the instance.

Table 49 Instance Document Editor Options

Option	Description
New	Creates a taxonomy document or instance document.
Open	Opens another taxonomy document or instance document. **Note:** When an instance document is opened, a taxonomy must be discoverable from the instance document, or else specified in the Open Instance Document window.
Close	Closes an open taxonomy document or instance document. If the instance document has not been saved and changes are made, you are prompted to save the instance document.

Option	Description
Save	Saves the selected instance document. If the instance document has not been saved previously, you are prompted to indicate where to save the instance document.
Save as File	Saves directly to a location you select.
Save to Server	Saves directly to a Web location you select.
Load taxonomy	Loads a taxonomy into the namespaces collection of this instance document. Allows instance document to contain fact values described by elements of the specified taxonomy.
Load mapping	Loads a mapping specification replacing a mapping specification that was previously loaded.
Load instance data	Appends instance data to the current instance document. XBRL 2.1 and XBRL 2.0 instance documents can be loaded. Contexts, Units, and Fact Values may be selected for loading from the Open Instance Document window.
Import	Import instance document fact values, contexts, and units, from Microsoft Excel, Access, or Word, or OLE DB Provider databases, CSV, or XML files. If a mapping specification is loaded, mapped source data may be imported, and an audit trail of mappings captured for reporting.
Page Setup	Displays the page setup window where print orientation and other print options can be set.
Print Preview	Preview a report of instance document data, contexts, and source data mapping audit trail.
Print	Print a report of instance document data, contexts, and source data mapping audit trail.
Exit	Exits XBRL Taxonomy Designer.

Table 50 Instance View Menu Options

Option	Description
Tree Order	Select how you want to view the instance document. See "Instance View Pane" on page 111.
Show	Use a non-tree view for the Instance View pane. There are several options. See "Instance Show View Menu" on page 113.

Table 51 Instance Tool Menu Options

Instance Tools Menu Options	Description
Analyze Business Rules	Validate the instance document validation with a set of business rules, usually provided as a set of formulas in an extension taxonomy to the instance data's taxonomy. Defaults to saving fact values produced by business rules formulas in an instance document.
Validate Instance Document	Validate the instance document for selected XML and XBRL options. Performs calculation linkbase validation. Formulas for elements related in calculation linkbase attempt to "fire" during calculation linkbase checking, those of elements unrelated by calculation linkbase attempt to "fire" at end. Options to compute missing fact values, to save them, and various reporting features.
Options	Opens the application Options form to set XBRL Taxonomy Designer options.
Plug-ins	Plug-in tools dynamically loaded from plug-in directory are provided here. Includes XBRL Taxonomy Designer provided and end user provided plug-ins.

Instance Tools Menu Options	Description
XBRL GL Importer	Performs a conversion to a financial reporting taxonomy instance document for a loaded and selected XBRL GL instance document,
XBRL Stylist	Allows interactive editing and preview of formatted instance document display using XSLT style sheets.

Table 52 Instance Windows Menu Options

Option	Description
Show All Dockable Controls	Show all dockable controls which are hidden for some reason.
Hide All Dockable Controls	Hide all dockable controls.
Taxonomy Properties	
Instance Document Properties	

Table 53 Instance Help Menu Options

Option	Description
Disclosure Management Online Help	Opens the online help that is the most up to date help for XBRL Taxonomy Designer .
License agreement	Displays the XBRL Taxonomy Designer license agreement.
Application registration	Enter application registration codes into the application.
About	Displays the XBRL Taxonomy Designer About box.

Instance Editor Toolbar Buttons

This section explains the Instance Editor toolbar.

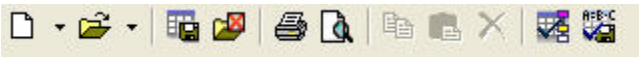


Table 54 Instance Editor Toolbar Buttons

Option	Description
	Same functionality as File, then New.
	Same functionality as File, then Open.
	Same functionality as File, then Save.

Option	Description
	Same functionality as File, then Close.
	Same functionality as File, then Print, then Print.
	Same functionality as File, then Print Preview, and then Print Preview.
	Same functionality as Tools, then Validate Instance Document, and then Validate Instance Document.
	Same functionality as Tools, then Validate Business Rules, and then Validate Business Rules.

Instance View Pane

The Instance View pane provides a relationship tree view of the instance. You can select the hierarchical order in which the instance is displayed by clicking View, then Tree Order.

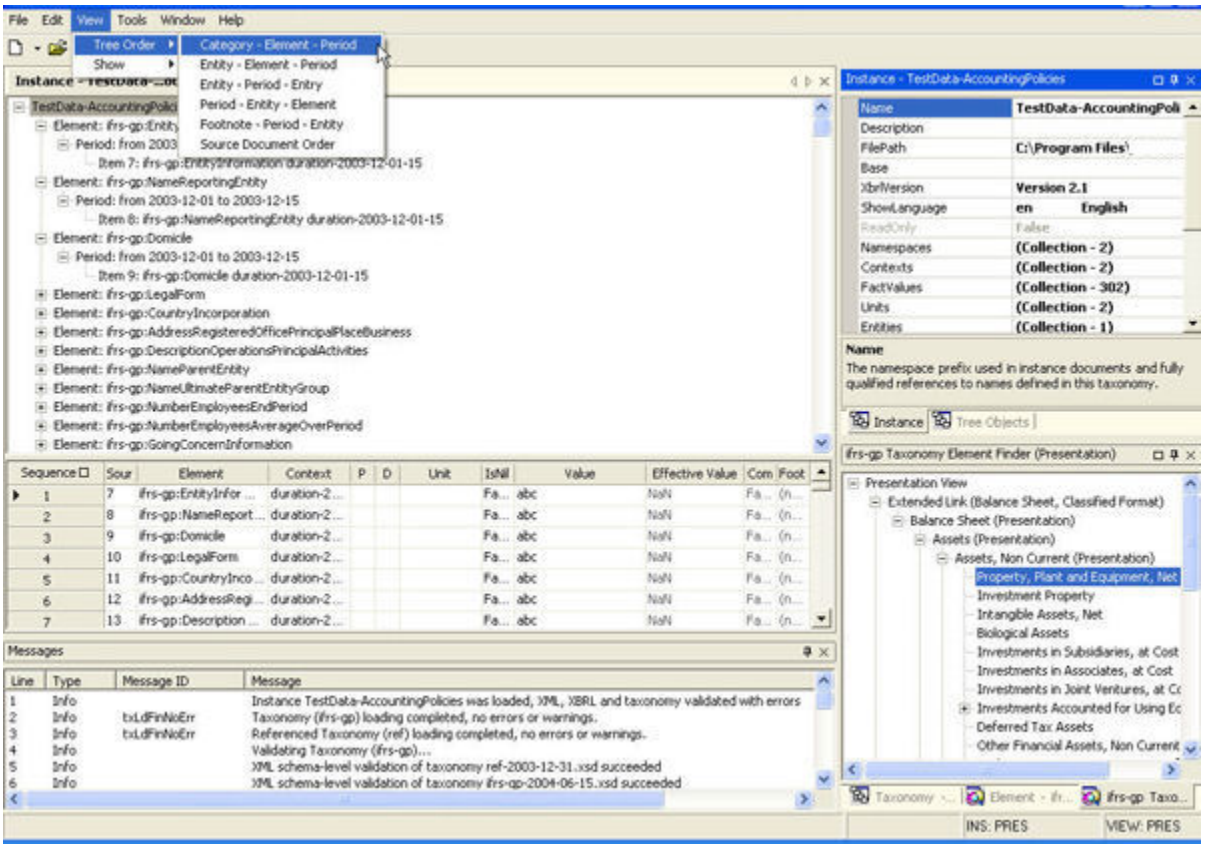


Table 55 Instance Document Tree Order Views

Tree Order View	Description
Category – Entry – Period	This view represents a presentation of linkbase-oriented grouping of fact values organizing by: 1. The presentation linkbase position of the fact's element. 2. The context period. 3. The fact entry order. This usually organizes most along the presentation intentions of the taxonomy authors and is the default tree order.
Entity – Element – Period	This view sorts by: 1. The business entity of context, where fact values aggregate data for multiple reporting entities. 2. Presentation linkbase position of fact's element. 3. Context period. 4. Fact entry order.
Entity – Period – Entry	This view sorts by: 1. Business entity of context. 2. Context period 3. Presentation linkbase position of fact's element 4. Fact entry order.
Period – Entity – Element	This view sorts by: 1. Context period 2. Business entity of context 3. Presentation linkbase position of fact's element 4. Entry order.
Footnote – Period – Entity	This view is used as a special case where the footnotes are used as the major ordering of the tree. Footnotes form a means of organizing related fact values by instance linkbase. Within each footnote, the view is sorted by: 1. Context period 2. Business entity of context 3. Presentation linkbase position of fact's element 4. Fact entry order.
Source Document Order	This view shows the order of facts in the source instance document file.

Instance Show View Menu

➤ To use a view other than Tree Order, select **View**, then **Show**, and then select options.

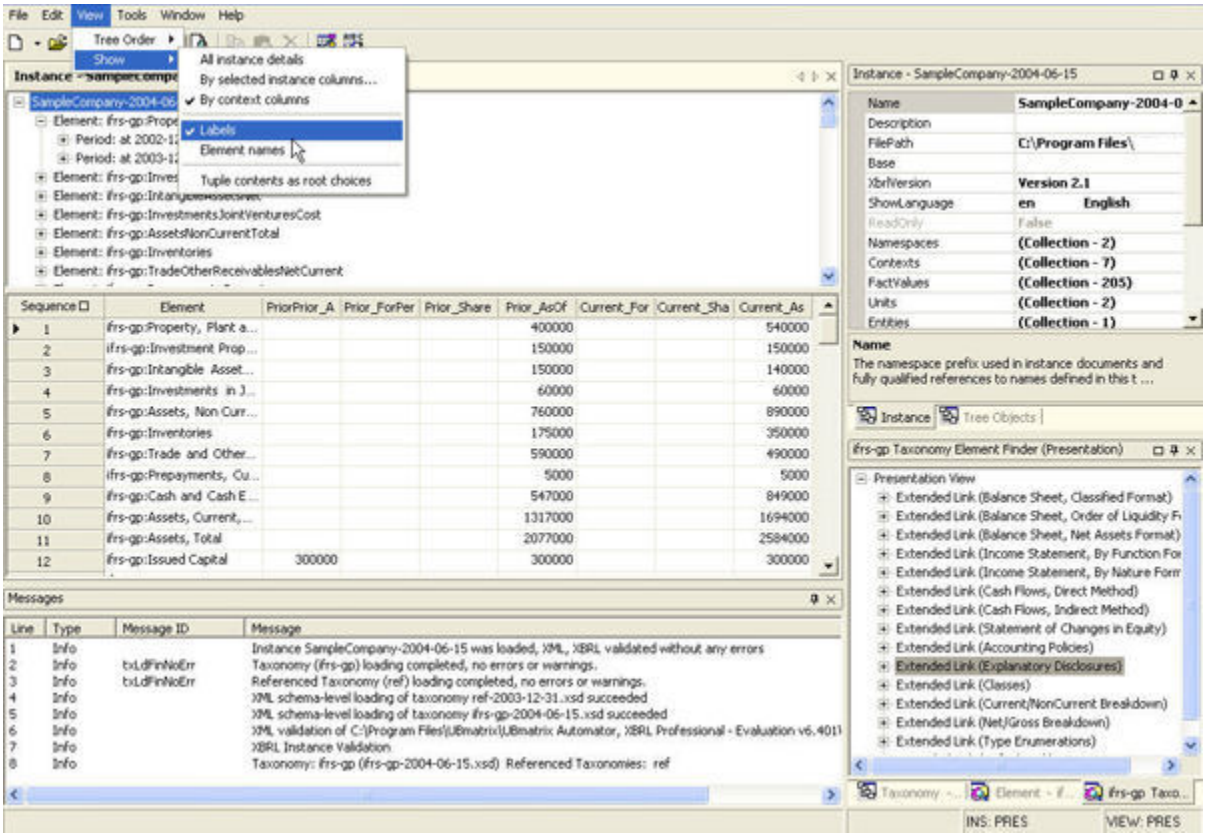


Table 56 Instance Show View Menu Options

Instance Show View Menu Option	Description
All Instance Details	Shows all of the Instance Details.
By selected Instance Columns	Show all of the details of the select columns.
By context labels	Categorizes the view by context labels.
Labels	Categorizes the view by the labels.
Element Names	Categorizes the view by the element names.
Tuple Contents as root choices	Ccategorizes the view by the tuple contents.

Cut, Copy and Paste (Fill Down, Up, and Across)

When editing (filling down/up/across) multiple cells, you may wish to multi-select and fill multiple cells. Select these cells, and then selecting Ctrl+X / Ctrl+C deactivates the controls when selecting individual cells, and enable you to cut/copy and paste into multiple cells.

Note: The use of these functions bypass the XBRL Taxonomy Designer normal functions and work directly on the data grid. This is not recommend for normal use and may lead to errors.

Selecting the Ctrl key when selecting a cell with the mouse left button down and dragging down cell copies the value in the first cell to the other cells when the mouse button is released.

Example:

2	Sequence	Element	Context	Unit	Value
3	1	ci:land	Context-2003	Euros	
4	2	ci:building			
5	3	ci:furniturefixtures			
6	4	ci:ComputerEquipment			
7	5	ci:Other			
8	6	ci:TotalPropertyPlantEquipment			

1. Select a range of cells and enter Ctrl + C to copy the cells.

Sequence	Element	Context	Unit	Value
1	ci:land	Context-2003	Euros	
2	ci:building			
3	ci:furniturefixtures			
4	ci:ComputerEquipment			
5	ci:Other			
6	ci:TotalPropertyPlantEquipment			

2. Select a range of cells to paste (fill with a value) and then enter Ctrl + P to paste the cells.

Sequence	Element	Context	Unit	Value
1	ci:land	Context-2003	Euros	
2	ci:building	Context-2003	Euros	
3	ci:furniturefixtures	Context-2003	Euros	
4	ci:ComputerEquipment	Context-2003	Euros	
5	ci:Other	Context-2003	Euros	
6	ci:TotalPropertyPlantEquipment	Context-2003	Euros	

Note: Pasting invalid values into cells causes the error shown below.

Sequence	Element	Context	Unit	Value
1	ci:land		Context-2003	Euros
2	ci:building		Context-2003	Euros
3	ci:furniturefixtures		Context-2003	Euros
4	ci:ComputerEquipment		Context-2003	Euros
5	ci:Other		Context-2003	Euros
6	ci:TotalPropertyPlantEquipment		Context-2003	Euros

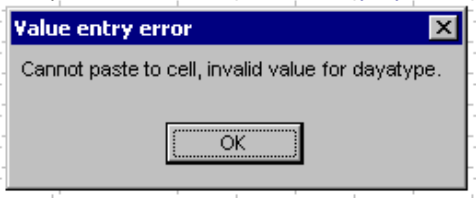


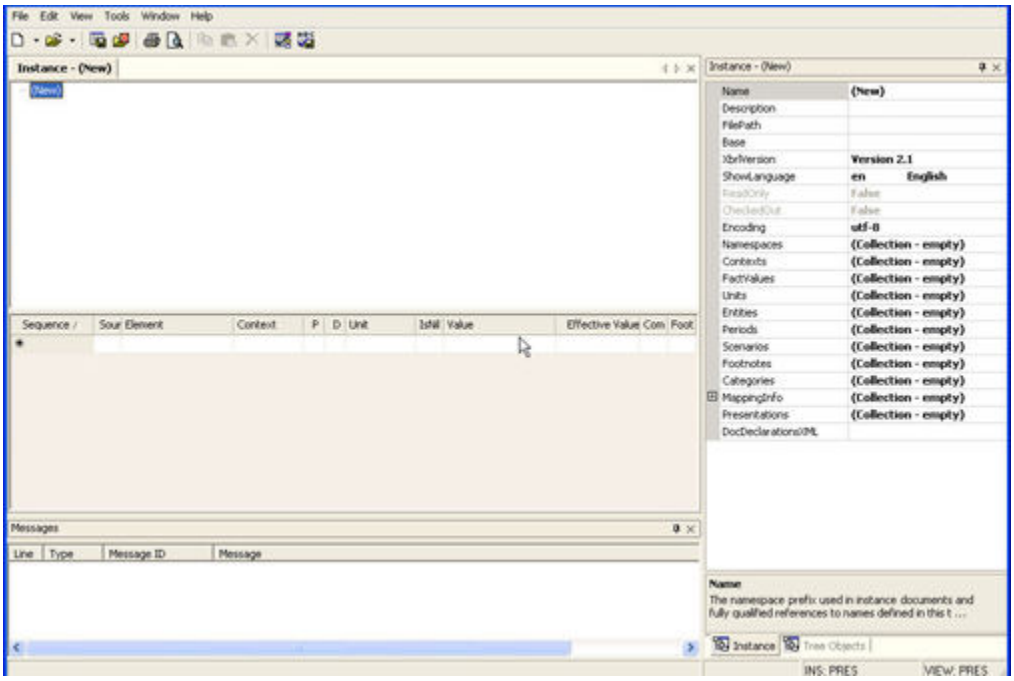
Table 57 Data Grid Shortcut Keys

Shortcut	Description
Ctrl+C	De-activates the drop-down and sets focus on cells for a copy operation.
Ctrl+X	De-activates a cell dropdown, and sets focus on cells for a cut operation.
Ctrl+D	Deletes the row (vertical row-by-row view).
Delete	Deletes what is in the selected field.

Creating an Instance Document

➤ To use XBRL Taxonomy Designer to create an instance, from within the Instance Editor, select **File** then **New**, and then **Instance**.

From the instance editor, you can add to this instance document by creating contexts, entities, periods, and fact values.



Related Topics

"Using the Context Collection Editor" on page 127

"Using Entities" on page 131

"Using the Period Collection Editor" on page 134

"Fact Values Pane" on page 128

Opening an Instance Document

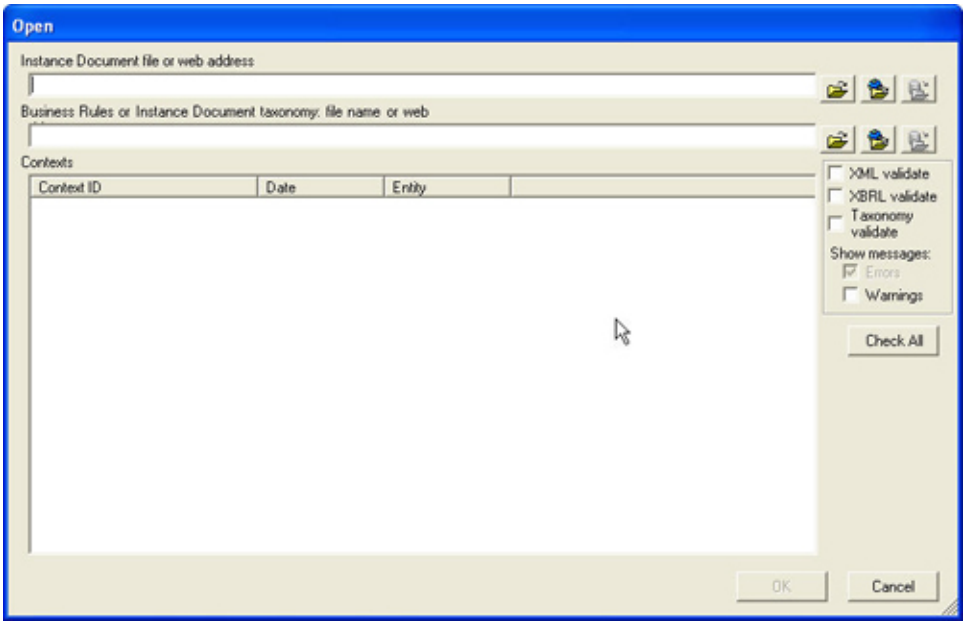
This section explains how to open an instance document using a:

- local machine
- the web

Opening an Instance Document Using a Local Machine

➤ To open an instance document from your local system from the XBRL Taxonomy Designer main screen:

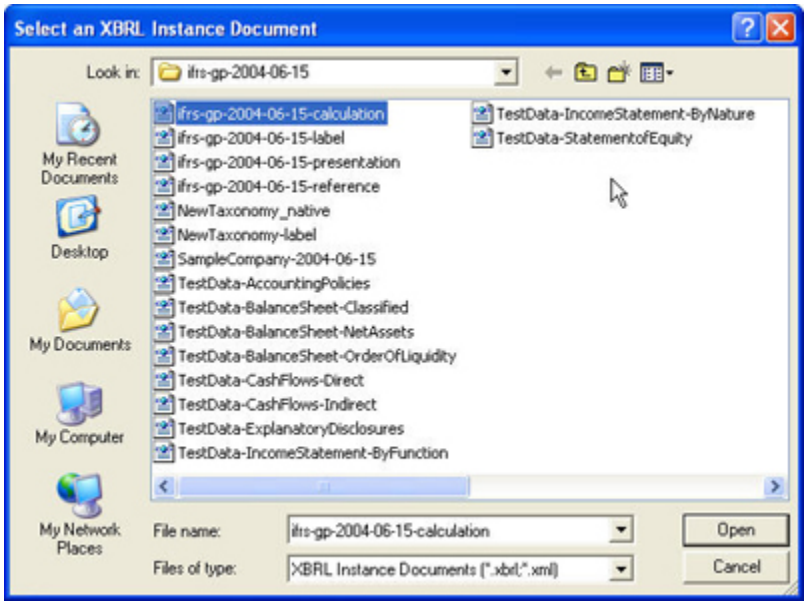
1 **Select File, then Open, and then Instance.**



Three icons in the top right of the window allow you to open instance documents.

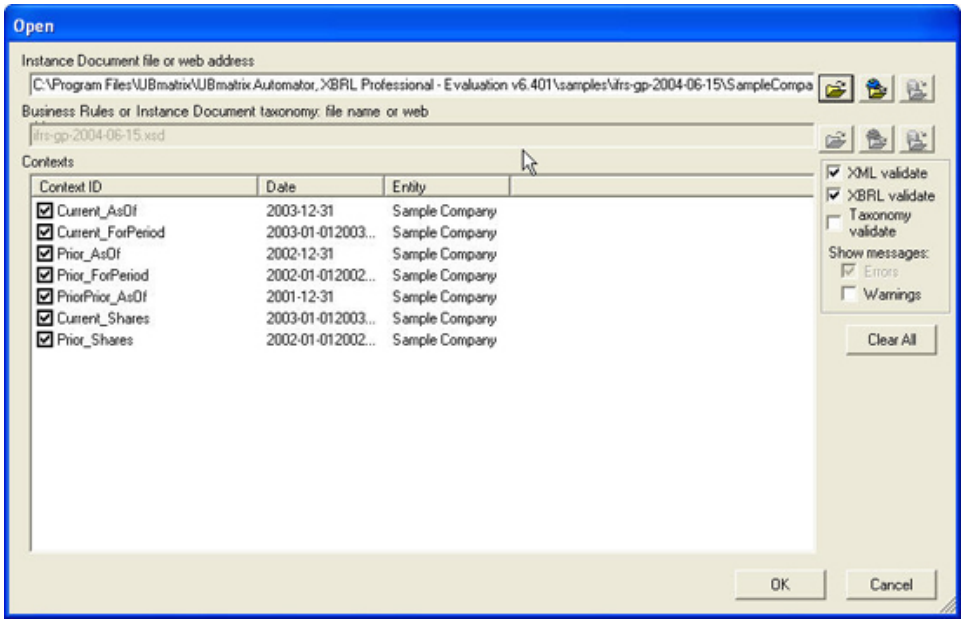
2 **Click  to open the Select an Instance Document browse window and navigate to the location of an Instance document file.**

3 **Browse to the instance document.**



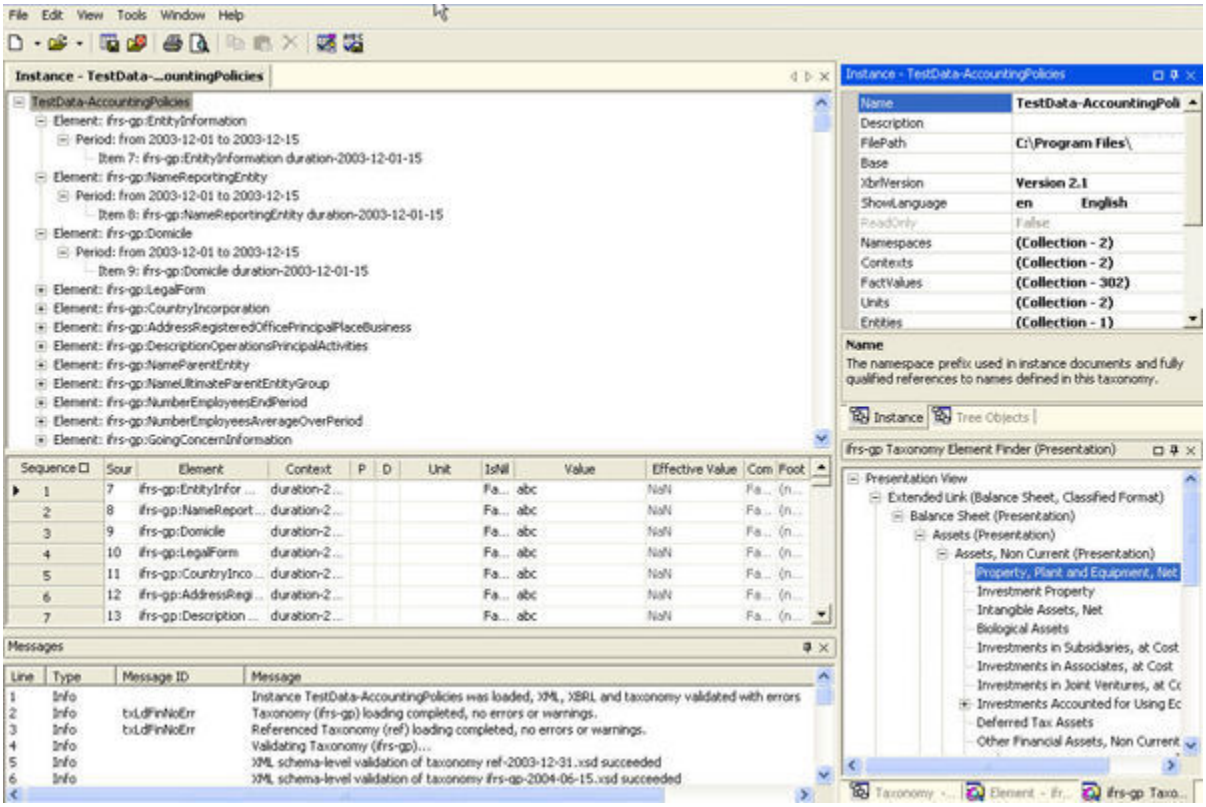
4 **Double-click the appropriate instance document file or select it and click Open.**

Labels links, Presentation links, and Calculation links information for the instance document displays in the Open Instance Document window as shown below.



5 **Select the appropriate load options and click OK at the bottom of the window to load the selected instance document.**

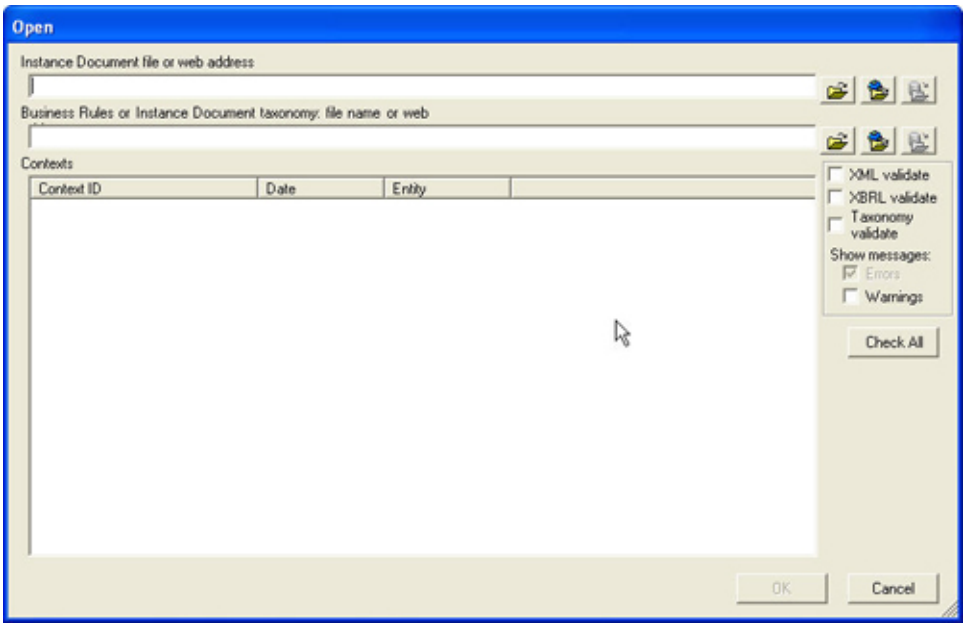
The Instance Document Editor screen is displayed with the selected instance document loaded and ready for edit as seen below. If you selected any of the validation options from the Instance Document Load section, XBRL Taxonomy Designer automatically validates the instance document and reports all errors and warnings in the message section at the bottom of the screen.



Opening an Instance Document Using the Web

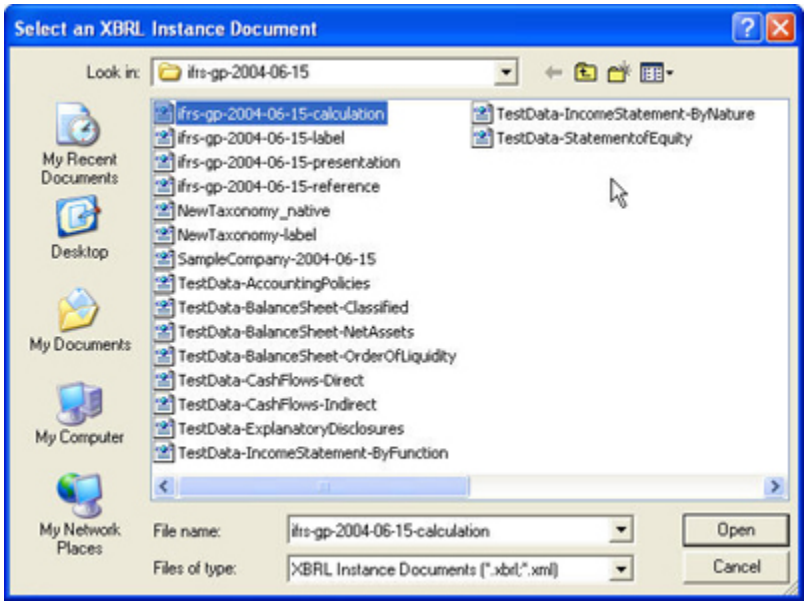
➤ To open an instance document from your local system from the XBRL Taxonomy Designer main screen:

1 Select **File**, then **Open**, and then **Instance**.



2 Click  to open the Select an Instance Document using the Web browse window.

3 Select the appropriate file and click **Open**.

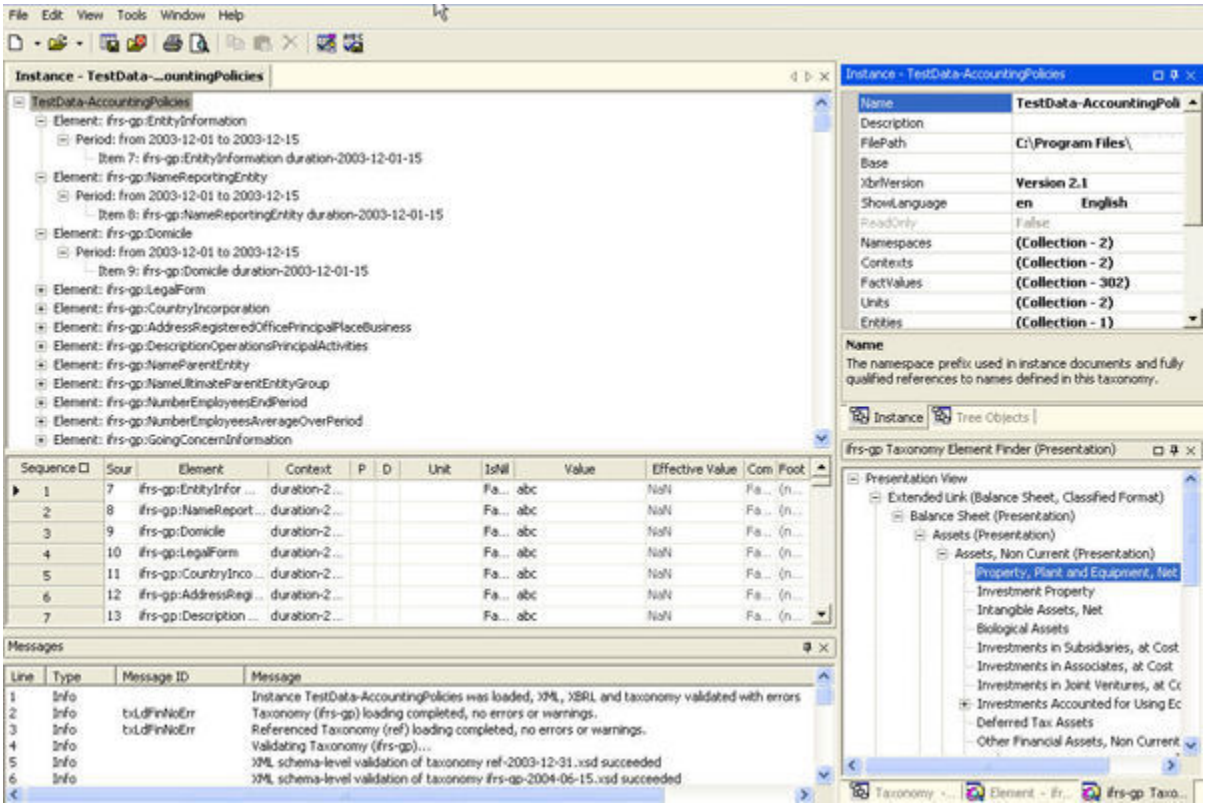


4 Double-click the appropriate instance document file or select it and click **Open**.

Labels links, Presentation links, and Calculation links information for the instance document displays in the Open Instance Document.

5 Select the appropriate load options and click **OK** at the bottom of the window to load the selected instance document.

The Instance Document Editor screen is displayed with the selected instance document loaded and ready for edit. If you select any of the validation options from the Instance Document Load section, XBRL Taxonomy Designer automatically validates the instance document and reports all errors and warnings in the message section at the bottom of the screen.



Instance Document Load Options

There are several Instance Document load options in the Open Instance Document window. Each of the taxonomy load options are detailed in the table below.

Table 58 Instance Document Load Options

Instance Document Load Options	Description
XML validate	Activates the XBRL Taxonomy Designer XML Validation of the Instance Document during load.
XBRL validate	Validates the instance according to standard XBRL practices.
Taxonomy validate	Validates the taxonomy associated with this instance document.
Show Messages: Errors	Sets the level of errors which appear in the messages dialog. Check this button to see non-fatal errors in Instance or Instance Document validation.
Show Messages: Warnings	Sets the level of warnings which appear in the messages dialog. Check this button to see non-fatal errors in Instance or Taxonomy validation.
Enable Editing	Unlocks read only instance documents. Use this function to make an extension of a taxonomy or instance, such as the US-GAAP.

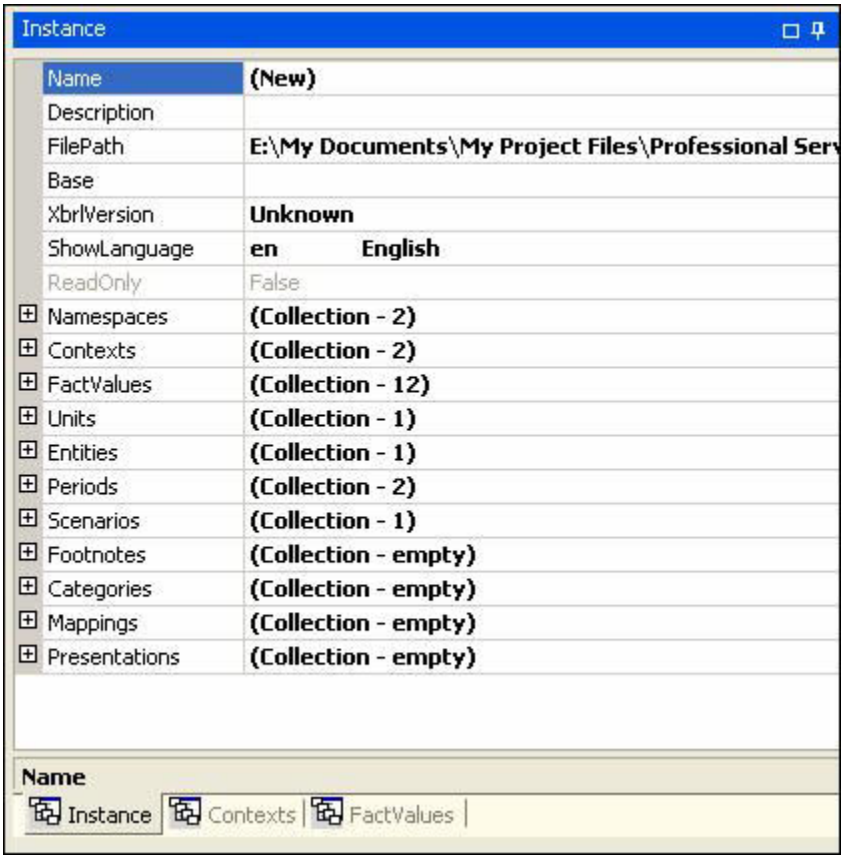
Properties Grid

This section explains how to use the instance document properties grid.

Instance Document Properties Grid

You can edit the properties of the instance document or of individual contexts using the Instance Document Properties Grid pane. The tabs at the bottom of the grid allow you to select which set of properties you are viewing. The options include Instance, Context, Period, or Fact Value.

The figure below shows the properties that display when the Instance tab is selected.



The table below gives descriptions of the properties listed in the Instance Properties Grid.

Table 59 Descriptions of Instance Properties

Property	Description
Name	Instance document name (for example, “Balance Sheet – 2003-12-31”)
Description	Instance document description (for example, “Balance sheet for the year ended December 31, 2003, released version, approved by the CFO”
File Path	Physical location of the instance document, usually a local drive path or URL
Base	Supports XML:Base attributes

Property	Description
XBRL Version	Instance document XBRL version, for example, “XBRL Version 2.1”
Show Language	Indicates the user preference of thelanguage to use within this instance document. Useful when multiple label linkbases exist for the instance document. **Note:** The instance document language is distinct from the taxonomy language.
Read Only	Indicates that the instance document is read only. You may change this option to make the instance document read only. For example, False which indicates the instance document is read/write.
Namespaces	Collection of all namespaces included in this instance document. Ellipsis button activates Namespaces Collection Editor. Expansion button allows in-place editing. Taxonomies are loaded in course of loading instance documents and by File, then Load, then Taxonomy. Namespaces that refer to `.xsd` files which are not taxonomies can be explicitly specified here.
Contexts	Collection of contexts for this instance document. Ellipsis button activates the Contexts Collection Editor. Expansion button allows in-place editing. Contexts are automatically loaded with instance documents; they can be imported by the import facility; they can be manually entered by the Contexts Collection Editor, by in-place editing, and from the Fact Values Data List Contexts column Add/Choose option.
Fact Values	Collection of fact values for this instance document. Fact values are shown in the Fact Values Data List of the Instance Document Editor. The ellipsis button activates an additional pop up Fact Values Data List control. Expansion button allows in-place editing. The Fact Values dockable control allows expansion of individual fact values. Fact Values are automatically loaded with instance documents; they can be imported by the import facility; they can be manually entered by the Fact Values Collection Editor, and by in-place editing.
Units	Collection of units for this instance document. Ellipsis button activates the Units Collection Editor. Units are automatically loaded with instance documents; they can be imported by the import facility; they can be manually entered by the Units Collection Editor, and from the Fact Values Data List Units column Add/Choose option.
Entities	Collection of entities for this instance document. An XBRL Entity is a reporting organization such as Universal Business Matrix, LLC. Ellipsis button activates the Entities Collection Editor. Entities are automatically loaded with instance documents; they can be imported by the import facility; they can be manually entered by the Context Add/Choose window or Contexts Collection Editor, Entity property, and from the Fact Values Data List using the contexts column, Add/Choose, context’s entity property.
Substitution group	The substitution group attribute is defined on every element. By default, it must be an item or a tuple. It can also hold the value of other elements, which are derived from items or tuples. An item is the default substitution group for a concept that is reported in an instance document with a value. A tuple is used to group of items in an instance document.
Periods	Collection of periods for this instance document. Ellipsis button activates the Periods Collection Editor.
Scenarios	Collection of scenarios for this instance document. Ellipsis button activates the Scenarios Collection Editor.
Categories	Collection of categories for this instance document Ellipsis button activates the Category Collection Editor. **Note:** A category is an arbitrary piece of an instance document. Categories are used to help organize the instance document and to partition the instance document into physical files, if appropriate.

Property	Description
MappingInfo	Collection of mappings for this instance document including. Property Description FilePath The file path of the mappings specification Type Type of mapping (see Taxonomy Mapping) Key Column Columns mapped in source data Group Key Start Pattern For repeating groups (as in cash flow statements) Group Repeat Period The time period for the import group. For example, 1 Year Mappings Collection of mapping specifications (from Taxonomy Mapping design)

Instance Grid Views

This section explains how to work with instance grid views.

Element, Property, or Value

Depending on which cell you select, a list of concepts, properties, or a value control is displayed to enable you to build an instance. Examples are shown below.

Figure 12 Elements

Figure 13 Properties

Figure 14 Values

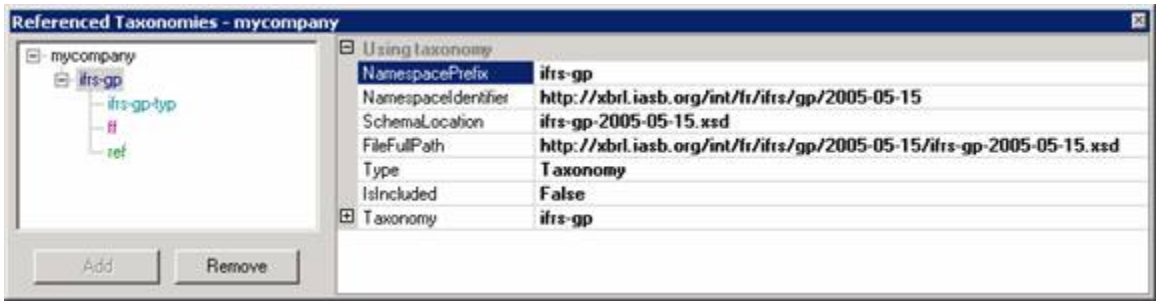


Figure 15 Tuple Name

The string which is displayed in the value field for a tuple header row "(tuple)" is the first concept within the tuple definition which happens to be a string. If the tuple has an id (IDREF), then you may display the (tuple) annotation. (This feature has not been implemented.) You can also edit/change this string.

Figure 16 Excel Grouped Concepts

	A	B	C	D	E
1	Element	Context	Units	P	D
2	Director				
3	Name	Context-String			
4	Salary	Context 2003	Euros		INF
5	Bonus	Context 2003	Euros		INF
6	Director Fees	Context 2003	Euros		INF
7	Fair Value of Options Granted	Context 2003	Euros		INF
8					
9					

Tuple contained concepts are grouped using a colored line as listed per the tuple definition.

Sequence	Element
1	ci:Director
1.1	ci:Name
1.2	ci:Salary
1.3	ci:Bonus
1.4	ci:DirectorFees
1.5	ci:FairValueOptionsGranted

Figure 17 Nested tuple (using indent)

Sequence	Element	Context	Unit	Value
1	ci:RelatedParty			(tuple) John Doe
1.1	ci:NameRelatedParty	Context-2003		John Doe
1.2	ci:NatureRelationship	Context-2003		Director
1.3	ci:TypeRelationship	Context-2003		iascf:SignificantStockhold...
1.4	ci:TransactionWithRelatedParty			(tuple) Sales
1.4.	ci:DescriptionTransaction	Context-2003		Sales
1.4.	ci:TypeTransaction	Context-2003		iascf:Sales
1.4.	ci:TransactionPricingPolicy	Context-2003		Cost
1.4.	ci:AmountTransaction	Context-2003	Units-Monetary	100000
1.5	ci:TransactionWithRelatedParty			(tuple) Loans
1.5.	ci:DescriptionTransaction	Context-2003		Loans
1.5.	ci:TypeTransaction	Context-2003		iascf:Loans
1.5.	ci:TransactionPricingPolicy	Context-2003		Cost
1.5.	ci:AmountTransaction	Context-2003	Units-Monetary	500000
1.6	ci:TransactionWithRelatedParty			(tuple) Lease for office bu...
1.6.	ci:DescriptionTransaction	Context-2003		Lease for office building
1.6.	ci:TypeTransaction	Context-2003		iascf:Lease
1.6.	ci:TransactionPricingPolicy	Context-2003		Market
1.6.	ci:AmountTransaction	Context-2003	Units-Monetary	1000000

Figure 18 Nested tuple (using tuple contents indent and tuple concept coloring)

Sequence	Element	Context	Unit	Value
1	ci:RelatedParty			(tuple) John Doe
1.1	ci:NameRelatedParty	Context-2003		John Doe
1.2	ci:NatureRelationship	Context-2003		Director
1.3	ci:TypeRelationship	Context-2003		iascf:SignificantStockhold...
1.4	ci:TransactionWithRelatedParty			(tuple) Sales
1.4.	ci:DescriptionTransaction	Context-2003		Sales
1.4.	ci:TypeTransaction	Context-2003		iascf:Sales
1.4.	ci:TransactionPricingPolicy	Context-2003		Cost
1.4.	ci:AmountTransaction	Context-2003	Units-Monetary	100000
1.5	ci:TransactionWithRelatedParty			(tuple) Loans
1.5.	ci:DescriptionTransaction	Context-2003		Loans
1.5.	ci:TypeTransaction	Context-2003		iascf:Loans
1.5.	ci:TransactionPricingPolicy	Context-2003		Cost
1.5.	ci:AmountTransaction	Context-2003	Units-Monetary	500000
1.6	ci:TransactionWithRelatedParty			(tuple) Lease for office bu...
1.6.	ci:DescriptionTransaction	Context-2003		Lease for office building
1.6.	ci:TypeTransaction	Context-2003		iascf:Lease
1.6.	ci:TransactionPricingPolicy	Context-2003		Market
1.6.	ci:AmountTransaction	Context-2003	Units-Monetary	1000000

Item concepts are listed individually.

Sequence	Element
1	ci:Land
2	ci:Land

Instance Grid Field Display

When you open an instance document, by default all instance details are shown. These titles and values are justified within the fact value grid below. The grid control acts are similar to Excel in that you can edit cells values directly, or enter values from available value lists. You can also use the fill, cut, copy , or paste features between cells. Where these operations should be prohibited, then error handling is defined. When grid cells are displaying non-editable data this is reflected by graying out the item (or another similar color.)

Collection Editors

This section explains how to work with the following editors:

- Namespace Collection Editor
- Context Collection Editor
- Fact Value Collection Editor

Namespace Collection Editor

This Editor can be found in the Instance Property Grid.

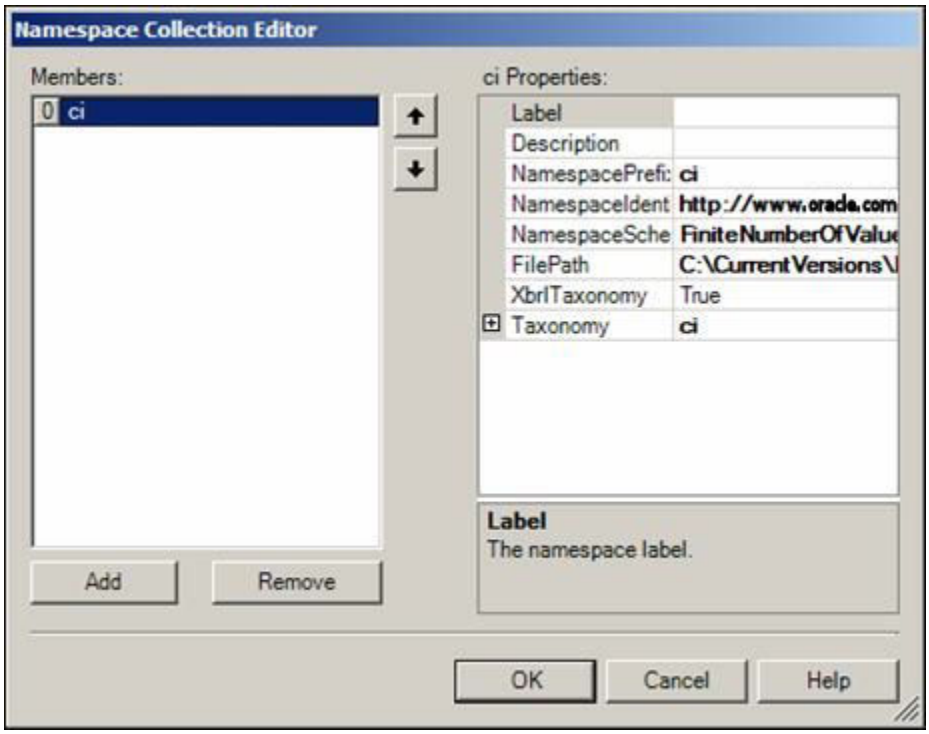


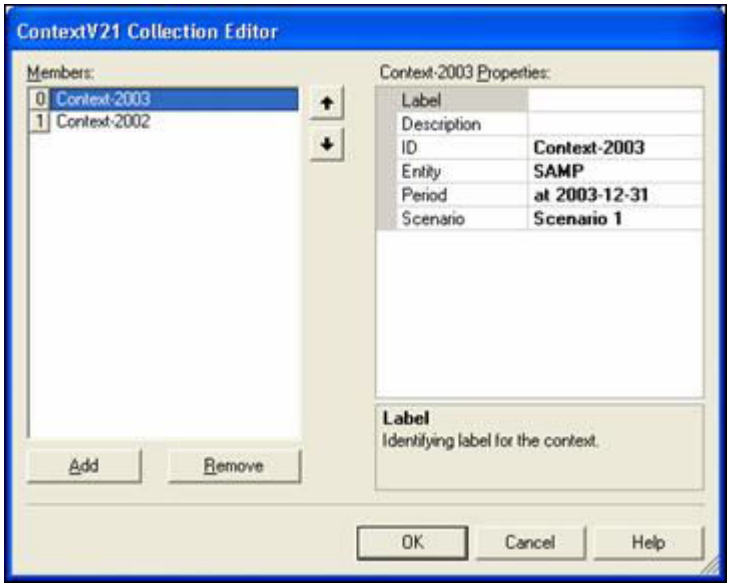
Table 60 Namespace Collection Editor Fields

Property	Description
Label	Namespace label (for example, US GAAAP CI). The namespace label is derived from the taxonomy, if it exists.
Description	Namespace description.
Namespace Prefix	Taxonomy namespace prefix, which is the same as the namespace's xmlns entry in the instance document. It represents the instance document's prefix for the taxonomy (or schema file), which can differ from the one the taxonomy (schema file) uses or the ones other referencing taxonomies (schema files) use.
Namespace Identifier	Namespace identifier for the taxonomy or schema file. A unique identifier must be the same everywhere.
Namespace Schema Location	Schema location the instance document specifies. This is not the same as the file path if a local cache or resolver is involved.

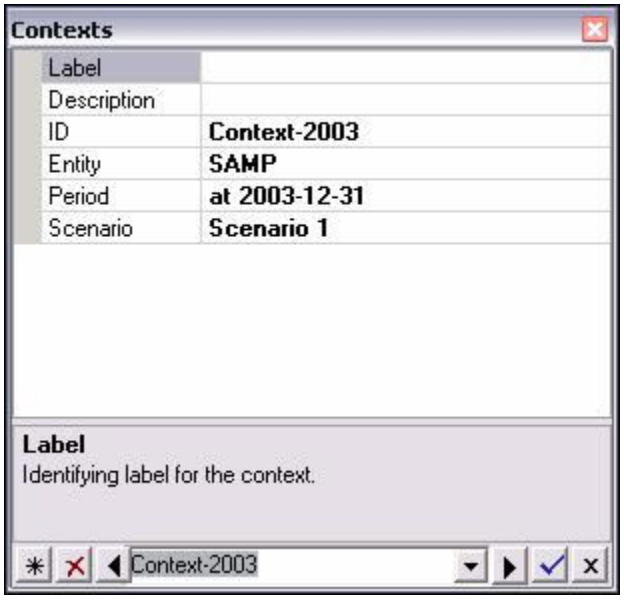
Property	Description
File Path	Physical location of the taxonomy on local disk or URL (for example, `C:\Temp\MyTaxonomy.xsd`).
XBRL Taxonomy	Boolean value that indicates whether the taxonomy is an XBRL taxonomy or other schema (for example, True indicates that the taxonomy is an XBRL taxonomy).
Taxonomy	If the schema file is an XBRL taxonomy, then expand this property to access the taxonomy properties, including referenced taxonomy and linkbase collections. If you have read and write permissions for the taxonomy, you can edit these taxonomy properties.

Using the Context Collection Editor

This Editor is located in the Instance Property Grid.



This is the Contexts Add/Choose window for adding, choosing, and editing contexts from within the Fact Values Data List:



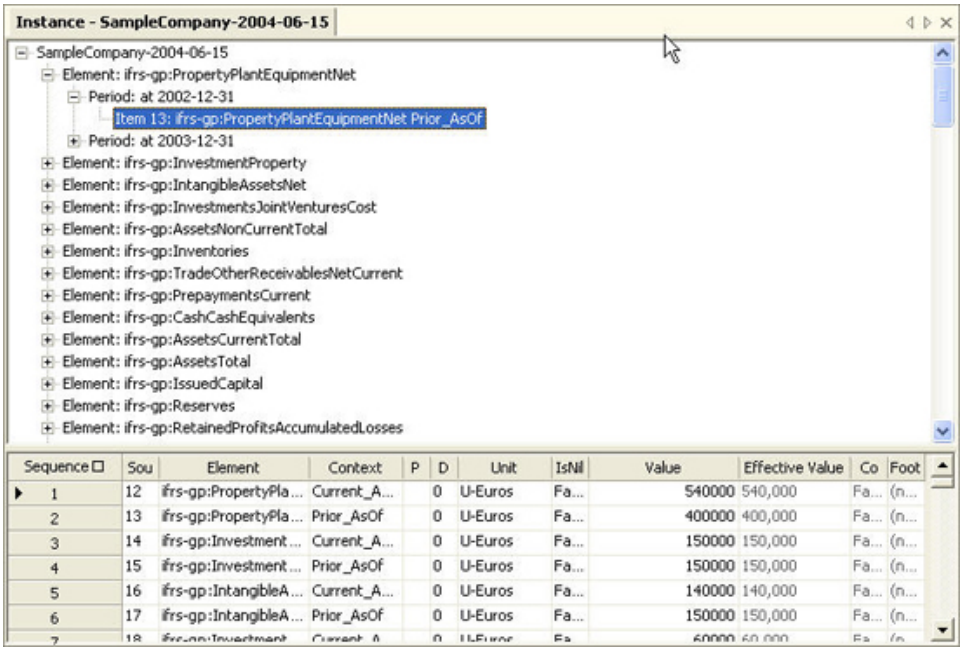
The Contexts Add/Choose window displays a property grid of the current context, and a browser bar at the bottom. The browser bar lets you add (✳) contexts to the collection, delete (☒), scroll, or accept (☑) a context choice, or cancel (☒) context changes for the selected property.

Table 61 Fields in the Context Collection Editor

Property	Description
Label	Associate a label with a context. This label must be unique for each context, and is used in dialog boxes when referring to contexts within the application (for example, “Current Period”).
Description	Context description (for example, “Current reporting period for the company”).
Context ID	XBRL context ID assigned to the context. This must be an XML ID (for example, “CurrentPeriod”).
Entity	The Entity label, a unique entity ID from the entity collection within this instance document (for example, “Sample Company”. The drop-down button activates the Entity Add/Choose window.
Period	The Period label, a unique period ID from the period collection within this instance document (for example, “2003”). The drop-down button activates the Period Add/Choose window.
Scenario	The Scenario label, a unique scenario ID from the scenarios collection within this instance document (for example,“ –Actual–”. The drop-down button activates the Scenario Add/Choose window.

Fact Values Pane

The figure below shows the Fact Values pane window.



Note: Fact values which are tuples have no context, unit, decimal, precision value, or value associated with them.

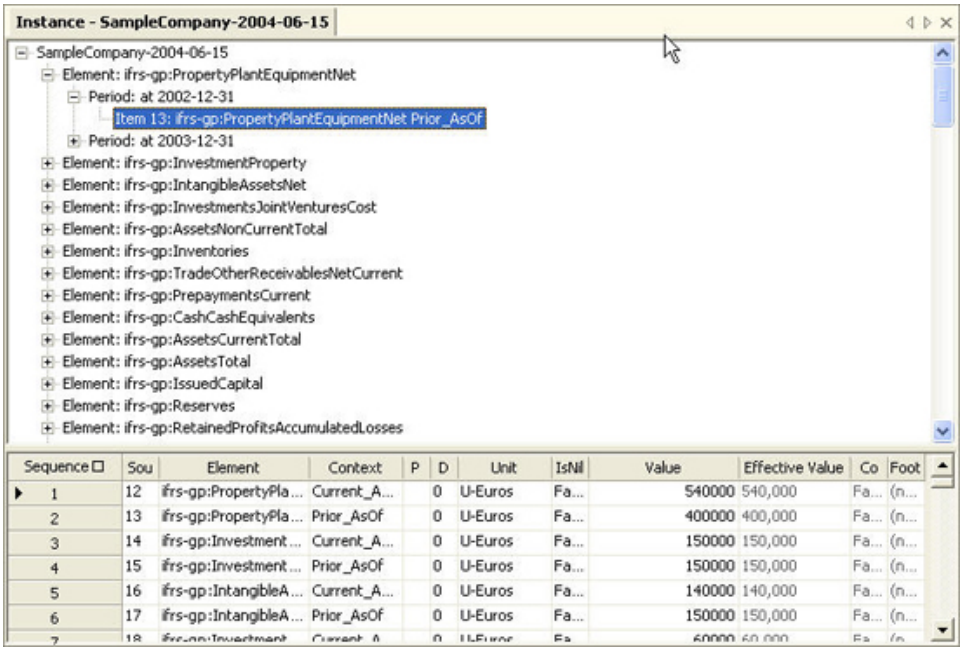
Table 62 Fact Values Collection Editor Fields

Option	Description
Sequence	Shows the physical order of the fact value in the instance document. The sequence provides outlining to expand/hide tuple contents where tuples are available. Tuple elements are distinctly colored, and when the tuple is expanded, the shading of internal tuple items distinguishes tuple contents from others.
Element name	Taxonomy element name for the fact value. When selected, an incremental-searching taxonomy element name finder is displayed as a drop-down combo box. When the element names combo box is dropped down, as you type the letter of a prefix, the closest is highlighted. Typing more letters selects between several which begin with same pattern. If the right prefix is highlighted, the right arrow button moves the cursor past the prefix highlight, and you can type the element name. Names are scrolled into view to correspond to incremental search by the letters typed, to make finding names in long lists very fast.
Context	The context reference from a context within the contexts collection within this instance document. For example, –Context-2003–. The drop-down button activates list of available contexts with bottom entry –...Add/Choose...–, which activates Context Add/Choose window (q.v., above). Must be blank for tuples.
Precision (Detail Column, abbreviated –P–)	A value for precision. For example, –7–. **Note:** A fact value may have a decimal value, or a precision value, but not both. The application default is decimals, but the default can be changed in instance document options. Must be blank for non-numeric and tuple elements, or if decimals are specified.

Option	Description
Decimals (Detail Column, abbreviated –D–)	A value for decimals. For example, –-3– to mean accurate to thousands. **Note:** A fact value may have a decimal value, or a precision value, but not both. The application default is decimals, but the default can be changed in instance document options. Must be blank for non-numeric and tuple elements, or if precision is specified.
Unit	The unit reference from a unit within the units collection within this instance document. For example, –Units-Monetary–. Drop-down button activates list of available units with bottom entry –...Add/Choose...–, which activates the Unit Add/Choose window (q.v., above). Must be blank for non-numeric and tuple elements.
Is Nil	Indicates a nill value for the item fact value. Must be blank for tuple elements
Value	The value of the item. For example, –930000–. (For tuple elements, the first string value of the tuple is shown grayed out on the tuple line, to provide a name or account field hint of what an unexpanded tuple contains in its item fields.)
Effective Value (Detail Column, read only)	The computed effective value, for numeric fields, taking into account decimals and precision. Shows the effective value for calculation validation and formula processing. Blank for non-numeric fields and tuple elements.
Computed Value (Detail column, read only, abbreviated –Comp–)	Set to true if the value was computed (and saved) by XBRL validation from calculation roll-up or formula execution.
Footnotes	Provides access to the footnote collection editor. Footnotes represent a way of providing data-centric associations between fact values. The drop-down button activates a check list of available footnotes with bottom entry –...Add/Choose...–, which if clicked on the name or check box, activates the footnote add/choose.

Fact Values Data List

A collection editor used to create, update, or delete facts in the instance document. In addition, you can set or modify the facts' Context or Units.



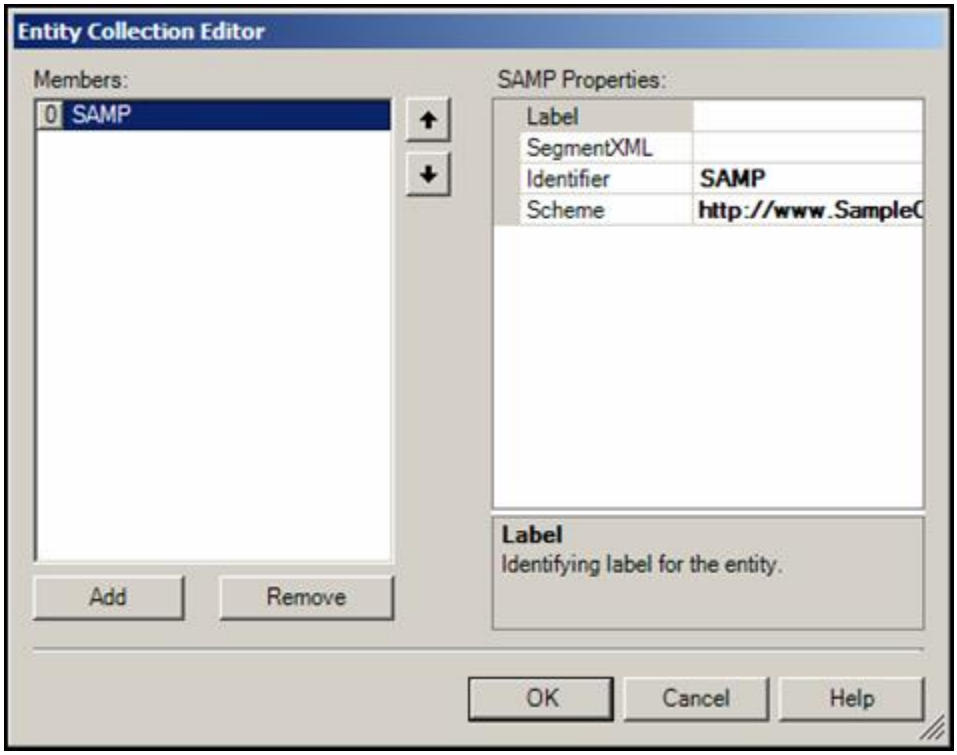
The Fact Values Grid provides a shortcut menu of operations. This menu can be accessed from within the Fact Values Grid or from the factValues window accessed by clicking the ellipsis button in the Instance Properties Grid. These menu options are described in the table below.

Table 63 Fact Value Grid Shortcut Menu Options

Option	Description
Insert	Rows above: Inserts rows above selected row. Rows below: Inserts rows below selected row. (Rows can also be appended to the end of the collection by tabbing or entering –past– the last extant row of the data grid.)
Delete	Deletes the currently selected row.
Copy	Copies the current data grid, as visible, to the clipboard, for pasting into Excel or a text editor.
Paste	Pastes data values copied from Excel into the Fact Values Data List.
Show	Enables you to choose to display fact values using labels or element names. Also allows toggling between displaying or hiding additional table columns..
Refresh	Reloads the fact values collection.
Fill	Center the appropriate result in the first cell. Highlight that cell and click the column of that cell. Right-click and select Fill. This fills the rest of that cell when the same result as the highlighted cell.

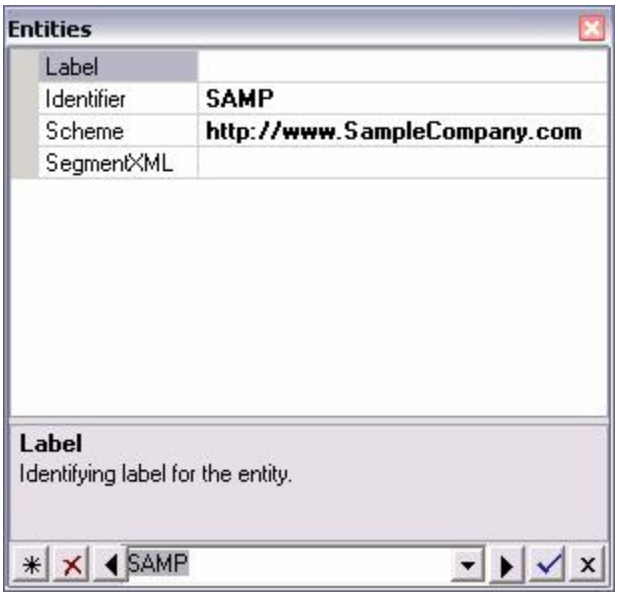
Using Entities

The figure below shows the Entity Collection Editor window. This editor is located in the Instance Property Grid.



➤ To add a member to the Entity Collection:

1 Select **Add (*)**.



2 Use the following table to specify member properties.

Table 64 Entity Collection Editor Fields

Property	Description
Label	Label for entities used within this instance document (for example, "–Sample company–"). The label must be unique within the instance document.

Property	Description
Identifier	Identifier associated with the entity as defined by the XBRL specification (for example, "–SAMP–").
Scheme	Scheme associated with the entity as defined by the XBRL specification (for example, "–http://www.sampleCompany.com–").
Segment XML	XML fragment, as defined by the XBRL specification, that can be used within the segment element within the entity portion of a context (for example, "–<my:Region>Europe<my:Region>–") **Note:** If the segment XML uses entities not defined in the taxonomy (as data types), then you must add an appropriate namespace to the namespace segment entity definitions collection (for example, to generate an "–xmlns:my–" construct).

Using Categories

Categories help organize instance document fact values and partition instance document fact values into documents. Use the Collection Editor to edit categories.

The figure below shows the Category Collection Editor window. This Editor is located in the Instance Property Grid.

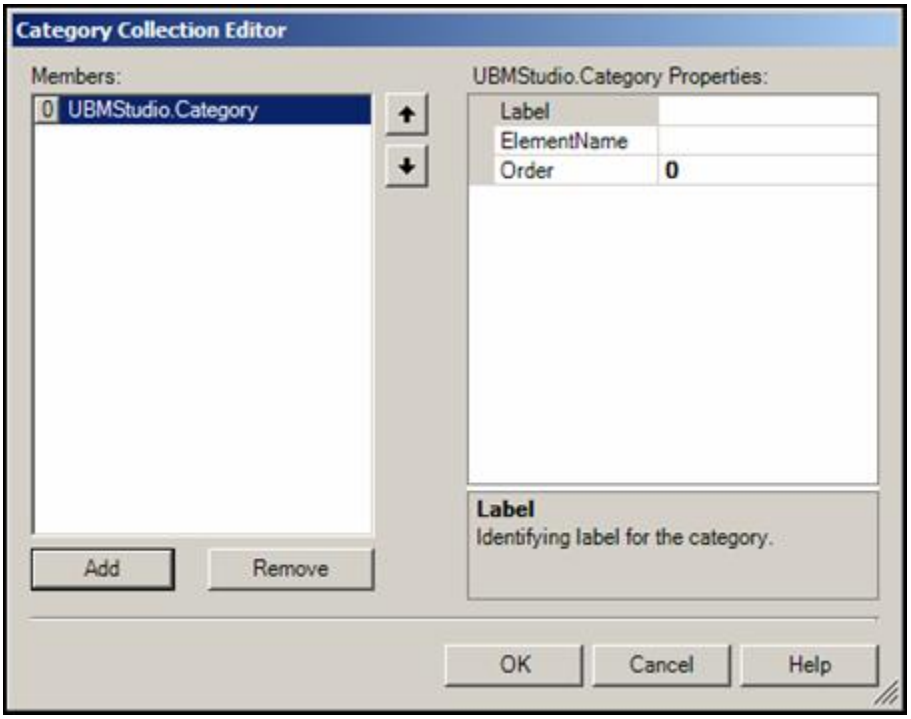
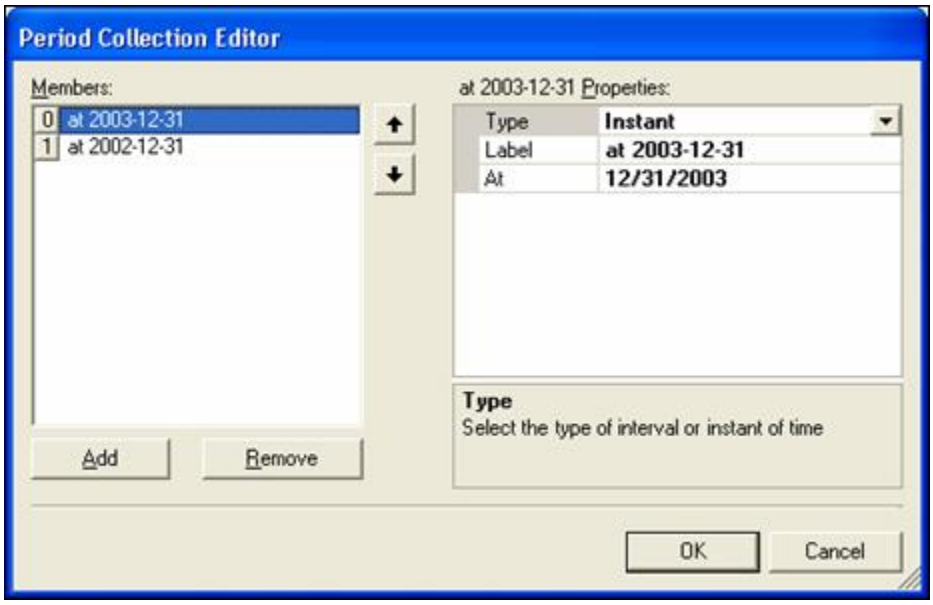


Table 65 Fields in the Category Collection Editor

Option	Description
Label	The category label, which must be unique within the instance document (for example, "–Balance Sheet–" for all fact values in the balance sheet document)
Element name	Category description
Order	Fact value collection for the category

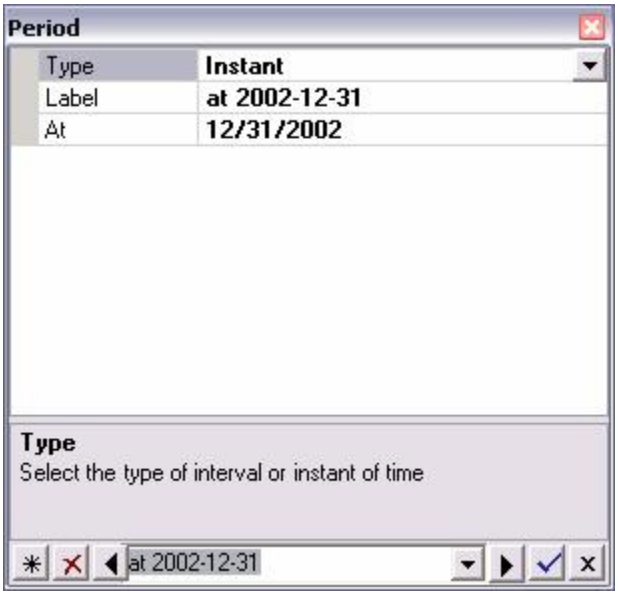
Using the Period Collection Editor

The figure below shows the Period Collection Editor window. This Editor can be found in the Instance Property Grid.



➤ To add a member to the Period Collection:click **Add** (*).

1 Select **Add (*)**.



2 Use the following table to specify the period fields.

Table 66 Fields in the Period Collection Editor

Option	Description
Label	Human readable label for each period which is used in combo boxes. Must be unique within an instance document. For example, –As of January 31, 2004–.

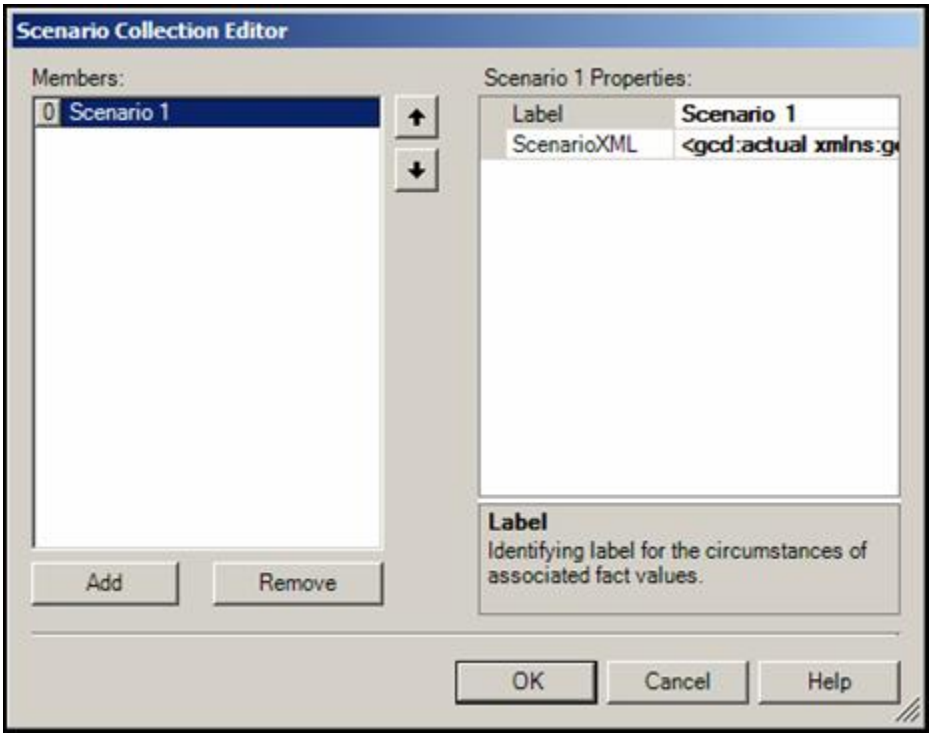
Option	Description
Type	Indicates the type of period, values are –Instant–, –Duration– or –Forever–. **Note:** If –Instant– is selected, then the property grid displays a property –Instant– to enter an instant value. If –Duration– is selected, then –StartDate– and –EndDate– is displayed. If –Forever– is selected, no other properties are displayed.
Instant (Only displays if type of –Instant–.)	Instant per the XBRL specification. For example, –2003-12-31–. The drop-down button activates a date-chooser control.
Start Date (Only displayed if type of –Duration– is selected.)	Date and time per the XBRL specification. For example, –2003-01-01–. The drop-down button activates a date-chooser control.
End Date (Only displayed if type of –Duration– is selected.)	Date and time per the XBRL specification. Must be greater than the value for Start Date. For example, –2003-12-31–. The drop-down button activates a date-chooser control.

Using the Mapping Collection Editor

See "Mapping Collection Editor" on page 173.

Scenarios Collection Editor

The figure below shows the Scenarios Collection Editor window. This Editor can be found in the Instance Property Grid.



➤ To add a member to the Scenario Collection, click **Add** (*).

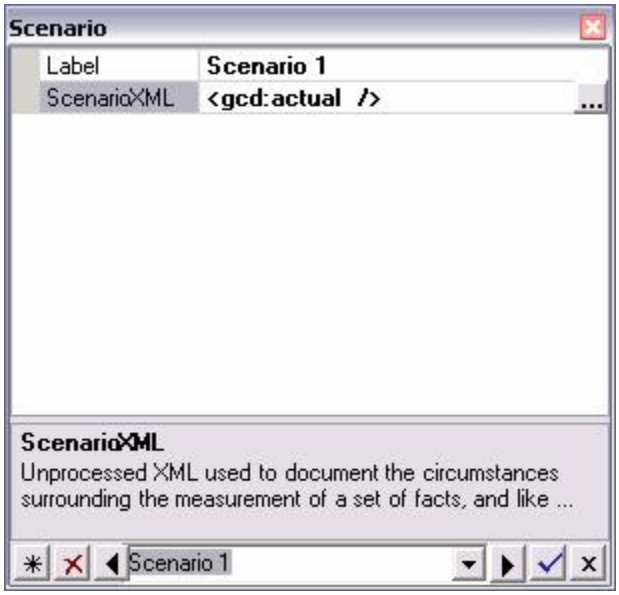
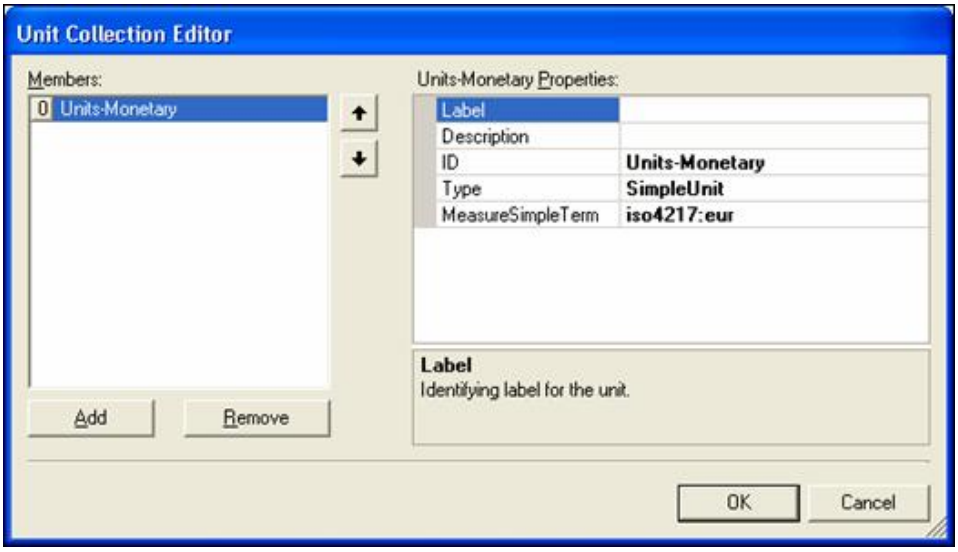


Table 67 Fields in the Scenario Collection Editor

Property	Description
Label	Human readable label for units which is used in combo boxes in the application. Each scenario label is unique to the instance document. For example, –Actual–.
Scenario XML	Well-formed XML fragment, as defined by the XBRL specification, which can be used within the scenario element of a context. For example, –<my:Scenario>Actual<my:Scenario>– **Note:** If the scenario XML utilizes entities not defined in a taxonomy (as data types), then an appropriate namespace must be added to the collection of namespaces for any scenario definitions, for example, to generate an –xmlns:my– construct..

Units Collection Editor

The figure below shows the Units Collection Editor window. This Editor can be found in the Instance Property Grid.



➤ To add a member to the Unit Collection, click **Add** (*).

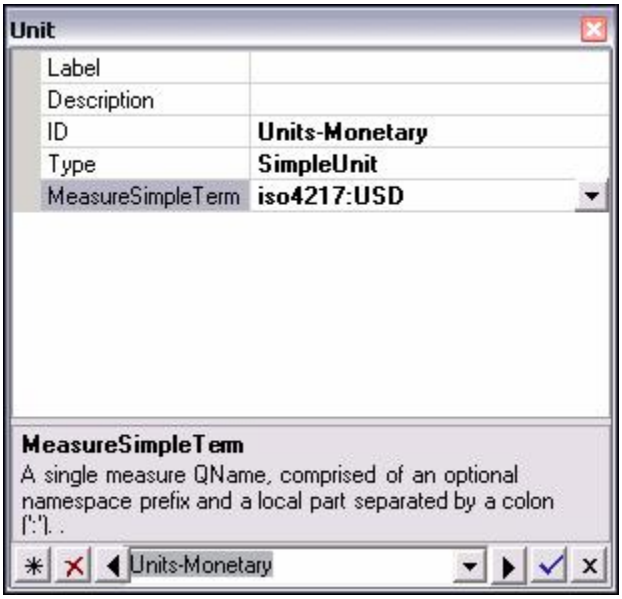


Table 68 Fields in the Units Collection Editor

Property	Description
Label	Human readable label for units which is used in combo boxes in the application. Each units label is unique to the instance document. For example, –Monetary–.
Description	Readable optional description which can be associated with each unit to make understanding the units easier. For example, –Monetary units in US dollars which should be used for all values.–
Units ID	XBRL Unit ID which is written in the instance document. Must be an XML ID. For example, –Units-Monetary–.
Type	Indicates if it is it single valued unit, a multiplication of terms, or division of multiplied terms.
Measure	Measure value. For example of single-valued unit, –iso4217:USD–, –shares–, –pure–. For multiplied or divided terms, an ellipsis button activates a collection editor of the multiplied, or divisor/dividend term collections.

Linkbases Collection Editor

The Linkbases collection editor shows the detached and linked linkbases, extended links, and extended link contents for each taxonomy in the DTS. This Editor can be found in the Instance Property Grid.

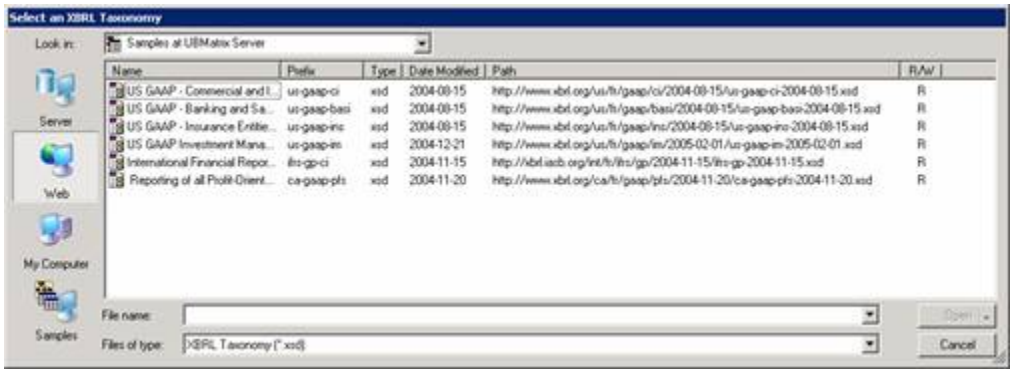
Extended link contents are relationship collections (presentation, collection, definition, and custom linkbases), and otherwise resource collections (labels, documentation, references, and formulas).

Open the linkbase collection editor from the taxonomy properties window or from reference taxonomies, select a schema from the DTS and then select Linkbases.

When the Linkbases collection editor is activated from the linkbases property of an opened base taxonomy, then the top taxonomy's linkbases are those shown.

When activated from a Referenced Taxonomies (DTS) Collection editor, any of the DTS taxonomies may be selected and their linkbases edited.

The linkbases collection editor manages a taxonomy's linkbases and, within each linkbase, extended links which contain labels, formulas, references, or relationship arcs.



The tree view of linkbases is opened from a selected taxonomy's properties; that taxonomy is the root of the linkbase collection tree. Below the taxonomy in the linkbase collection tree are linkbases containing at the next levels extended links, each containing at the next levels the respective extended links' contents (collections of labels, formulas, references, and relationships).

With the tree root taxonomy selected, its taxonomy property grid is shown in the right pane. (For convenience, you can click the taxonomy's reference parts collection to bring up any of the referenced taxonomies' linkbase collection editors while the subject taxonomy linkbase collection editor is also displayed.)

The first linkbase of each taxonomy is that which, when populated, is physically within the .xsd file in the appinfo section. In the default and most common situation, this linkbase is not utilized.

The usual and default linkbases are labels, references, relationships of presentation, calculation, and definition, and the formulas linkbase.

Linkbases can be linked or detached. When they are linked using a linkbaseRef from the schema, they are discovered during the loading process and are considered part of the DTS. When the linkbase are not linked, you can add them from the open dialog. See the Detached / Linked Linkbase section.

You can also Linkbases can also be editable (read/write) or read-only, from within the linkbase collection editor you can choose to make the linkbase editable or read-only. Note you can choose to make the schema, linkbases, and reference taxonomies editable or read-only from the load dialog or from the reference taxonomies dialog.

When a linkbase node selected on the linkbases tree view, the property grid contains the linkbase properties described in the table below.

Table 69 Linkbase Properties

Property	Description
Documentation	Contains documentation explaining the linkbase (for example, “Label links, English, standard role”. The documentation is saved in the linkbaseRef title attribute.
File Name	The filepath of the linkbase. For default linkbases, this is the taxonomy .xsd file name appended by the linkbase name. The file name may be a relative file (no c:\ or http:// prefix), a local disk path (for example, c:\something), or a URL http://something). If the taxonomy name changes, then that part of the file name which starts with the taxonomy file name is replaced. If the file name is a full path, when saved, the saver attempts to make it relative to the taxonomy full path.
ConnectionLink Count	Linkbases can be linked to the taxonomy schema, or detached and added manually during load. You can choose to attach or detach a linkbase to a taxonomy schema using this option. The number of labels, references, formulas, or relationships in the extended links of this linkbase.
ReadOnly	Linkbases can be editable (read/write) or read-only. You can set any linkbase to read-only, preventing any changes.
Link Count	The number of labels, references, formulas, or relationships in the extended links of this linkbase.

When an extended link of a linkbase is selected on the linkbase tree view, the property grid contains the extended linkbase properties described in the table below.

Table 70 Extended Linkbase Properties

Property	Description
Extended link Role	The extended link role is a selection from the taxonomy’s Extended Link Roles collection. Pressing the ellipsis button on this property activates the Role chooser/editor, described below.
Title	The extended link title attribute value. A title attribute is shown on the relationships tree, but is not unique to a role of extended link, as there may be multiple of the same extended links in other linkbases and in referenced or higher level taxonomies, with inconsistent titles.
Selection Formula	The formula that determines which links are allocated to the extended link. An absent formula means this is a default extended link which collects otherwise unallocated contents.
Documentation	Describes the purpose of the extended link.
ID	The extended link ID value. Generally not used.
Base	Feature currently not implemented

The Extended Link Role Chooser/Editor contains a property grid of the current extended link role, and a browser bar on the bottom. The browser bar allows adding (*) roles to the collection, deleting (X), scrolling, accepting (4) a choice, or canceling (X) for the extended link which activated the chooser/editor.

When an extended link is expanded (by clicking on the plus box) the contents of the extended link are shown — collection count of labels, references, formulas, or relationships.

Selecting the collection of the extended link provides a tabular view of the collection. This is a convenient way to review the collection.

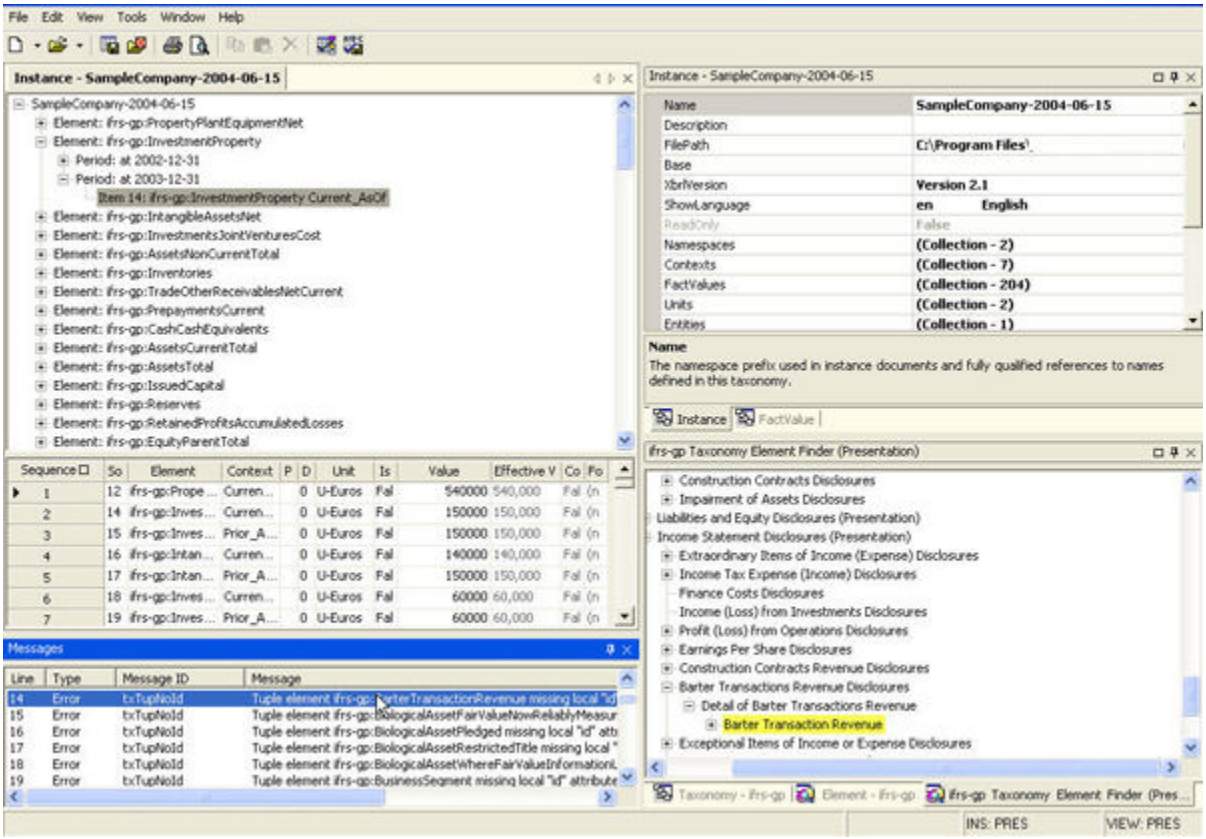
The screenshot above shows a label collection for the default extended link of the label linkbase. The collection shown has columns which correspond to the type of collection. For example, labels provide the label or documentation text, including longer strings in a text box below the data grid.

Instance Document Messages Pane

The Messages List pane shows error messages, warnings, and other informational messages from loading an instance document, validating an instance document, importing information, and other features where XBRL Taxonomy Designer needs to communicate to you.

As a process occurs, you can abort the process if errors are generated, allow the application to correct the errors, or enable you to correct the errors manually.

In the example below, an element was deleted and then the instance document was validated. The errors as a result of this can be seen in the message pane. By double clicking on one of the errors, you can highlight the source of the error in yellow in the Taxonomy Element Finder.



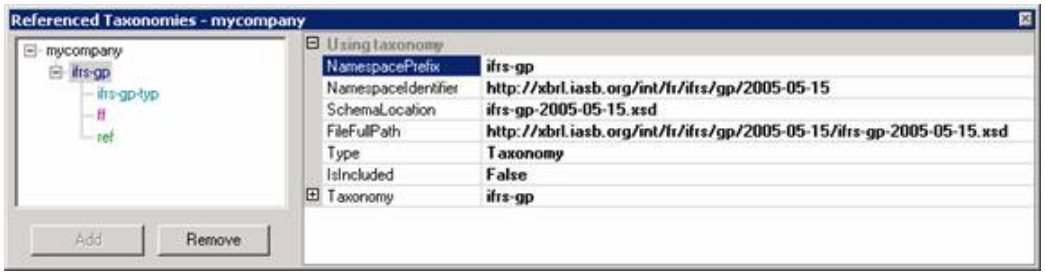
Detached and Linked Linkbases

Linkbases (Presentation, Label, Calculation, Definition, Formula) may be linked or not to an XBRL taxonomy schema using a linkbaseRef. When linkbases are physically linked or attached to a taxonomy schema, they are discovered during the loading process of that taxonomy and are considered part of the DTS. Linkbases that are not linked are referred to as detached.

➤ To add detached linkbases from the open dialog:

1 Select  and browse to select the detached linkbases added when loading the XBRL document.

Detached linkbases that are added during load do not have their Role Namespace Identifier filled out.



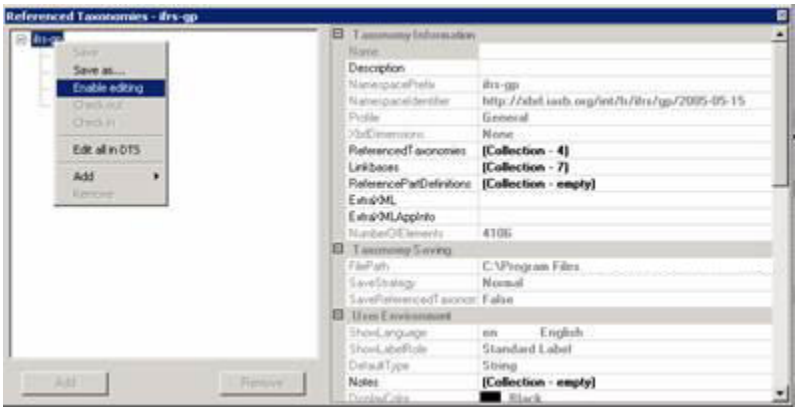
2 From the Taxonomy property window, select **linkbases** to open the Linkbase Collection Editor.

Select the added linkbase, and note that its Connection property is Detached.

3 **To attach this linkbase to the taxonomy set as LinkbaseRef, and then save the taxonomy.**

If a taxonomy or linkbase file is read-only, before saving the taxonomy change the file to editable. After the file is set to editable, saving the taxonomy attaches the linked linkbase files to the schemas.

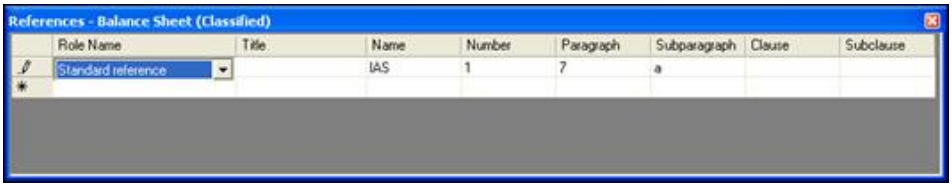
Screenshot of linkbase collection editor where you can set the editable and read-only flag on a linkbase. Setting the editable flag enables you to edit linkbase properties. To attach a linkbase to a read-only schema, set the editable flag for both the linkbase and the schema so that the appropriate linkbaseRef is added to the schema.



Use the Referenced Taxonomies dialog to make the taxonomy editable in the DTS view.

References Collection Editor

Taxonomy elements contain collections of references associated with that element. References can be edited from this list view. The reference parts defined for the taxonomy are the columns, or fields, which have potential entries for a reference. Examples of reference columns, or fields, are Name, Number, Paragraph.



Note that the reference parts are taxonomy defined. References and the concepts they apply to need not be in the same taxonomy. Several references for the same element could be in separate

taxonomies from each other. The selection of reference parts depends on the taxonomy owning the reference. Thus, it is possible that going down the lines of references, even for the same element, if they belong to separate taxonomies, the column headings change from line to line to reflect the reference's owning taxonomy.

References can be edited in the References Collection Editor, activated by the ellipsis button of a taxonomy element, and additionally within the Element Properties Grid by expanding the plus-box of the References collection property. Selecting the (+) sign next to each reference exposes additional property fields which may be edited.

Table 71 Property Fields for References

Option	Description
Role	Reference role assigned to this reference, based on the collection of reference roles available to this taxonomy. For example, "Reference" or "Definition" or "Example".
Title	Title which may be assigned to the reference. This value is not required by XBRL, it is optional.
Reference Parts: **Name, Number, Paragraph, Subparagraph, Clause, Subclause**	See "reference parts" which defines the fields which appear in the references list view.

Saving Instance Documents

This section explains the available options for saving an instance document.

Save/Save As (Local File System)

The Save as function enables you to save the instance document to the local file system. When File, then Save as is selected, the standard Windows save dialog screen is displayed:

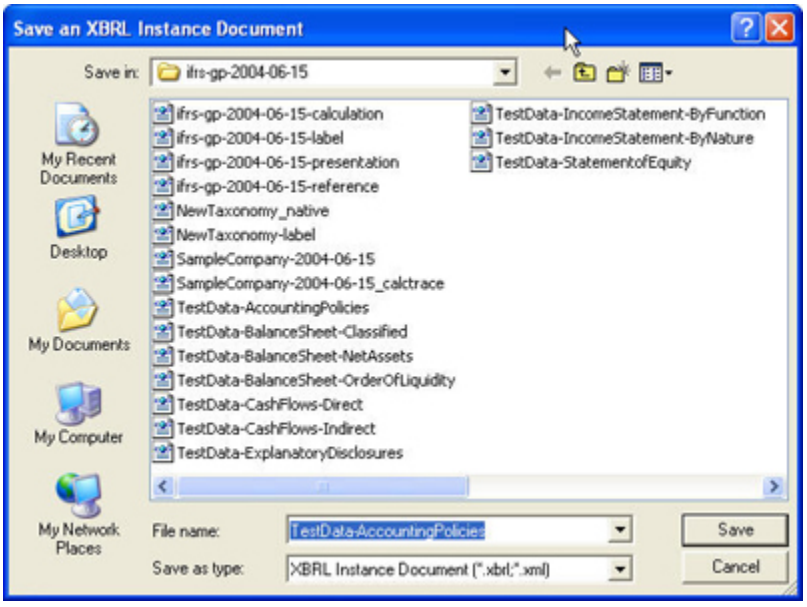


Table 72 Save As Dialog Box Fields

Option	Description
Save in	Indicates the subdirectory into which the file is saved.
File name	Indicates the name of the file to be saved.
Save as type	Indicates the file type to be saved. For saving instance documents, the file type is XBRL Instance Document (`*.xml`).

Save/Save As (Server)

The Save as function enables you to save the instance document to a server. When File, then Save to Server is selected, the standard Windows save dialog screen is displayed.

Table 73 Save As Server Dialog Box Fields

Option	Description
File name	This field indicates the location and the name of the saved file. If you are saving this file to a Web server, the URI of the file must be entered.
Save	Indicates the file type to be saved. For saving instance documents, the file type is XBRL Instance Document (`*.xml`)

Generating Instance Documents

This section explains the options available for generating an instance document.

Sample Instance Document Generator

An instance document sample can be generated from a taxonomy or part of a taxonomy.

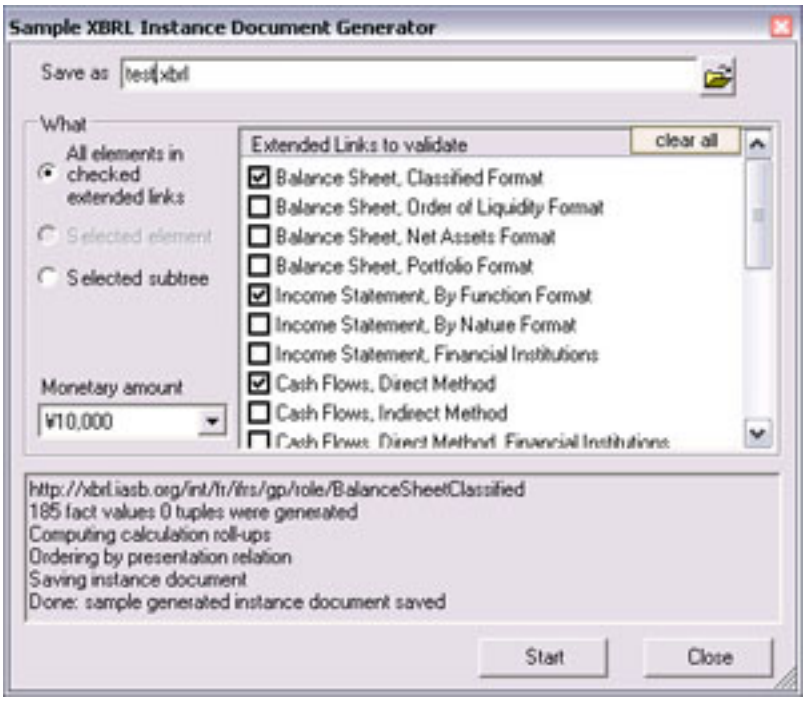
For a simple taxonomy one can simply generate instance facts for the whole taxonomy by selecting the All option and checking all extended links.

For a subset of a complex taxonomy, selected subtrees or selected extended links can be chosen (such as to generate for some small subtree of a balance sheet), or a full set of chosen extended links can be selected.

For example, for IFRS, one could check the all button, and check off one each of balance sheet, income statement, and cash flow extended links (as in the screen shot). A monetary unit can be chosen, such as thousands of euros, or hundred-thousands of yen.

For a detailed example of the use of the Sample Instance Document Generator, go to "Generating a Sample Instance Document" on page 146 .

➤ To access the **Sample Instance Document Generator**, select **Tools**, and then **Sample Instance Generator**.



When started, you specify an instance document file to which the sample is saved. The What section identifies what was selected on the relation tree, and how to proceed with generation.

Table 74 Save As Options

Option	Description
Save as	File path to save instance document upon completion of generation.
What Section	· All elements: Generate instance fact values for all applicable elements in the taxonomy · Selected element: Generate instance fact values for only the selected element (of the active relation view) · Selected subtree: Generate instance fact values for all applicable elements of the selected subtree (as noted in the accompanying text)

Start causes the identified elements (all, selected element, selected subtree) to be considered for generating instance document facts. If a selected element is a tuple, then a sample tuple and all of its items (sequence, all) are generated. If the selected element is an abstract element, it is skipped. Otherwise, if the selected element is an element outside of a tuple, a fact value is generated. Users are encouraged to modify the source code of this example for their own needs. An arbitrary string (“abc”) is generated for string facts. Decimal facts are given the value 1, and unit measure pure, monetary value 1, unit measure USD, and stock value 1, unit measure stock.

A numeric fact which participates in a calculation relation is adjusted to be the sum of the numeric facts which are generated in the instance and roll up to it.

The resulting instance document can be viewed and XML / XBRL validated by opening as an instance document. Use of element types other than the ones mentioned above may require modification of the plug-in source code. The plug-in can be re-compiled without any need for

XBRL Taxonomy Designer source code, and is simply put into the plug-in directory to be seen on startup.

Generating a Sample Instance Document

A taxonomy becomes fruitful when it yields an instance document. XBRL Taxonomy Designer sample instance generator automatically generates a template of an instance document, which can then be loaded into the instance document editor, and then validated, printed, formatted, and used to explain how your taxonomy works.

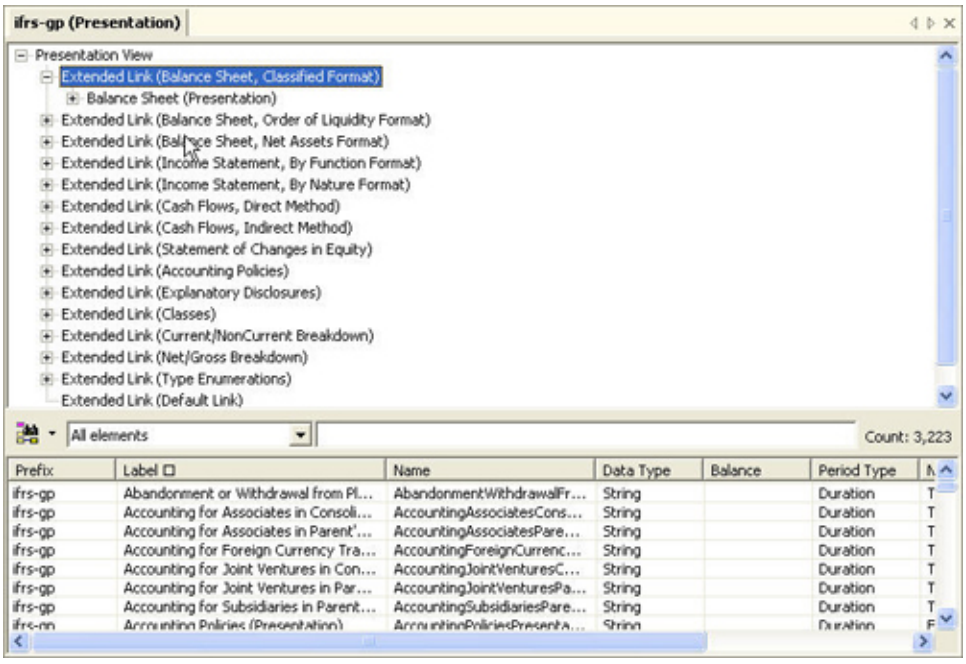
➤ To generate a sample instance document:

1 **Load the `ifrs-gp` taxonomy file into the Taxonomy Editor.**

Note: This file is saved in the Samples folder that was automatically installed when you first installed XBRL Taxonomy Designer . The default location for this folder is `C:\Program Files\Oracle\Disclosure Management\XBRL Taxonomy Designer\samples`, but it may be in another location if you installed XBRL Taxonomy Designer into another folder. See "Open an XBRL Taxonomy from the Local File System" on page 36.

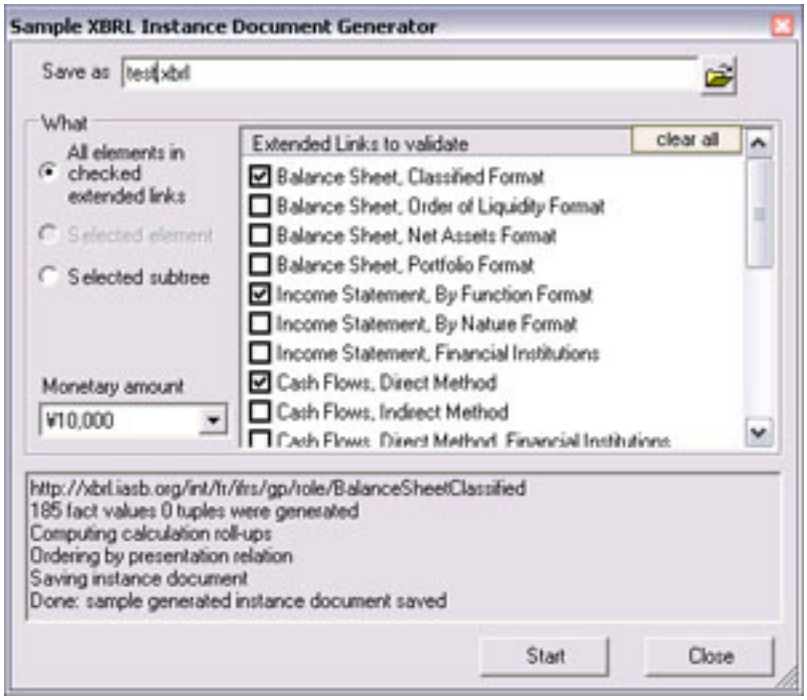
2 **Select a part of the taxonomy from which to generate a sample instance document.**

This could be a balance sheet subtree, or an Extended Link. For this example, choose this Balance Sheet subtree.



3 **Select Tools, and then Sample instance generator.**

The Sample XBRL Instance Document Generator window is displayed.

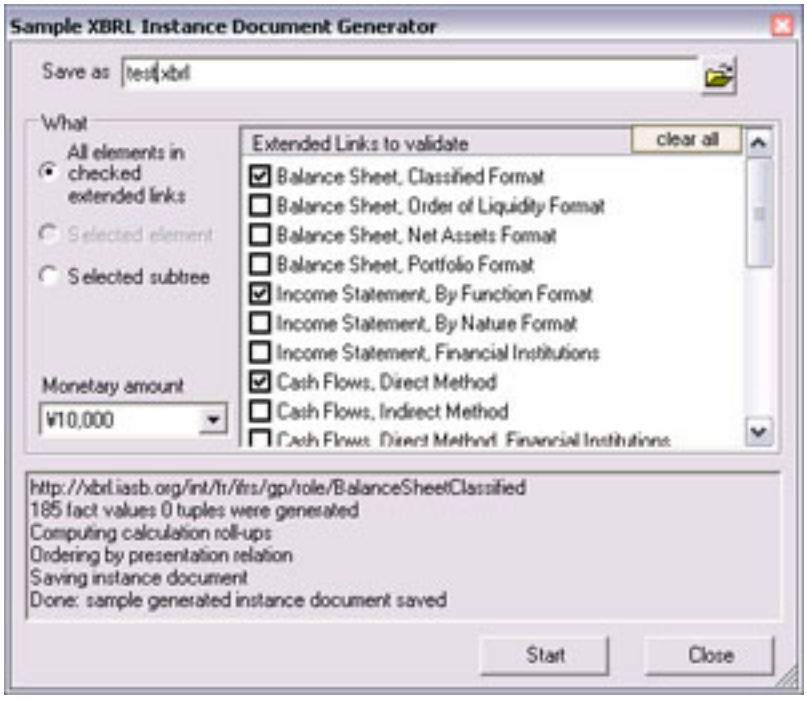


4 Specify a **Save as** file to receive the instance document.

For this example, use `temp.xml`.

5 In the **What** section choose **Selected subtree**.

6 Click **Start** to generate an instance document.

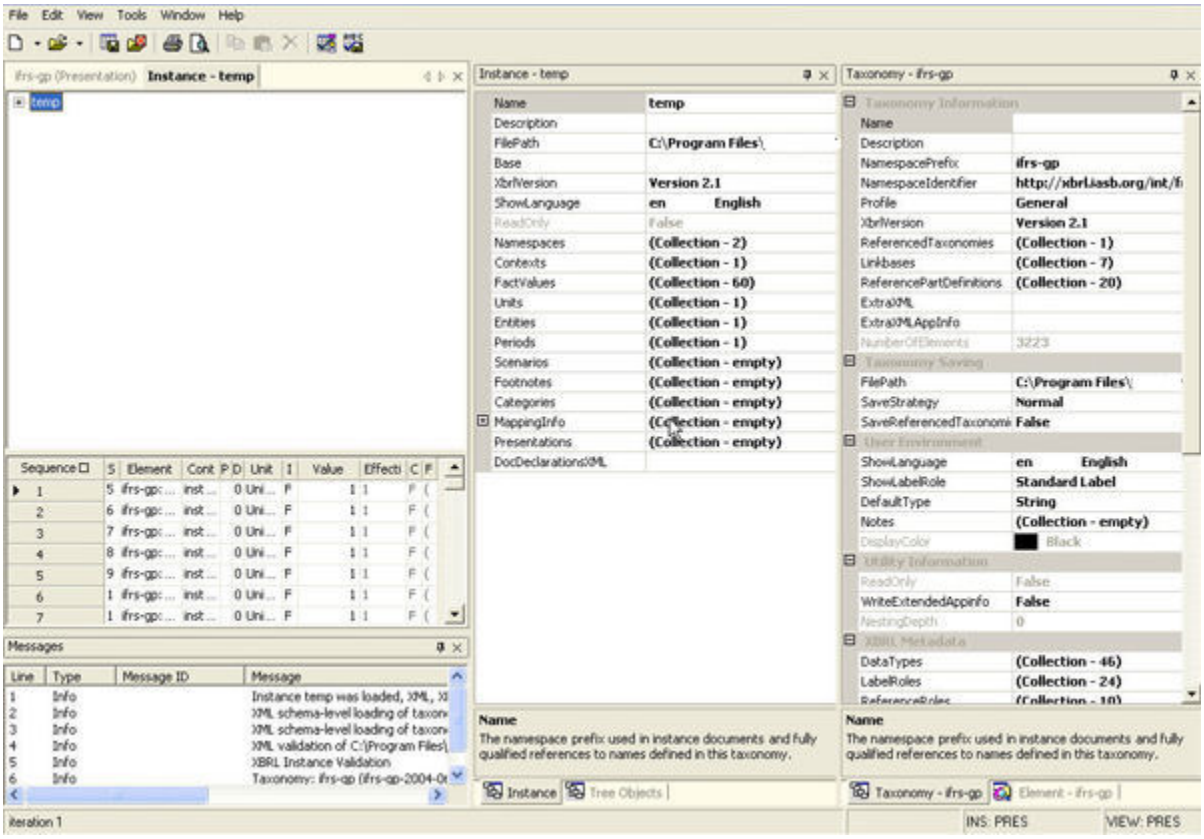


7 After the generation has completed, click **Close** to close the window.

Now we can take a view our newly-created instance document using the XBRL Taxonomy Designer Instance Document Editor screen.

➤ To do this, open the `temp.xml` file into your Instance Document Editor.

See "Opening an Instance Document" on page 116. The result should like the example below.



Validating Instance Documents

After you create an instance document, receive a file containing an instance document, or import instance data from another application, you can make sure there are no errors in the documents you create or receive by performing a validation.

The data in the instance document is validated to ensure it "foots and cross-castes" relative to the calculation links expressed in a taxonomy. For example, if Total Property and Equipment must equal Land plus Building plus Fixtures within a context, then the instance document is tested to ensure these values are accurate.

This is a sample validation to show the validation process. This is sample data represented in an instance document.

Figure 19 Validation Sample

	2003	2002
	€'000	€'000
Land	5,347	1,147
Buildings	244,508	366,375
Furniture and Fixtures	34,457	34,457
Computer Equipment	4,169	5,313
Other	6,702	6,149
Total	295,183	413,441

Figure 20 Output Report

Context/Item		Value	Source	Message
Context [at 2003-12-31 for SAMP Scenario 1]				
Total Commitments Under Non Cancelable Leases	C	2,421,000	inst	***error***
		2,112,000	comp	
(0) Not Later than One Year	C	309,000	inst	
(1) Later than One Year and Less than Five Years	C	1,420,000	inst	
(1) Later than Five Years	C	692,000	inst	
Context [at 2002-12-31 for SAMP Scenario 1]				
Total Commitments Under Non Cancelable Leases	C	2,666,000	inst	***error***
		2,369,000	comp	
(0) Not Later than One Year	C	297,000	inst	
(1) Later than One Year and Less than Five Years	C	1,439,000	inst	
(1) Later than Five Years	C	930,000	inst	

Note that the output presents a "trace" of the calculations in the instance documents. Errors are highlighted in red. The weights and balance types are presented on the report. Messages are provided to indicate errors encountered. The source of the values is provided, whether they come from the instance document, are computed, or come from the calculation of a formula.

➤ To validate an instance document, with the instance document loaded in the Instance Document Editor:

1 Select **Tools**, and then **Validate Instance Document**.

The Instance Validator window is displayed. Take a moment to look over the possible options. For this example, however, accept the default settings.

2 Click **Start**.

Validation is performed by the system and the validation output is presented in another window.

Context/Item		Value	Source	Message
Context [at 2003-12-31 for SAMP Scenario 1]				
Total Commitments Under Non Cancelable Leases	C	2,421,000	inst	***error***
		2,112,000	comp	
(0) Not Later than One Year	C	309,000	inst	
(1) Later than One Year and Less than Five Years	C	1,420,000	inst	
(1) Later than Five Years	C	692,000	inst	
Context [at 2002-12-31 for SAMP Scenario 1]				
Total Commitments Under Non Cancelable Leases	C	2,666,000	inst	***error***
		2,369,000	comp	
(0) Not Later than One Year	C	297,000	inst	
(1) Later than One Year and Less than Five Years	C	1,439,000	inst	
(1) Later than Five Years	C	930,000	inst	

3 Click the **X** on the header bar to close the window.

Analyzing Business Rules

Business rules analysis produces an instance document of fact values resulting from formula execution of the business rule extension taxonomy.

➤ To perform this validation, select **Tools**, and then **Analyze Business Rules**.

The Business Rules Analyzer window is displayed.

Table 75 Fields and components of the Business Rules Analyzer Window

Option	Description
XML Schema Validation	Performs XML Schema level validation on the instance document, using the Xerces validating parser (apache.org). Validates correctness of values, units, measures, scenarios, segments, and tuple structure.
XBRL Validation	Performs XBRL validation on the instance document. Performs context checks, calculation roll up checks, and executes formulas.
Compute missing values	During calculation linkbase roll up computation, if a sum of items is not present in the source data fact values, compute and use the computed sum. Identified by –comp– in the trace reports.
Keep computed values	Indicates if the computed values should be retained if the instance document did not provide them. For example, if the instance provides only leaf-node values the sum element values can be retained. Retained fact values differ from those of business rule execution, as they are merged to the instance document of source data and may be saved with the source data instance document (as well as used in subsequent processing, such as export or further import and subsequent further validation).
Show full audit trail	Results dialog, or trace file, provides a full audit trail of the calculation linkbase roll up processing.
Show formula trace	Results dialog, or trace file, provides full trace of execution of formulas, including steps to bind terms to source data, units matching, and exceptions raised if any.
Show results dialog	Outputs the validation results into a screen dialog (in addition to, or in lieu of, saving report file).
Save report in	Enables you to specify the path and name of a text file to which validation results are written.
Save trace in	Enables you to specify a file for the requested trace reports.

Option	Description
Instance document	For business rules, the result of the business rules are fact values which are written into an XBRL instance document, corresponding to the business rules extension taxonomy.

Validating Instances

XBRL Taxonomy Designer offers business rule validation and XML and XBRL level validation.

➤ To perform this validation, select **Tools**, and then **Validate Instance Document**.

The Instance Validator window is displayed. The Instance Validator window is similar to the Business Analyzer window described above.

See Table 75 for descriptions of the fields and components of this window.

Reading a Calculation Trace

A calculation trace organizes the results of all of the calculations of an XBRL document into the same tree view. It sorts the data by extended links and units hyperlinked to extended links and units (at bottom) and noting discrepancies where the addition differs from instance values representing sums.

For dimensions, a calculation trace organizes data by the tree of dimensional aggregation roll ups, following the aggregation trees. It creates aggregation trees for each of the cross-products of dimensions, varying one dimension at a time.

Each Calculation roll-up in the calculation trace is grouped by a network (Extended link role uri). Information is nested by context and unit.

Arcs are referred to as “summation-item” arcs. Summation-item arcs must represent relationships only between concepts that are in the item substitution group and whose type is numeric They represent aggregation relationships between concepts. Each of these relationships is between one concept, referred to as the summation concept, and another concept, referred to as the contributing concept.

A complete summation-item arc set for a given summation concept is defined in the context of the DTS supporting an XBRL instance. It is the set of all summation-item arcs, defined in calculationLink extended links with the same XLink:role attribute value that associate contributing concepts to the given summation concept. A summation item is an occurrence of a summation concept in an XBRL instance.

Calculation checks work exclusively on the information that is explicitly provided in the instance; items and values that can be inferred through essence-alias relationships are not considered. Several items (all corresponding to the one concept) can bind to a summation item if they are not duplicates because they are not period equal. This is relevant in the context of calculation scoping through tuples and means that detection of duplicates is not a sufficient test for double counting problems in XBRL instances.

The total of a binding calculation is defined to be the sum of the rounded values of the contributing numeric items in the binding, each multiplied by the value of the weight attribute on the item's associated calculationArc. This multiplication takes place after any necessary rounding is performed. The rounded value of a numeric item results from rounding the value of the numeric item to its precision or inferred precision (see Section 4.6.6 and 4.6.7). A binding calculation is consistent if the rounded value of the summation item equals the total rounded to the precision or inferred precision of the summation item.

5 Importing and Exporting

In This Chapter

Taxonomy Import Overview

To supplement interactive entry, taxonomy information can be directly imported into the application. Import formats include Microsoft Excel, Access, Word, an OLE DB Provider database, and CSV and XML files.

Importing taxonomy information is done by selecting File, and then Import. The Taxonomy Import window is displayed.

Figure 21 Figure 51 Taxonomy Import Window

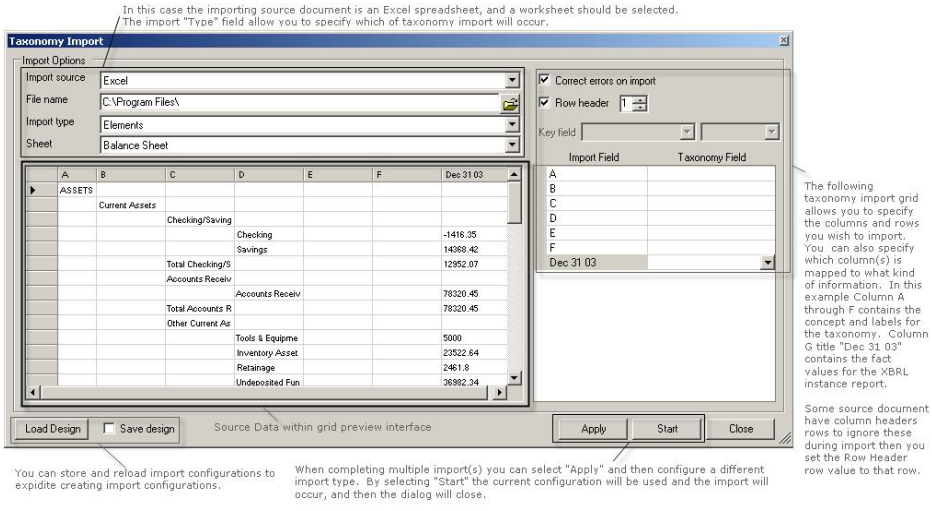


Table 76 Options in the Taxonomy Import Window

Option	Description
Import Source	Indicates the source of the import. Options include: Excel, Access, Word, Database, XML, and CSV. For CSV the field separator is provided under Tools, then Options.
File Name	Indicates the path and file name of the file which contains the import information. For database this is the database source (generated by the Browse button to the right).
Import Type	Indicates the type of import that is performed: ● General import—Enables you to import general information about a taxonomy, including elements, labels, references and relationships ● Labels—Taxonomy element labels ● Presentation relations ● Calculation relations ● Definition relations ● References ● Formulas ● Tuple
Sheet (or table, bookmark, element)	Indicates the Excel workbook which contains the data, or the Access/Database table name, Word-book-marked table, or XML top element name.
Correct errors on import	Indicates that errors should be detected and corrected. Alternatively, if errors are encountered the export is terminated.
Row header	The row of data (in a spreadsheet or CSV file) is the column headers. The adjacent spinner control is the line number of the header row to ignore any titling rows above that line. Not applicable for other import formats.
Save design	Enables you to save an import template so the import can be repeated. Saves the import type, sheet/bookmark/ table name, row header information, key field, and input field to taxonomy field mappings in an xml file. A Save as file name Browse button is presented when the Apply or Start buttons are pressed so multiple import mappings can be saved.
Load design	Enables you to load an import design that was created previously by the Save design feature, or by the external tool environment. The Load Design button presents a file chooser, and then sets the control information of import type, sheet/bookmark/table name, row header information, key field, and input field to taxonomy field mappings from the selected xml file.
Key field	The key field to used to determine which taxonomy element with which to associate the updated information. Not applicable when importing into an empty taxonomy.
Source Data Grid	The source data grid in the lower left corner of the form shows the source data and allows selection of header rows and mapping information visually.
Import Field	Column or field name of each import data source.
Taxonomy Field	The taxonomy field into which the import field data is placed. Allows for dynamic mapping of the import source to the element.

The import functionality applies reasonable data checks to determine allowed data and operations. Disallowed data is not imported.

If a receiving taxonomy is empty, all records attempt to create entries. If a receiving is not empty, import data is used to update items, and add items, based on the records in the import. If import items do not match elements, labels, and so on, then you are asked whether to add such items from the import data to the application. Error messages inform you of detected errors.

The import types correspond to the spreadsheets, database tables, CSV and XML outputs of export; however, there is an input field to the taxonomy field mapping process. This allows importing from sources with column names which differ from those which are output by export.

For the supported columns by input type, see "Exporting Taxonomy Overview" on page 157 for general, elements, labels, relations, references, formulas and tuples.

The mapping grid is automatically populated by an automap feature that looks for recognizable input columns or field names. The mappings should be reviewed and corrected as needed before starting import.

General format can contain multiple label and parent columns (Taxonomy Builder 5 format), or multiple rows with additional columns specifying language and role (XBRL Taxonomy Designer export format). Automap identifies multi-column input by the word label (case and spacing insignificant), "-", and the role name. Automap identifies multi-column language by "(en)" or "(fr)" patterns after label or label-terse (en). The property grid can override automap discovery.

On general and label import, the label import field is a special field with a number of items (clicking the plus box expands the list on a label import).

A label field might be indented over several columns. In some cases the indent implies parent-child tree relationships (General input type).

A language can be associated with a label in three ways:

- **Name input field**—Another column or field has the name of the language, for example English. Choose the column/field containing the language name under Language Value.
- **Code input field**—Another column or field has the code of the language, for example en. Choose the column containing the code of the language under Language Value.
- **Value selection**—No columns contain the language, but specify the language by choice of languages for this column/field in Language Value. For example, select English or French for this column.

A role can be associated with a label in three ways:

- **Name input field**—Another column or field contains the name of the role, for example Terse. Choose the column/field containing the role name under Role Value.
- **URI input field**—Another column or field has the URI of the role, for example http://www.xbrl.org/2003/role/terseLabel. Choose the column containing the URI of the language under Role Value.
- **Value selection**—No columns contain the role but they specify the role by choice of role names for this column/field in Role Value. For example, select Terse for this column from the Role Value drop-down selection list.

Parent relation is similar to label, in that it expands to specify arc role and extended link role. These can be taken from a column or selected from a list for the column. There can be multiple parent relation columns, just as for multiple label columns, where different columns have a different role or extended link role. For example, multiple columns could be used for presentation and calculation relationships.

An arc role can be associated with a label in three ways:

- **Name input field**—Another column or field contains the name of the arc role, for example, Child to Parent, or Item Summed to Summation. Choose the column/field containing the role name under Arc Role Value. Note that this is a child to parent role.
- **URI input field**—Another column or field has the URI of the arc role, for example, http://www.xbrl.org/2003/arcrole/child-parent. Choose the column containing the URI of the language under Arc Role Value.
- **Value selection**—No columns contain the arc role, but specify the arc role by choice of arc role names for this column/field in Role Value. For example, select Item Summed to Summation for this column from the Role Value drop-down selection list.

Extended link role can be associated with a label as follows:

- **Name input field**—Another column or field contains the name of the extended link role, for example, Default Link. Choose the column/field containing the role name under Ext Link Value.
- **URI input field**—Another column or field has the URI of the extended link role, for example http://www.xbrl.org/2003/role/link. Choose the column containing the URI of the language under Ext Link Value.
- **Value selection**—No columns contain the arc role, but specify the arc role by choice of extended link role names for this column/field in Ext Link Value. For example, select Default Link for this column from the Ext Link Value drop-down selection list.

Extended Link Title input field—Select an extended link title column/field name from the selection list, if any (else leave blank).

Key field specifies how to identify elements in a taxonomy when imported data is intended to update elements. In this case the key field specifies whether to match on Name, Label, and so on. Furthermore, there are key field matching options of exact match or comma-separated options matching. With comma-separated options, only one of the comma separated keys in the input data for the key field column need match the element to locate it.

In some cases, import field to taxonomy field mapping is automatic, such as when exported XBRL Taxonomy Designer or Taxonomy Builder 5 tables/files are re-imported and column headers are as expected by XBRL Taxonomy Designer. In other cases considerable design can go into setting up the mapping. For these cases, the mapping design can be saved. Check Save design to capture the import settings to an XML file. When this is checked, Apply or Start provides a Save As window for the design settings XML file.

➤ To reload design settings, click **Load Design**.

These design settings can also be used on importing by external program, such as a VB batch processing program or Web service.

Exporting Taxonomy Overview

To interface with external tools and products, taxonomy information can be exported to a variety of formats such as Microsoft Excel or Access, OLE DB provider Databases, and to CSV or XML files. This feature might be used to transfer data to other applications, export data for bulk update in another application and subsequent import into the XBRL Taxonomy Designer application, etc.

Exporting taxonomy information is done by selecting File, and then Export.

See "Formats for Exporting" on page 161.

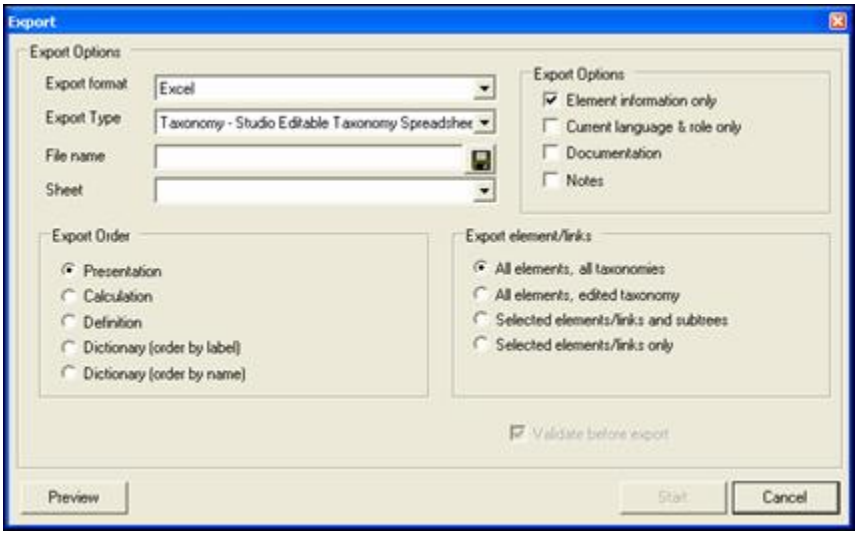


Table 77 Options in the Taxonomy Export window

Option	Description
Export format	Indicates the export format. Supported formats include: ● Excel ● Access (Existing or new database) ● OLE DB Provider database ● XML ● CSV
File name	Indicates the export file path and name, or the database source name. Use the file directory icon to specify or select the Excel file to which the information is saved. Click Start to begin the export.

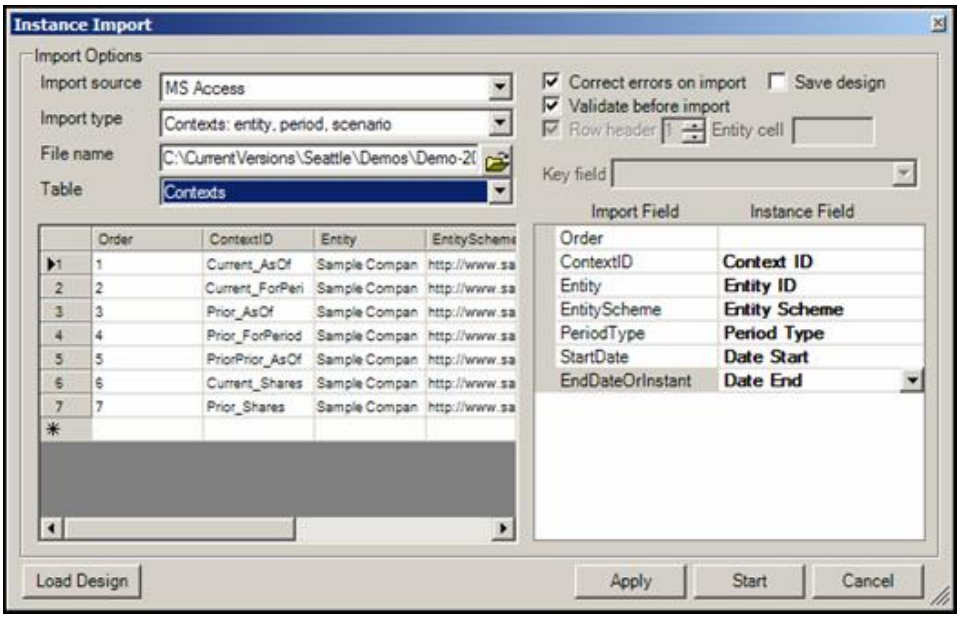
Option	Description
Export type	Indicates the export type. Export types include: ● Taxonomy—General Export—an inclusive format containing element information, the currently selected label, relationships and references (unless taxonomy information only is checked). Multiple rows are displayed for an entity with multiple labels, relations, or references ● Taxonomy—Tree View—Tree view export of taxonomy information. For Excel this includes exporting VBA code to automatically synchronize dictionary and tree views of elements. ● Elements—Taxonomy concept elements ● Labels—Export taxonomy labels, including multiple labels and documentation elements for each concept element ● Presentation relations ● Calculation relations ● Definition relations ● References ● Formulas ● Tuple structure ● All—Exports multiple worksheets, tables, CSV, or XML elements for element, labels, relationships, references, formulas, and tuples. For CSV, multiple files are output. Taxonomy Editor—Edit list contents are a re-importable format for Excel.
Sheet (or Table or Top Element Name)	Specifies the Excel worksheet name, the table name if Access or Database is selected, or the top element name if XML is selected. This option is not used with CSV.
Export order	Indicates the export order. The order of exported elements can be: ● Presentation link ● Calculation link ● Definition link ● Dictionary order by label ● Dictionary order by name
Element Options	Element options include: ● Element information only—Exports only element information and no information of relationships ● Current language and role only—Exports only the currently selected based on the language and label role settings ● Documentation—Exports taxonomy documentation in the export
Export elements and relationships	Specifies the export scope as one of the following: ● All elements, all taxonomies ● All elements, edited taxonomy ● Selected elements and subtrees ● Selected elements only
Preview	Displays the information that will be exported to the external file.
Start	Starts the export. The status bar indicates the progress of the export.
Cancel	Cancels the export.

Importing Instance Documents

Import functionality enables you to import instance document information from common data formats such as Excel, XML, Access, OLE DB Provider Databases, CSV, and Word table data.

In the batch processing import mode. data can be obtained from multiple sources, imported, validated, and then used to generate an instance document.

➤ To import instance document information, select **File**, and then **Import**.



XBRL Taxonomy Designer comes with standard import templates and automatically maps an import format to the appropriate instance field. The data field names in the import format are used to help determine the import field to which the data value is to be applied; or users can create their own import formats and simply map their data field names to the instance fields. These mappings can be saved and rerun in the future.

Exporting Instance Documents

You can export instance document information to common data formats such as Excel, XML, Access, OLE DB Provider Databases, and CSV tables.

The three output tables an export generates are Context, Unit, and Fact Values. Exported information is sufficient to import and reestablish the same instance document if the same taxonomy and supporting schema files are available.

➤ To export instance document information, select **File**, and then **Export**.

The Export Data window specifies export format and type, and provides an preview option that lists what information will be exported.

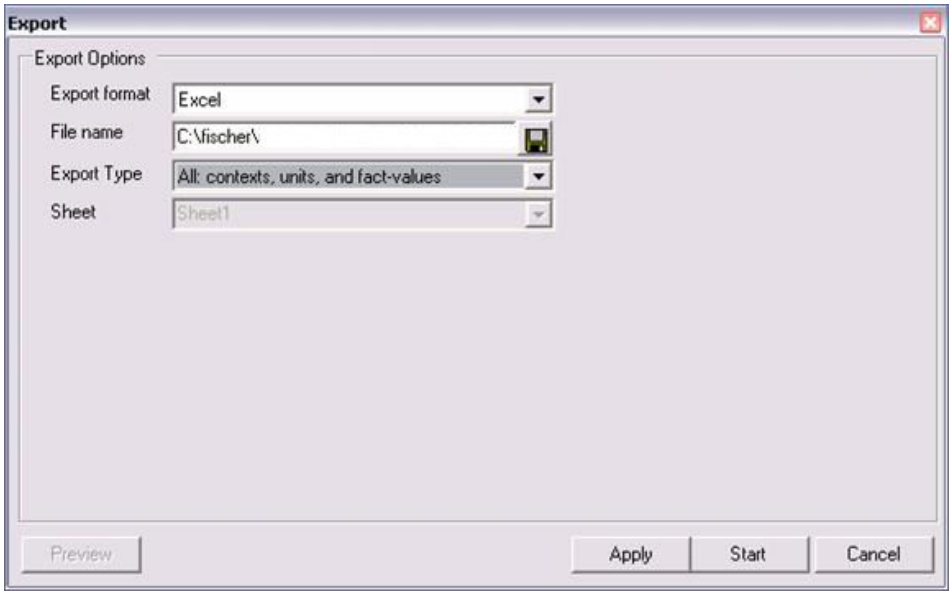


Table 78 Export Data Window Fields

Option	Description
Export format	Indicates the export format. Supported formats include: ● Excel ● Access (Existing or new database) ● OLE DB Provider database ● XML ● CSV
File Name	Indicates the path and file name of the export file. Or, if this is an OLE DB Provider Database, indicates the Database source name. You must use the file directory icon to specify or select the Excel file to which the information is saved. Simply typing in the Excel file name in the text box does not enable the Start button.
Export Type	Indicates the type of export that is performed: ● Contexts: entity, period, scenario—exports instance document context information ● Units: measure—exports instance document units information ● Fact values: items, tuple—exports instance document fact values and tuples All: outputs multiple worksheets, tables, CSV, or XML elements for Contexts, Units, and Fact Values. For CSV, multiple files are output.
Sheet (or Table or Top Element Name)	Indicates the name of the worksheet if Excel is selected. Indicates the name of the table if Access or Database is selected. Indicates the top element name if XML is selected. Not used for CSV. For the –All– option, tables are named automatically and this feature is disabled.
Preview	Provides a pop-up table preview of the information that is exported to the external file. For the –All– option this feature is disabled.
Start	Starts the export process. The status bar indicates the progress of the export.
Cancel	Cancels the export.

Export formats are as follows:

- **Contexts**—Columns or fields are context ID, context label, context description, entity ID, entity scheme, entity segment, period type (instant, duration, forever), instant date, period start and end dates.
- **Units**—Column or fields are unit ID, unit label, unit description, and unit measure. For single valued measures, the measure is output as a simple term, for example, iso4217:USD or shares. For multiplication and division terms, this field is an XML expression.
- **Fact Values**—Column or fields are fact ID, tuple name (QName, if in a tuple), element name (QName), context Ref (context's ID), unit Ref (unit's ID), decimals or precision, and value.
- **All**—provides the tree outputs of Contexts, Units, and Fact Values. For Excel, multiple worksheets are output. For databases, multiple tables are output. For XML, multiple sections. If CSV is selected, multiple files are output.

Formats for Exporting

There are several types of export formats:

- **General**—provides maximal information in one table, including elements in taxonomy, relationships to parent elements, multiple labels, and multiple references. This format is intended for compatibility to Taxonomy Builder 5, and other tools. This view can be re-imported, but becomes tricky to edit when there are multiple references and relationships per element. (The taxonomy contents tables are generally easier to edit and process externally than this "all-in-one" format.)
- **Tree view**—provides a taxonomy tree view (Excel or XML). The Excel tree includes active VBA programming to allow expanding/contracting tree levels and synchronized worksheets displaying dictionary and tree view. Double-clicking Excel dictionary worksheet entries expands and make tree elements visible. The Excel tree view expands and contracts similar to the Taxonomy Editor's Relationship View.
  - Taxonomy contents tables—selectable tables, worksheets, or files of elements, labels, presentation, calculation, definition, reference, and formulas.
  - Elements list view—provides the Taxonomy Editor table contents of the Elements List view

Columns in Export Formats

This section describes how to work with columns in export formats.

General Export Types

- Sequence—An element sequence number (may repeat on multiple rows, where there are multiple associated rows, such as multiple relationships, labels, or references for the same element)
- Prefix—Element's taxonomy prefix

- Namespace Identifier—Element's taxonomy namespace identifier
- Name—Element name (auto-generated or manually entered)
- Data type name—From taxonomy metadata, prefixed only when from referenced taxonomy
- Balance type
- Period type
- Weight—Provided only when calculation linkbase is checked
- Label—Three columns with repeating rows for possibly multiple labels
    - Label—the label character string
    - Label Language—language code, for example, EN or JA
    - Label Role—the role, for example, Terse
- Level—The level in tree if presentation, calculation, or definition relationship checked
- Relationship (to parents, if calculation, or definition relationship checked, else if element information only not checked, to children), with repeating rows if needed
    - Order (in parent's children list, for parents only)
    - Name (of parent or child)
    - Arc Role
    - Whether prohibited (role to children only)
    - Extended link role
    - Extended link title
- Reference parts, with repeating rows if needed, one reference part column per reference part in element's taxonomy reference parts collection

Tree View Export for XML Format

The tree view type for Excel export format: includes two worksheets with special purpose VBA coding for tree worksheet expand/synchronized to dictionary worksheet

Elements

- Sequence—Element sequence number (may repeat on multiple rows, where there are multiple associated rows, such as multiple relationships, labels, or references for the same element).
- Namespace Prefix—Element's taxonomy prefix
- Namespace Identifier—Element's taxonomy namespace identifier
- Name—Element name (auto-generated or manually entered)
- Data type name—From taxonomy metadata, prefixed only when from referenced taxonomy
- Balance type
- Period type

- Nillable

Labels

- Sequence—Element sequence number (may repeat on multiple rows, where there are multiple labels for the same element).
- Namespace Prefix—Element's taxonomy prefix
- Namespace Identifier—Element's taxonomy namespace identifier
- Name—Element name (auto-generated or manually entered)
- Label—Label character string
- Label Language—Language code, for example, EN or JA
- Label Role—Role, for example, Terse

Presentation Relation

- Sequence—Relation sequence number
- Parent element—QName of the parent
- Child element—QName of the child
- Extended link role of the relation
- Extended link title of the relation
- Preferred label role if any, of the relation

Calculation Relation

- Sequence—Relation sequence number
- Sum element—QName of the summation
- Item element—QName of the term
- Weight—Multiplier of the term when summing
- Extended link role of the relation
- Extended link title of the relation

Definition Relation

- Sequence—Relation sequence number
- From element—QName of the arc source
- Child element—QName of the arc destination
- Arc Role—Arc role for this definition relation
- Extended link role of the relation
- Extended link title of the relation

References

- Sequence—Element sequence number (may repeat on multiple rows, where there are multiple references for the same element).
- Namespace Prefix—Element's taxonomy prefix
- Namespace Identifier—Element's taxonomy namespace identifier
- Name—Element name (auto-generated or manually entered)
- Reference role
- Reference parts, with repeating rows if needed, one reference part column per reference part in element's taxonomy reference parts collection

Formulas

- Sequence—an element sequence number (may repeat on multiple rows, where there are multiple formulas for the same element).
- Namespace Prefix—the element's taxonomy prefix
- Namespace Identifier—the element's taxonomy namespace identifier
- Name—the element name (auto-generated or manually entered)
- Formula role
- Formula expression

Tuples

- Sequence—Sequence number
- Tuple element—QName of the tuple
- Item element—QName of an item in the tuple (nested elements are noted by having an item which is a tuple itself, and subsequently items of the nested tuple
- Min occurs
- Max occurs

All

Not a format or type, but a request to output all of these worksheets, tables, XML sections, or CSV files: Elements, Labels, Presentation, Calculation, Definition, References, Formulas, and Tuples.

Taxonomy Editor

Elements List Contents: this has the columns in the control indicated, in whatever order they are (rearranged to) in the data grid.

Troubleshooting Import Errors

When an import does not succeed, XBRL Taxonomy Designer displays messages indicating the cause of the problem.

This example is from "Mapping Lesson One" on page 285.

Line	Type	Message ID	Message
1	Info		Import to SAMP completed: 1 errors, 0 warnings
2	Info		Rows read: 57, skipped: 1, imported: 56
3	Info		Check for duplicates will be based on: Label
4	Info		No elements in taxonomy, all elements will be newly added from Excel
5	Error	txImpEltNoLabel	Row 58 indented label not found, row ignored

The message pane reports that 57 rows were read and 56 were imported. In this example, the Row Header is set to 1, which specifies that XBRL Taxonomy Designer reads each row starting with row two. XBRL Taxonomy Designer stops on row 57, because that is the last row with information.

XBRL Taxonomy Designer also reports duplicates, if found. If no duplicates are found, XBRL Taxonomy Designer scans through the source document until it no longer finds anything to import. Note in the example that XBRL Taxonomy Designer indicates when it reads the final element.

When the import is finished, XBRL Taxonomy Designer displays the number of elements imported. If XBRL Taxonomy Designer does not detect any elements to import, the number of elements is zero.

6 Mapping

In This Chapter

Mapping Overview

XBRL Taxonomy Designer supports importing and mapping of data from external sources, such as accounting, survey, tax, and credit reporting systems. Importing is a general function for direct entry of labeled external data facts. Mapping adds matching criteria to associated and aggregate external data with taxonomy concept facts.

Much data that can be expressed using an XBRL instance document is contained in external systems. Using XBRL Taxonomy Designer import and mapping functions, you can design import and mapping interfaces, extract data from these systems, convert the data into XBRL, and then validate the data to ensure its accuracy.

Import brings in facts of data from non-XBRL data media, such as Excel worksheets, Word tables, Access tables and queries, CSV files, OLE provider databases (Oracle SQL queries, etc.), and generalized XML. Each fact of data imported must be identified with a taxonomy concept by coding in source media with the name or label of its taxonomy concept. Import also can associate an XBRL context by coding its identifier or date in the source media and optionally coding units (such as currency) in the source media.

External systems often categorize data with their own table keys or XML attributes (such as account codes), labeling systems, or multi-dimensional fields. Data may need re-aggregation or allocation to fit the data model of a taxonomy.

Mapping means identifying which taxonomy concepts correspond to a datum of information in the imported source based on patterns of source fields, such as account codes, or other text, such as account titles or labels, or based on its position in the media. XML data source field patterns are xPath expressions, allowing specification of key matching of sibling fields and xPath expressions identifying context. Mapping of numeric data provides a many-to-many association of imported data-to-taxonomy concept facts, with scaled posting to facts by taxonomy concept, possible application of context, unit, and decimals based on matching criteria. An audit trail reports the details of this activity.

After information is in an XBRL instance document, it has the data model of the associated taxonomy. To transfer to other organizations or to consolidate with other information, the source XBRL instance document may need to be converted into an XBRL instance document

of another taxonomy, data model, and accounting standard. AUTOMATOR's mapping and import functions likewise support XBRL instance mapping from a source to target taxonomy, with associated many-to-many mapping.

GUI or API

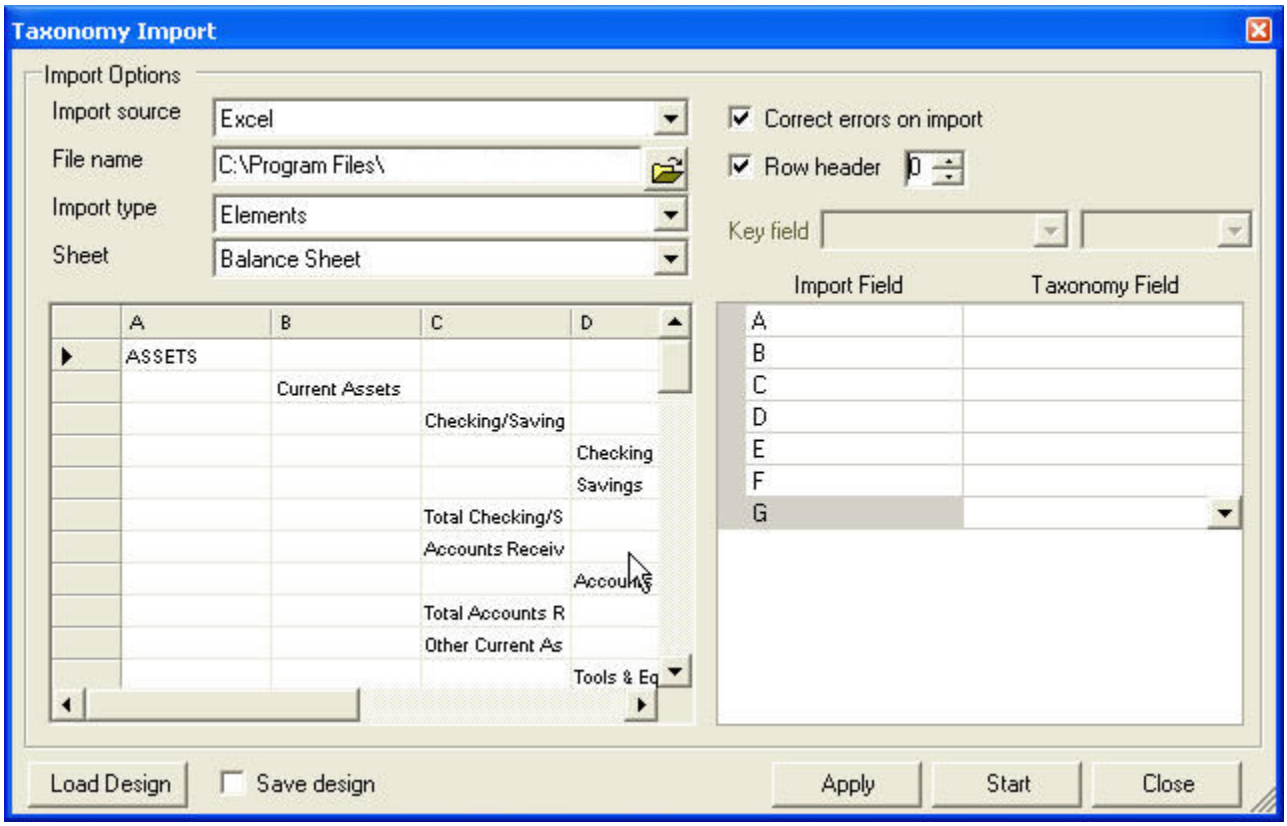
Mapping can be achieved through the GUI interface by users, but also through the AUTOMATOR DLL API for scripted and repeatable import mapping operations. In addition, plug-in functionality can be used to enhance application mapping features.

Design and Use

There are two phases of mapping:

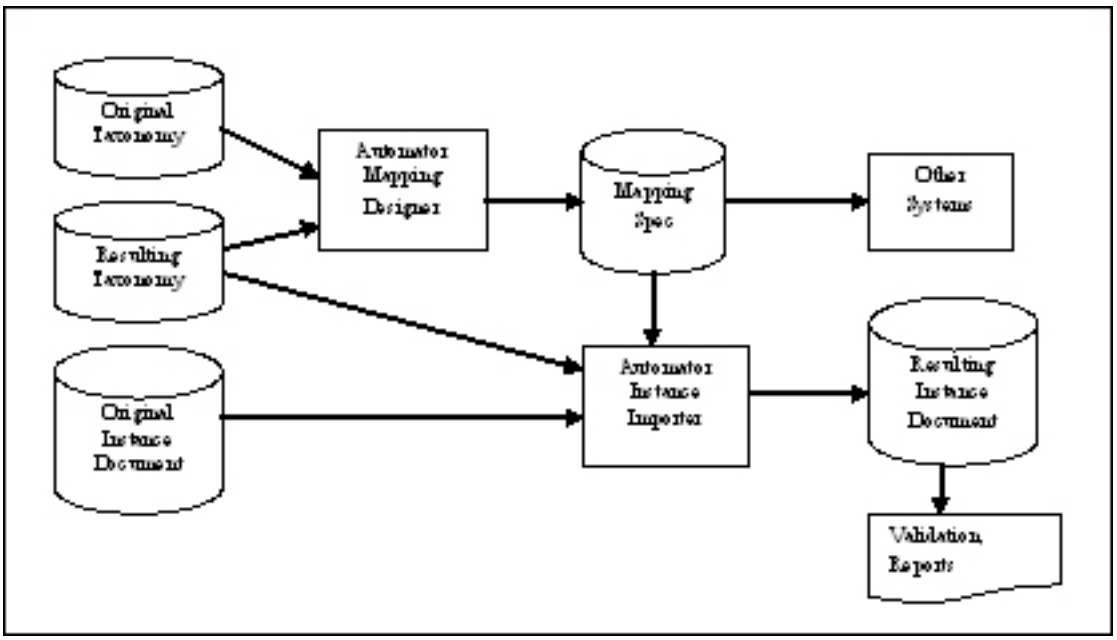
1. Design of XBRL mapping with taxonomies, and
2. Production import of data to instance documents and validation.

For a case where source data is mapped to taxonomy concepts, there is this general process:



1. The taxonomy of the instance document must begin mapping.
2. Next, a sample of source data or queries, or the instance document in a company chart of accounts is provided to the AUTOMATOR mapping designer.

3. The mapping designer provides a number of features to generate a mapping specification. The mapping specification can then be used in production importing.
4. Production importing by a scripted API program (or Web service) then uses the XBRL Taxonomy Designer import functions to read the accounting source data (or instance document under company chart of accounts) and, by using the mapping specification, to identify target taxonomy concepts to post scaled data into (or apportion by scaling to multiple concepts), as well as assign units, context, entity, scenario, and perform validation.
5. Some applications use only mapping as a design tool for other systems, without producing instance. For example one may design the interface between non-XBRL systems using mapping specifications to direct software interfaces without instances or validation. Other applications may import data to instance documents and may not need mapping design if the data is labeled for taxonomies, units, and contexts.
6. A general case involves both phases where data that comes from source systems requires XBRL mapping design, and production data are imported by the XBRL Taxonomy Designer Instance Importer into instance documents and validated.
7. If the data were a trial balance according to a company-unique chart of accounts, it could be imported to an instance document. If the company chart of accounts were identical to the taxonomy concepts, mapping may not be needed if there were a one-to-one association. Otherwise, mapping associates company chart of accounts codes to taxonomy concepts. Possible many-to-one mappings may occur if multiple sub-accounts in the trial balance aggregate to one taxonomy concept. After it is in the instance document based on the company chart of accounts, it could be left alone in that data model or possibly mapped from the company chart of accounts into IFRS or Jurisdiction GAAP by mapping the company-oriented instance document to the other taxonomies (taxonomy-to-taxonomy mapping).
8. Another approach would be to prepare mapping specifications for the accounting data to be imported with mapping into an instance document not based on the company chart of accounts, but instead mapped to another taxonomy (perhaps Jurisdiction GAAP). Mapping would associate company account codes with taxonomy concepts, providing scale factors, aggregating multiple source facts to taxonomy data model facts, and/or allocating by scale factor to multiple concept facts.
9. When mapping is used to convert a source taxonomy instance document to another target taxonomy, there is this process:



Multiple taxonomies can thus be handled by these use cases:

- **First case**—Source data is imported to appertaining taxonomy (by concept labeled source data or with mapping). For example, trial balance imports to taxonomy based on company chart of accounts. Subsequently, to provide an instance document to a jurisdictional taxonomy, this instance document is mapped from the appertaining to jurisdictional taxonomy.
- **Second case**—Source data is imported with mapping to jurisdictional-based or reporting taxonomy. For example, trail balance imports with mapping, by allocation and aggregation, to a company-tailored extension of jurisdictional financial reporting taxonomy or to a credit reporting taxonomy. No instance document is produced in company chart-of-accounts taxonomy.

XBRL Mapper

This section explains the XBRL Mapper.

XBRL Mapping Design

XBRL mapping expresses the relationship between an XBRL taxonomy concept, optional context and unit information, and a pattern of source data. An interactive design process enables you to inspect source data, determine basic mapping options (rows, cells, keys or key columns, and row groupings—for tables, xPath—for XML), and then relate taxonomy elements to rows, cells, or XML objects. In the process for tabular data it may be possible to specify contextRef, unitRef, and scale information, or contextRef and unitRef may be left for import field mapping. In the case of mapped XML objects, one can specify constants or xPath expressions to identify contextRef and unitRef for each datum.

There are three ways to associate a taxonomy element with source information:

- By drag-and-drop of a taxonomy element into the source document element.
- By automatic generation of best-matched taxonomy choices for each mapping and selecting from the best-fit choices.
- By manual entry.

Each mapping entry specifies the taxonomy concept, the filter expression for selecting source input, optional label and documentation (such as source system query and annotation), and instance document context, unit, decimals and scale factors.

Mapping supports source data mapping and taxonomy conversion of source instance document taxonomy to appropriate target taxonomy.

XBRL Mapper Design Operation

The basic XBRL Mapper design operation is to drag from the taxonomy tree or list of elements to an input source data table or input source XML tree.

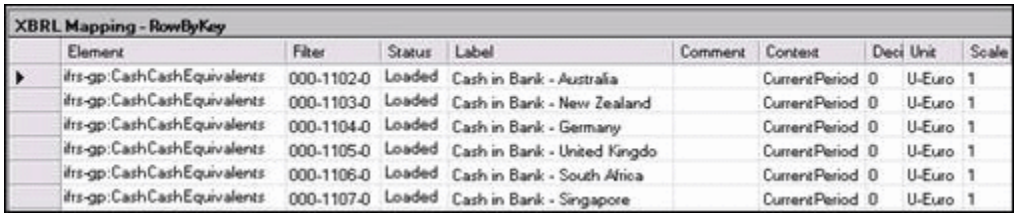
If one is mapping tabular rows by key or by ordinal, then a dragged taxonomy element may be dropped into any column of the row. The input source table row is colored to indicate it has a mapping specification in the lower mapping grid. To map one taxonomy element to a group of rows, first multi-select the group of rows in the source table data grid, then drop the appropriate element on any highlighted row. All selected rows are mapped to the same taxonomy element. Conversely, though less common, one may multi-select a group of taxonomy elements to map to one (or more) data table rows, such as when allocating a source data row value to multiple taxonomy accounts by fractional split values.

If one is mapping tabular cells by key and column header or by cell reference, then a dragged taxonomy element must be dropped onto the cell to be mapped. The mapped source table cell is then colored to indicate that it has a mapping specification in the lower mapping grid. There is no facility to multi-select cells for group mapping in this case.

If one is mapping XML tree objects then a dragged taxonomy element may be dropped onto the datum tree element. This automatically generates a filter xPath expression. Corresponding XML objects are then highlighted. If such a filter is not specific enough, you can edit the expression, such as by specifying a predicate expression of sibling element with an account code for a general ledger entry. Then the highlighted tree elements reflect only those matching the predicate expression (for example the same datum but for three years). You can also enter contextRef and unitRef xPath expressions (such as generating context-2004 by appending "context-" with 4 year digits from some sibling element of the value datum.

XBRL Mapping Design File

This screen shot shows the mapping file data grid for tabular source input



XBRL Mapping - RowByKey

	Element	Filter	Status	Label	Comment	Context	Deci	Unit	Scale
▶	ifrs-gp:CashCashEquivalents	000-1102-0	Loaded	Cash in Bank - Australia		CurrentPeriod	0	U-Euro	1
	ifrs-gp:CashCashEquivalents	000-1103-0	Loaded	Cash in Bank - New Zealand		CurrentPeriod	0	U-Euro	1
	ifrs-gp:CashCashEquivalents	000-1104-0	Loaded	Cash in Bank - Germany		CurrentPeriod	0	U-Euro	1
	ifrs-gp:CashCashEquivalents	000-1105-0	Loaded	Cash in Bank - United Kingdo		CurrentPeriod	0	U-Euro	1
	ifrs-gp:CashCashEquivalents	000-1106-0	Loaded	Cash in Bank - South Africa		CurrentPeriod	0	U-Euro	1
	ifrs-gp:CashCashEquivalents	000-1107-0	Loaded	Cash in Bank - Singapore		CurrentPeriod	0	U-Euro	1

The map file contains the mapping information and is stored as an XML file with the elements listed in Table 79:

Table 79 Map File Elements

Option	Explanation
Element	The XBRL concept element of the taxonomy being mapped to. For example, ifrs-ci:CashCashEquivents. On the display, for automatic mapping suggestions, “pending” choices are indicated by a combo box of suggested element names matched.
Filter	·The value from the key fields filled in automatically by dragging an element name onto the data source grid. For example: · Row by key. “001-1103-00” which is the general ledger account number · Row by row number. “4”. · Cell by row key and column name. “001-1103-00[2];Deferred”. See below. · Cell by cell reference. “C3” This field can contain * and ? wild-card expressions to generalize filters. For example for an account expression matching several keys, such as “00?-1103-*”, or several cell-row-groupings, such as “001-1103-00[*];Deferred”. Multiple filter column patterns are separated by a plus “+” character.
Status	An indication of the source of the mapping in the GUI data grid, not stored in the mapping file. The selections are: · Loaded—mapping loaded from file · Dropped—mapping dropped from taxonomy window drag operation · Pending—combo box of mapping concepts, user to make choice (line is highlighted in yellow pastel when pending) · Chosen—use has selected a choice from automatic choices After saved in a file, status becomes Loaded on the next taxonomy editing mapping design viewing session.
Label	An arbitrary label which can be contained in the mapping file to help you understand the mapping file and optionally serve as the label for the instance document. For example, “Cash – Operating”.
Comment	An arbitrary comment which can be contained in the mapping file to help you understand the mapping. This could be a database query, for example, if the source system obtained the data by query.
Context	If provided, a context of this mapping. If a context is provide in import data, such as columns for periods, then this field is left empty. If both are provided, this context overrides that provided in import data. Required for each entry when using cell mapping.
Decimals	The decimal value of the mapping to be used instead of an import data decimals field. This field can be left empty if decimals are specified in import data columns.
Unit	The unit reference. (Empty if provided in import data columns.)

Option	Explanation
Scale	The scale Enables you to “flip” the sign of the value or scale the value. For example, if the accounting system stores credit values as negative numbers, but the taxonomy requires the value to be positive in instance documents, assigning a scale of “-1” turns a positive value into a negative value. If an Excel spreadsheet contains values in millions, but the instance document requires the actual value, assigning 1000000 to the scale converts “15” in the Excel spreadsheet to “15000000” in the instance document. (If one is allocating one source table entry to multiple taxonomy elements, then fractional scale factors can be used, for example a scrip purchase for entertainment might be allocated “0.1” to charity and “0.9” to entertainment.)

Mapping Collection Editor

The figure below shows the Mapping Collection Editor window. This Editor can be found in the Instance Property Grid.

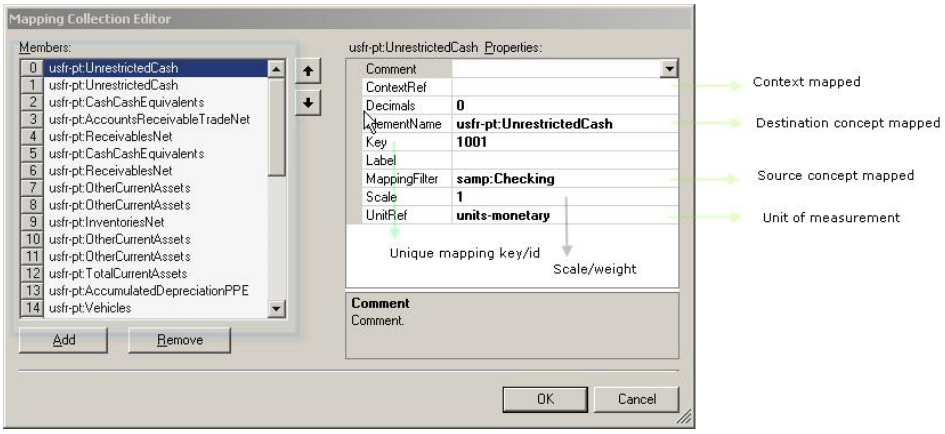


Table 80 Fields in the Mapping Collection Editor

Property	Description
Element Name	If the Mapping Filter is found in tabular data, the element name of the taxonomy to be mapped to. For example, –ci:PropertyPlantAndEquipment–. For XML Tree mapping to tuples, the path to the element name in the taxonomy tuple structure.
Mapping Filter	String, wildcard, or regular expression to match to tabular source data field for this element; for XML tree mapping, an xPath match expression to identify matching source data fields for this element.
Label	Descriptive entry of this mapping
Context Ref	If provided for tabular input, overrides import-determination of fact value’s contextRef. If provided for XML object input, provides an xPath expression producing a contextRef.
Unit Ref	If provided, for tabular input, overrides import-determination of fact value’s unitRef. If provided for XML object input, provides an xPath expression producing a unitRef.
Decimals	Decimals precision
Scale	Multiplier before adding (posting) to element fact value for this context (usually +1 or -1, but could be 0.6667 or anything else).

Mapping Cells by Row Key, Rows Group, And Column Name

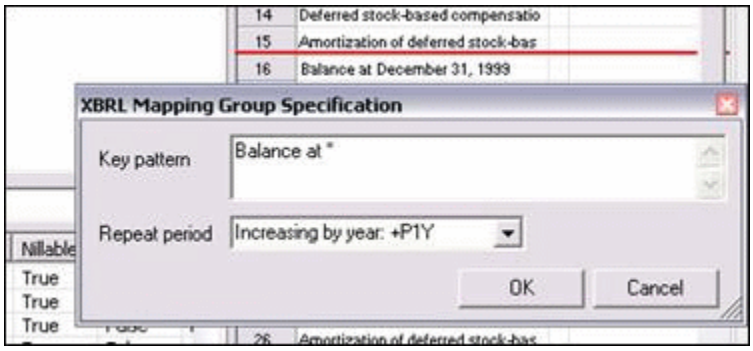
This option, for tabular input source data, allows handling source data, such as consolidated statements of stockholders' equity, with columns that map individually to different elements across the worksheet, and repeating groups of rows where usually each group of rows covers one reporting period.

If no repeating rows period groups are defined, then you can simply drag an element name to the source data grid. The mapping entry generated consists of the key column contents, [1] for group 1 (in this case the only repeating rows group), a ; and the column header, such as Deferred Compensation.

When there are repeating groups, you must first define where the repeating group starts. A typical example is: the first line is Balance at starting date, then some lines of gain or loss, then a Balance at the year end date, then more lines of gain or loss, etc., until a Balance at last year end date.

The group repeats at each Balance at date row.

Right-click any Balance at line, and choose Select Group Repeat. An XBRL Mapping Group Specification form shows the key field of that line (for example Balance at December 31, 1776) and any previously-selected repeat period. Change this to a repeating pattern, such as Balance at *. (Standard wild card characters * and ? are supported.)

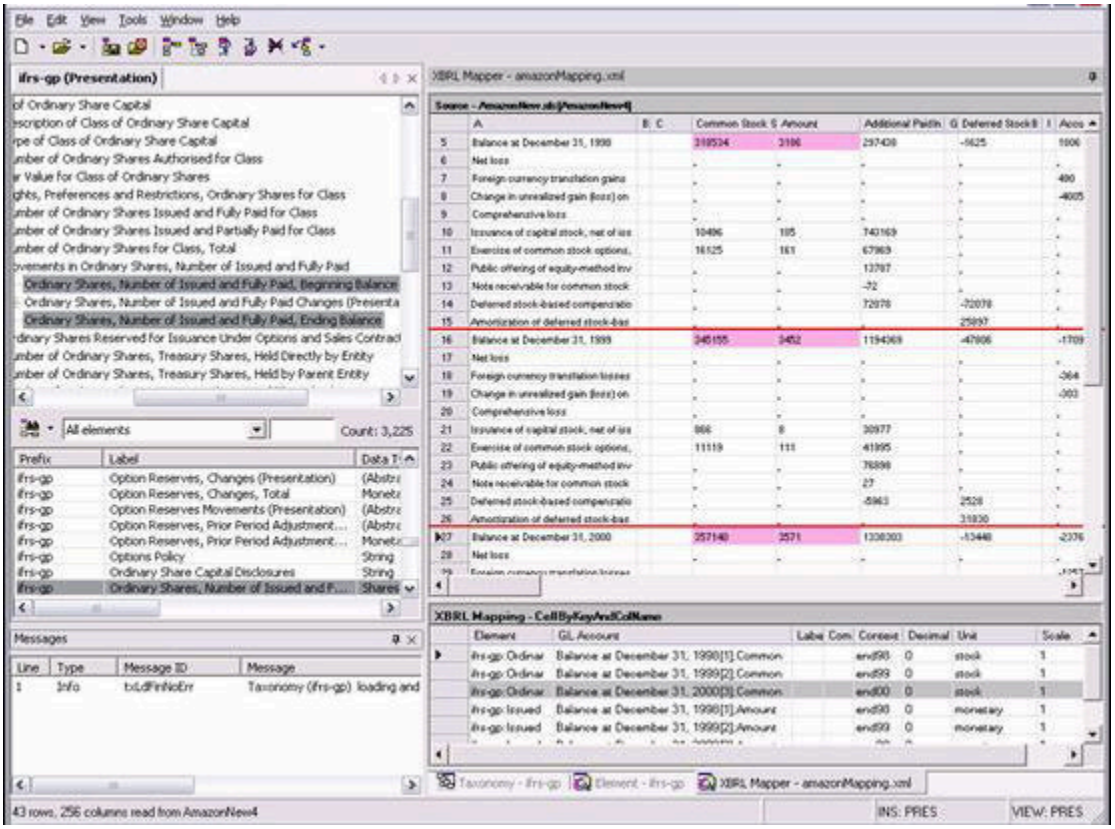


The Repeat period determines how the context period of the first repeating group is changed in each subsequent repeating group.

By using this feature only the first repeat period need be mapped. Subsequent groups can use the first as a template, offsetting their context in each period group. On the other hand, each and every cell in each and every group can be uniquely mapped, and then this feature is unused.

Many equity statements contain groups that increase by year in successive groupings of rows. The combo-box has period specifications for increasing and decreasing year, quarter, and month. A custom period can be entered as an XML duration expression, for example +P14D for two weeks (14 days) increasing periods, or –P7D for one week decreasing periods.

A red line is drawn between each repeating group.



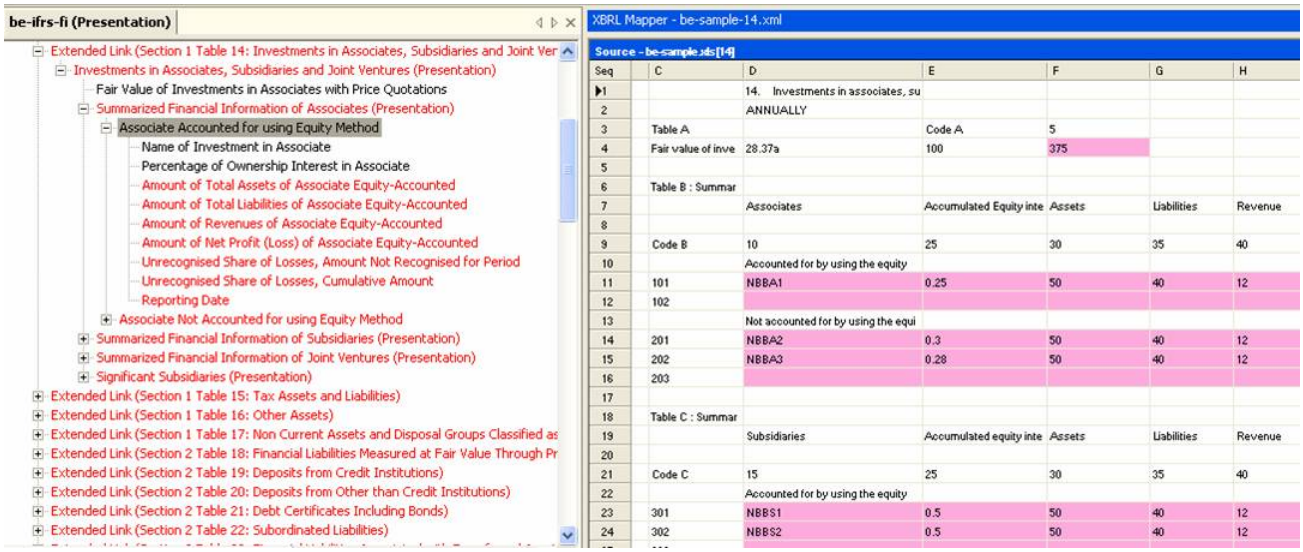
The match key generated is then in the form key column, [group number]; column header.

If a matching table entry is needed to match the same row of all repeating groups — in the case where context is not specified in the mapping table — then wild cards can be used in the "[group number]" expression, for example "[*]" means to match the given entry in all groups while importing. In this case the context period identifies the context of the initial repeating group and is increased (decreased) by the repeating period duration amount for each group.

Mapping By Transposing Cells and Rows

When each element of a tuple is the heading of a vertical column, there is no need to transpose rows or columns:

In Taxonomy	In Data Source (Excel)
Tuple:	Associate Accounted for using Equity Method
Element #1:	Name of Investment in Associate Column D
Element #2:	Percentage of Ownership Interest in Associate Column E
Element #3:	Amount of Total Assets of Associate Equity-Accounted Column F, and so on



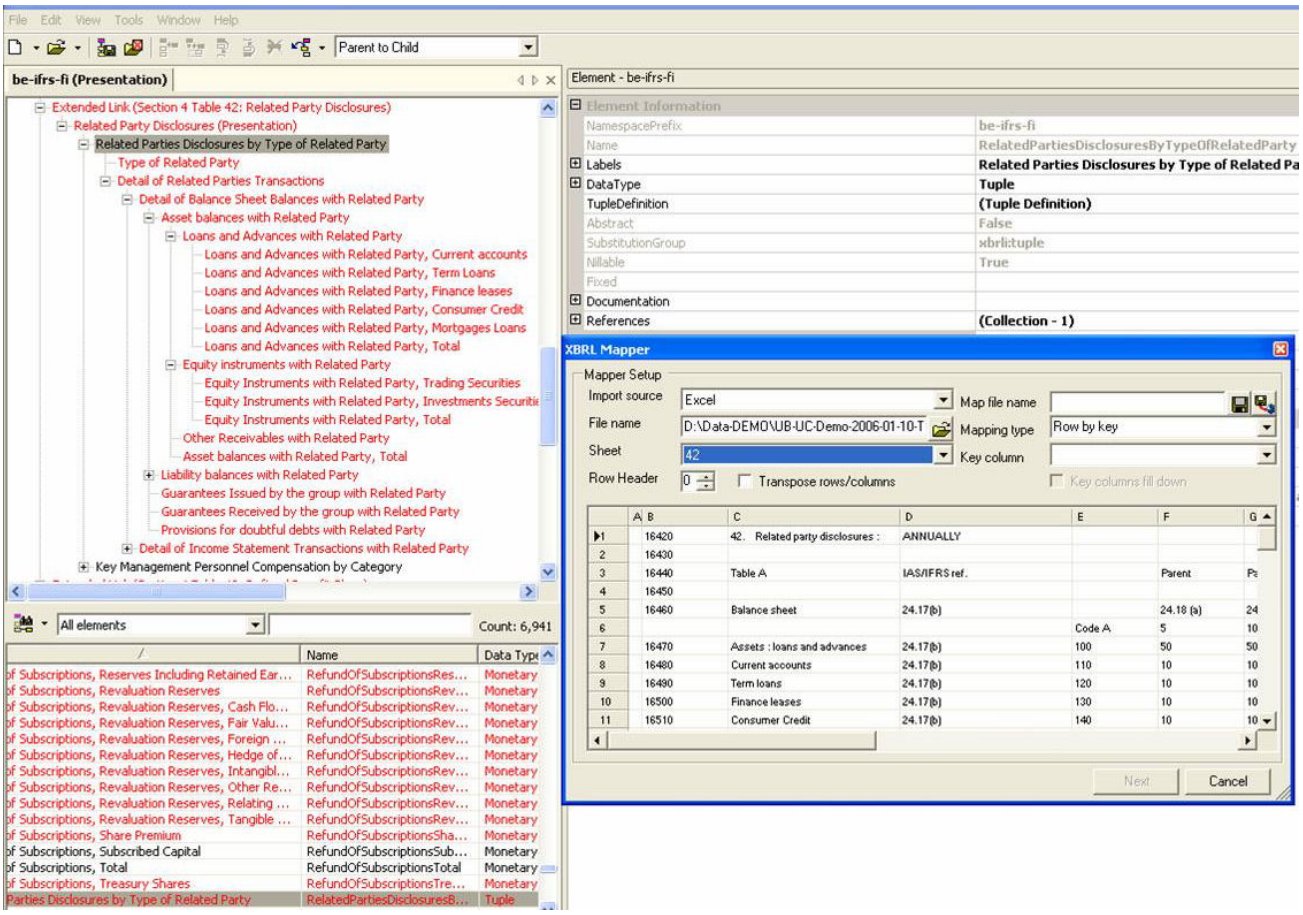
When the Tuple Elements are the headers of horizontal rows, XBRL Taxonomy Designer requires you to transpose rows/columns. In the example below: Current Account, Term Loans, Finance leases, etc.

Table A	IAS/IFRS ref.		Parent	Parent entities with joint control	Subsidiaries	Associates	Joint ventures where the entity is a venturer	Key management of the entity or its parent	Other related parties
Balance sheet	24.17(b)		24.18 (a)	24.18 (b)	24.18 (c)	24.18 (d)	24.18 (e)	24.18 (f)	24.18 (g)
		Code A	005	010	015	020	025	030	035
Assets : loans and advances	24.17(b)	100	50	50	50	50	50	50	50
Current accounts	24.17(b)	110	10	10	10	10	10	10	10
Term loans	24.17(b)	120	10	10	10	10	10	10	10
Finance leases	24.17(b)	130	10	10	10	10	10	10	10
Consumer Credit	24.17(b)	140	10	10	10	10	10	10	10
Mortgage loans	24.17(b)	150	10	10	10	10	10	10	10
Equity instruments	24.17(b)	160	20	20	20	20	20	20	20
Trading securities	24.17(b)	170	10	10	10	10	10	10	10
Investment securities	24.17(b)	180	10	10	10	10	10	10	10
Other receivables	24.17(b)	190	10	10	10	10	10	10	10
Total assets	24.17(b)	199	80	80	80	80	80	80	80
Liabilities: deposits	24.17(b)	200	20	20	20	20	20	20	20
Deposits	24.17(b)	210	10	10	10	10	10	10	10
Other borrowings	24.17(b)	220	10	10	10	10	10	10	10
Other financial liabilities	24.17(b)	230	40	40	40	40	40	40	40
Debt certificates	24.17(b)	240	10	10	10	10	10	10	10
Subordinated liabilities	24.17(b)	250	10	10	10	10	10	10	10
Share based payments	24.17(b)	260	20	20	20	20	20	20	20
Granted	24.17(b)	270	10	10	10	10	10	10	10
Exercised	24.17(b)	280	10	10	10	10	10	10	10
Other liabilities	24.17(b)	290	10	10	10	10	10	10	10
Total liabilities	24.17(b)	299	70	70	70	70	70	70	70
Guarantees issued by the group	24.17(b)ii	300	10	10	10	10	10	10	10
Guarantees received by the group	24.17(b)ii	310	10	10	10	10	10	10	10
Provisions for doubtful debts	24.17(c)	320	10	10	10	10	10	10	10

The Elements of the Tuple Related Parties Disclosure by Type of Related Parties

- Current Account
- Term Loans
- Finance leases
- Consumer Credit, and so on

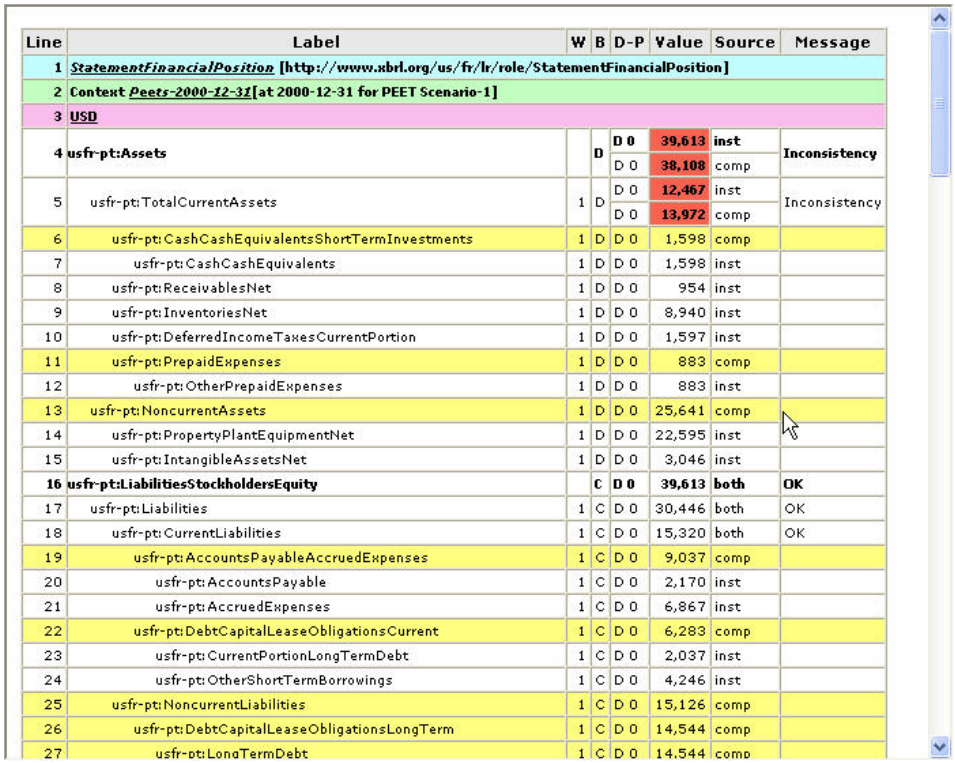
All of these are posted in Column C in the example above. To use this type of mapping, you must click the Transpose rows/columns in the XBRL mapper. The rest of the steps are the same as in "Mapping Lesson One" on page 285 .



Troubleshooting Mapping Errors

If not executed properly, mapping can cause errors. The quickly method to discover where these errors are located is to validate the instance document. The example below is a calculation trace for an instance document.

usfr-pt Assets and usfr-pt:TotalCurrentAssets each have an error, as marked by tagging their values in red. Each of them has the same error. The reported value does not match the computed value. This is an inconsistence as noted in the message column. As the computed value is higher, it is likely that an extra value was mapped to this concept.

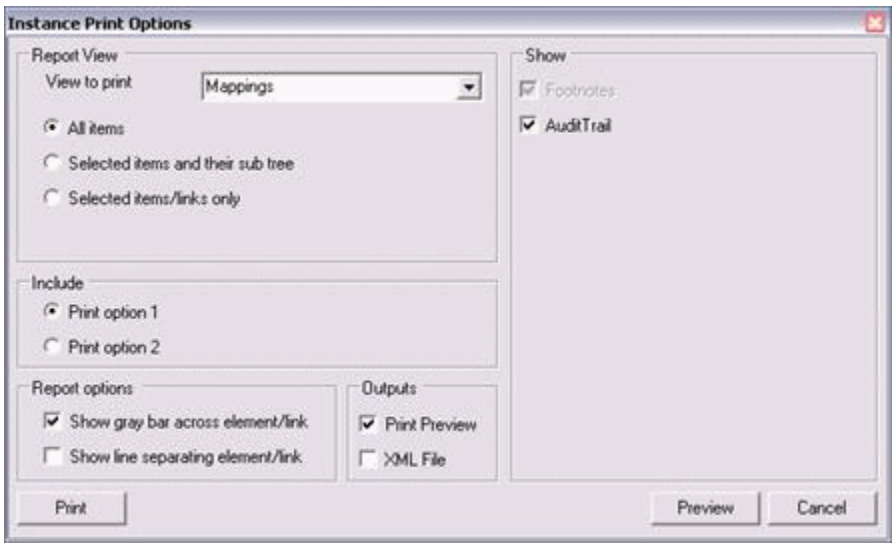


Line	Label	W	B	D-P	Value	Source	Message
1	StatementFinancialPosition [http://www.xbrl.org/us/fr/lr/role/StatementFinancialPosition]						
2	Context Peets-2000-12-31 [at 2000-12-31 for PEET Scenario-1]						
3	USD						
4	usfr-pt:Assets		D	D 0	39,613	inst	Inconsistency
				D 0	38,108	comp	
5	usfr-pt:TotalCurrentAssets	1	D	D 0	12,467	inst	Inconsistency
				D 0	13,972	comp	
6	usfr-pt:CashCashEquivalentsShortTermInvestments	1	D	D 0	1,598	comp	
7	usfr-pt:CashCashEquivalents	1	D	D 0	1,598	inst	
8	usfr-pt:ReceivablesNet	1	D	D 0	954	inst	
9	usfr-pt:InventoriesNet	1	D	D 0	8,940	inst	
10	usfr-pt:DeferredIncomeTaxesCurrentPortion	1	D	D 0	1,597	inst	
11	usfr-pt:PrepaidExpenses	1	D	D 0	883	comp	
12	usfr-pt:OtherPrepaidExpenses	1	D	D 0	883	inst	
13	usfr-pt:NoncurrentAssets	1	D	D 0	25,641	comp	
14	usfr-pt:PropertyPlantEquipmentNet	1	D	D 0	22,595	inst	
15	usfr-pt:IntangibleAssetsNet	1	D	D 0	3,046	inst	
16	usfr-pt:LiabilitiesStockholdersEquity		C	D 0	39,613	both	OK
17	usfr-pt:Liabilities	1	C	D 0	30,446	both	OK
18	usfr-pt:CurrentLiabilities	1	C	D 0	15,320	both	OK
19	usfr-pt:AccountsPayableAccruedExpenses	1	C	D 0	9,037	comp	
20	usfr-pt:AccountsPayable	1	C	D 0	2,170	inst	
21	usfr-pt:AccruedExpenses	1	C	D 0	6,867	inst	
22	usfr-pt:DebtCapitalLeaseObligationsCurrent	1	C	D 0	6,283	comp	
23	usfr-pt:CurrentPortionLongTermDebt	1	C	D 0	2,037	inst	
24	usfr-pt:OtherShortTermBorrowings	1	C	D 0	4,246	inst	
25	usfr-pt:NoncurrentLiabilities	1	C	D 0	15,126	comp	
26	usfr-pt:DebtCapitalLeaseObligationsLongTerm	1	C	D 0	14,544	comp	
27	usfr-pt:LongTermDebt	1	C	D 0	14,544	comp	

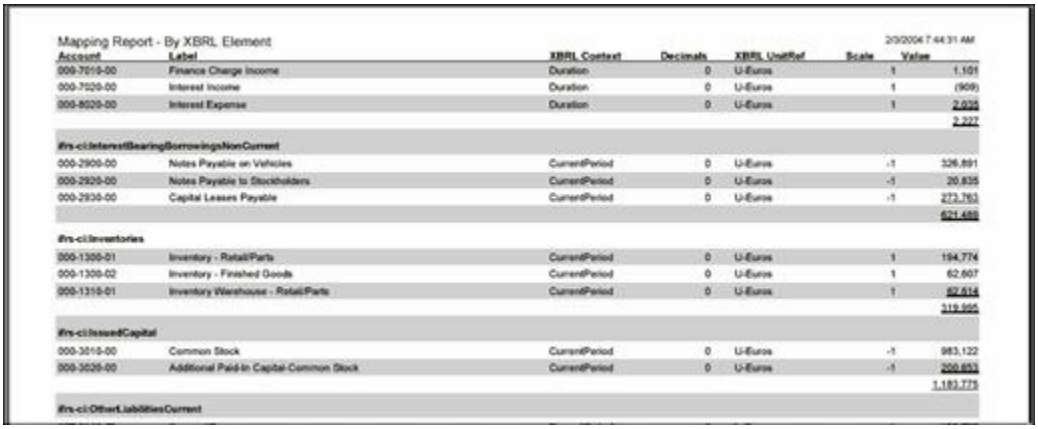
Printing the Audit Report

➤ To print a mapping audit report, select **File**, then **Print Preview**.

A mapping audit report looks like this:



This screen shot shows the mapping audit report preview:



Mapping Report - By XBRL Element

Account	Label	XBRL Context	Decimals	XBRL UnitRef	Scale	Value
000-7010-00	Finance Charge Income	Duration	0	U-Euros	1	1,101
000-7020-00	Interest Income	Duration	0	U-Euros	1	(909)
000-8020-00	Interest Expense	Duration	0	U-Euros	1	2,035
						2,227
#rs-ci:InterestBearingBorrowingsNonCurrent						
000-2900-00	Notes Payable on Vehicles	CurrentPeriod	0	U-Euros	-1	326,891
000-2920-00	Notes Payable to Stockholders	CurrentPeriod	0	U-Euros	-1	20,835
000-2930-00	Capital Leases Payable	CurrentPeriod	0	U-Euros	-1	273,763
						621,489
#rs-ci:Inventories						
000-1300-01	Inventory - Retail/Parts	CurrentPeriod	0	U-Euros	1	194,774
000-1300-02	Inventory - Finished Goods	CurrentPeriod	0	U-Euros	1	62,607
000-1310-01	Inventory Warehouse - Retail/Parts	CurrentPeriod	0	U-Euros	1	62,614
						319,995
#rs-ci:IssuedCapital						
000-3010-00	Common Stock	CurrentPeriod	0	U-Euros	-1	983,122
000-3020-00	Additional Paid-In Capital-Common Stock	CurrentPeriod	0	U-Euros	-1	200,653
						1,183,775
#rs-ci:OtherLiabilitiesCurrent						

This report shows the XBRL element (in bold on the left side), the source data table items (underneath the bold XBRL elements), the XBRL context assigned, the Decimals value assigned, the XBRL Unit Ref assigned, the scale applied to the source data, and the value actually placed in the instance document which is the sum of all values mapped to an instance document element name.

For row-mapped source data there is one report line when the source row has one context or multiple report lines when multiple contexts are processed (such as multiple year columns). For cell-mapped source data, there is one report line per mapped cell.

➤ To print the mapping audit report, after importing data into an instance, select **Output Print Preview**, **View to Print Mappings**, and select the **Show** checkbox for **Audit trail**.

Utilizing Mapping to Create Instance Documents

An instance document is created by importing and mapping source data to taxonomy concepts, XBRL context, and—for numeric entries—to unit, decimals/precision and scaled numeric values.

Import sources can be Excel, CSV, Word tables, Access tables and queries, OLE Provider database tables and queries, and instance documents produced under source taxonomies (to map to a taxonomy).

Imported data fields, such as columns of tables, are identified by their column headers or column position. A flexible process expresses these column relationships to instance data fields (including how to identify context and unit of each import field/column).

Import data field columns which match expected instance fields may in some cases be automatically recognized and automatically identified. Import field column headers which are not automatically recognized can be manually related to an instance import field in the import design process and that specification saved for later reuse manually and by scripted API programs.

Instance fact values might be (a) grouped by period into columns identified by import mapping, (b) specifically designated by context in the XBRL mapping, or (c) associated with a context in the import (such as by having pairs or triples of context and unit columns for each value column).

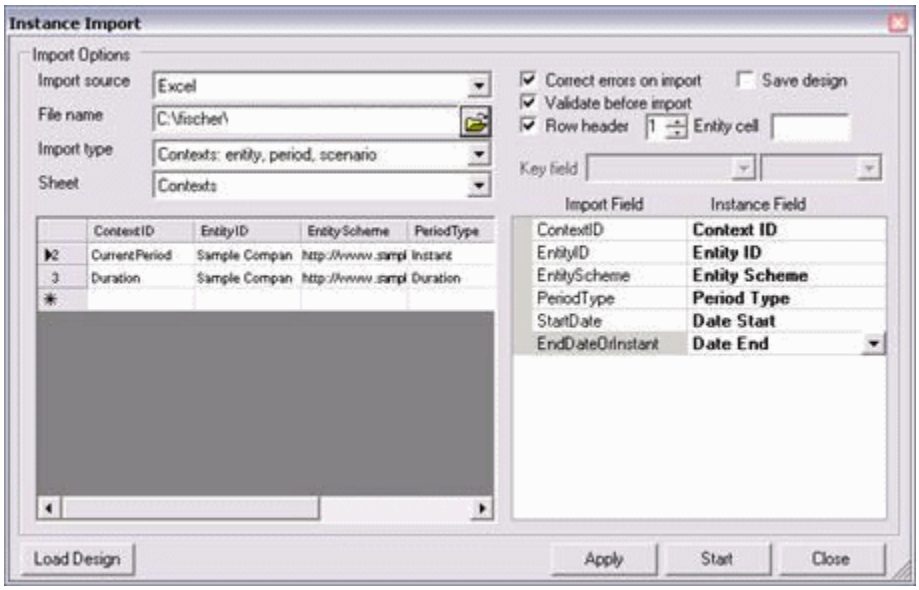
Instance fact values might also be individually designated cell by cell for mapping, optionally grouped by row sections (such as on equity statements). In this case, the XBRL mapping provides a template for the top (first) repeating group and each subsequent repeating group is a subsequent (or preceding) context period.

XML files issue from design of each of these (taxonomy and import column header) mappings.

Batch processing of the import process by API is achievable after the taxonomy mapping (row or cell mapping) and import field/column identification (column header mapping) are specified.

Instance Document Context Import Mapping

This screen shot shows an Instance Import field mapping screen for context import:



The Import Type Contexts selects a worksheet or database table called Contexts by default. A user can specify a worksheet or table of any other name.

For Excel or CSV, the first row is assumed to contain header names, but the row header can be advanced (for example, past rows of merged cells containing sheet names and dates). If no header row exists, select zero or uncheck row header. (For database the schema provides column names; for XML input this is determined from the XML tag names.)

The import field to instance field property grid attempts to automatically select best matches between columns as they are named in the import data and the corresponding instance field.

If user entries were made to associate import to instance fields, these selections can be saved (select the Save Design checkbox; it is saved at the next Apply or Start action).

If user entries are to be loaded from a previously saved association of import to instance fields, click the load design button and select a file.

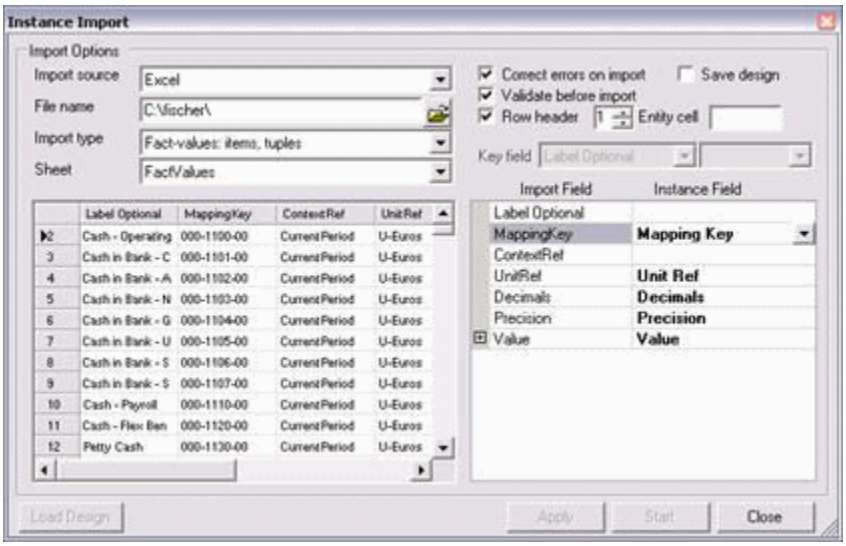
Table 81 Instance Fields

Field	Explanation
Context ID	An XML identifier for this context (for example, CurrentPeriod).
Context Label	A context label.
Context Description	Description of the context.
Entity ID	An identifier for the business entity (for example, a NASDAQ ticker moniker).
Entity Scheme	The namespace within which the business entity identifier is known (for example, http://www.nasdaq.com).
Entity Segment	A business segment within the business (for example, discontinuing operations).
Period Type	Instant, duration, or forever.
Date Start	A start date or instant date.
Date End	An end date, or instant date if not provided in the start date column.
Scenario	A business scenario.

In general, a minimal number of fields are needed to establish a workable context. It is feasible to have only a context ID, period type, and dates, but for valid XBRL instances an entity ID and scheme is also recommended. The context label and description are XBRL Taxonomy Designer extensions and completely optional. The Entity Segment and Scenario are XBRL fields and completely optional.

Instance Document Fact Value Import Mapping

This screen shot shows an Instance Import field mapping screen for Fact Value import (for tabular input source data).



See "Instance Document Context Import Mapping" on page 180 for a column header and design saving and loading discussion.

Instance fact value import can be tabular or XML tree object mapped. For tabular input sources input can be:

- Direct, by using an element label or name in each input row or record to identify the taxonomy element of the source fact
- XBRL Mapped, where the XBRL mapping (described previously) specifies a taxonomy element for each row or cell imported value, by using one of these:
  - A row mapping by key (such as GL Account number)
  - A row mapping by row number
  - A cell mapping by key and column header
  - A cell mapping by cell reference

For XML tree input sources, mapping can be XML tree object mapped by xPath expressions.

When tabular import is direct by element label or name, it is necessary also provide numeric values imported with unit identification and precision, and to provide all (numeric and non-numeric) values with context information.

Direct import of tabular values allows each value column to provide a period (date selection) or context (identifier selection), or refer to a companion column which contains that period or context identifier. When a period is specified by date selection, the element type determines whether an instant or duration period is needed, which is determined by context lookup. When a companion column is provided with a context identifier, then that specifies the value field's context. Direct import requires one input row or record per fact value. Multiple fact values for the same concept and context are entered using XBRL mapped import.

XBRL mapped import allows sophisticated mapping for accounting data, merging of multiple mapped input source values into one fact value with full audit trail, and dynamic determination of unit, context and scale.

Units can be specified on the XBRL mapped import, or if absent there they can be specified in the import field to instance field property grid.

Context likewise can be specified on the XBRL mapped import, or if absent it can be specified in the import field to instance field property grid.

Context for repeating rows of cell mapped data are advanced, group by group, by the repeat period property of the repeating row groups specification.

Table 82 Fact Value Fields

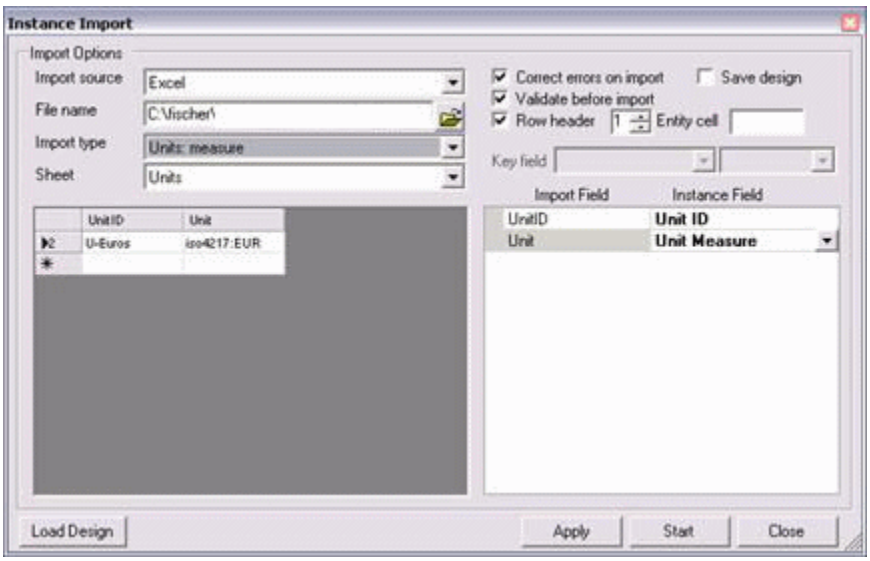
Field	Explanation
Fact ID	An optional identifier for this fact (for example, 29).
Tuple Name	The element name of the tuple containing the current fact. Must be input in tree nesting sequence.

Field	Explanation
Element Name, Element Label, or Mapping Key	Specifies how the import is matched to a taxonomy element. Element name or label implies by matching of the corresponding input field to a taxonomy element's name or label. Mapping key means that the XBRL Mapping specification loaded in the current instance is to be used to select the element. For label, this field can be expanded (plus box) for specification of label language and role. Each of these can be entered in the expanded selection, or specified in a companion column.
Unit Ref	Specifies that this input field contains a Unit ID for numeric input fields (which do not specify unit within XBRL Mapping specifications).
Decimals	Specifies that this input field contains a Decimals precision number for numeric input fields (which do not specify decimals within XBRL Mapping specifications).
Precision	Alternate to use of Decimals, specifies precision.
Value	Specifies that this input field contains a value. When selected this field can be expanded to specify period or context selection. A period or context can be specified in the expanded selection, or by a companion column which contains a period or context ID. For direct and XBRL Row mapping, every column which can contain a Value must have this selection for the import field. For XBRL cell mapping, import fields are identified only by the XBRL mapping specification and no Value import field to instance field association is necessary.

Generally only the Element Name/Label, Unit, Decimals/Precision, and Value fields are necessary. The specification of a tuple name is for situations where tuple-contained facts are provided. The specification of Mapping Key is for when an XBRL mapping specification provides a row or cell mapped specification for the value.

Using Instance Document Unit Import Mapping

This is the Instance Import field mapping screen for Unit import:



The Import Type Units selects a worksheet or database table called Units by default. However, you can specify a different worksheet or table name.

Table 83 Unit Fields

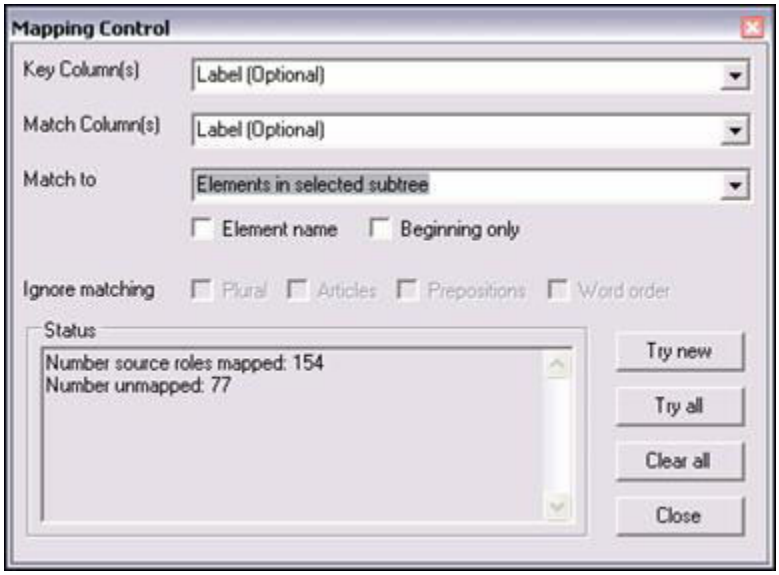
Field	Explanation
Unit ID	An XML unit identifier (for example, Monetary)
Unit Label	Unit label
Unit Description	Unit description
Unit Measure	One measure (for example, iso4217:EUR, iso4217:ISD., shares, or pure), or an XBRL expression of multiplication or division of measures (as a complete XML fragment)

Generally only the Unit ID and Unit Measure fields are required. The Unit Label and Unit Description are optional D extensions.

Mapping Control Window

The Mapping Control suggests mappings for row-by-key mappings.

This screen shot shows the mapping control for automatic label/name-based mapping suggestions:



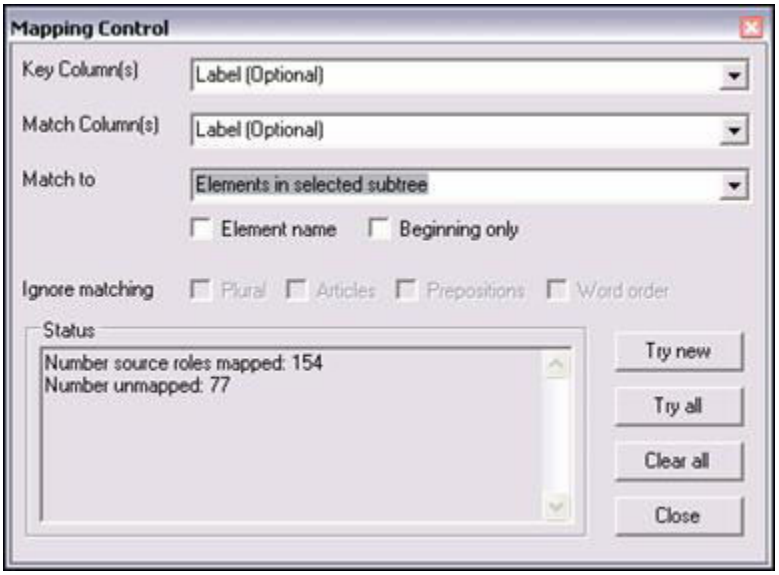
- **Key columns** indicates the source data grid columns which are the key columns. These columns form the filter pattern provided by the automatic mapping suggestion.
- **Match columns** indicates the source data grid columns which are matched against taxonomy labels or names, to find suggested alternative taxonomy concepts for each mapping suggestion.

  Key and match columns can contain multiple column names combined. Press the Ctrl key when selecting (or removing) multiple columns from the columns selection.
- **Match to** indicates what is to be matched in the taxonomy. The matching is based on the actively-selected taxonomy tree view (such as the presentation tree). For large and complete taxonomies, it may be suitable to narrow the part of the taxonomy being matched, such as

by highlighting a balance sheet subtree, and selecting to match to elements in the selected taxonomy view subtree.

- **Element name** is checked when name, instead of label, is to be matched. Beginning only is checked when the match columns are to match only the beginning of the label or word, instead of also looking for later words in a label or name.
- **Try new** uses the specified entries to suggest mappings of yet-unmapped source data rows. Try all clears mappings and try all rows. Clear all clears all current mappings.

The Mapping Properties Control suggests entry mapping properties for use as defaults when drag and drop or automatically mapping and for replacement of mapping properties (if entered or still blank).



- **Context** entries specify contextRef names to provide for duration and instant concept elements, for tabular input source. For XML tree input source an xPath expression resulting in contextRef names may be specified instead.
- **Unit** entries specify unitRef's to provide as default mapping properties for monetary, pure, and stock entries. For XML tree input source, an xPath expression resulting in unitRef names may be specified instead.
- **Decimal** entries specify default properties for monetary and other numeric entries. For example, monetary may be 2 (for dollars and cents), or -3 (for thousands rounding), but pure is usually 1.
- **Scale** is the multiplier on accumulating debit (usually scale 1) and credit (often -1) entries. Scale may be less than one to apportion to multiple concepts (such as .40 and .60). A scale of 0 or blank means the input value is multiplied as zero and thus entered as zero to the resulting instance document.

7 Dimensions

In This Chapter

This section explains how to work with XBRL dimensions.

Guide to Dimensions and Dimension Aggregation

Welcome to the XBRL Taxonomy Designer tutorial about Dimensions and Aggregation.

XBRL Taxonomy Designer and the XBRL Taxonomy Designer Dimensions tutorials currently support these versions of Dimensions and Aggregation:

- Dimensions 1.0, candidate release 3
- Aggregation 1.0 is currently in working draft format by the XBRL Consortium. XBRL Taxonomy Designer has implemented the schema `xbrldta-2005.xsd` for Aggregation. However, this has not yet been approved by the XBRL Consortium as a release.

Dimensions 1.0 is supplemental specification that enables you to apply dimensional concepts to XBRL documents. Aggregation 1.0 is an additional supplemental specification that enables you to sum values across dimensional concepts.

This tutorial is divided into incremental lessons to enable you to get a full understanding of dimensions and their use in XBRL:

- "Dimensions Overview" on page 188 explains the basic concepts of XBRL Dimensions.
- "Dimension Breakdown" on page 189examines the use of one dimension and provides an example.

- "Hypercube Breakdown" on page 190walks you through putting together a basic hypercube, identifies all of the parts of a hypercube, and explains how a dimension fits into a hypercube.
- "Dimensions and XBRL Overview" on page 191 discusses how XBRL and Dimensions work together. This lesson ties together all of the concepts from the previous three lessons.
- "Intermediate Hypercube" on page 192 adds complexity to the previous hypercube model and walks you through creating more difficult hypercubes.
- "Advanced Dimensions" on page 193covers the difference between implicit and explicit dimensions.
- "Aggregation Overview" on page 194 discusses the difference between Aggregation and Dimensions and how Aggregation can be applied to a hypercube.
- "Advanced Hypercube Example" on page 195 walks you through creating an advanced complex hypercube model using Aggregation.

Dimensions Overview

When business or financial reports are created, their purpose is to communicate a measurement of something—for example, Sales, Salaries, or Risk Values.

The item measured is often divided into categories, for example Sales by Geographical Region. XBRL Dimensions uses these categories as a way of describing how you got to that measured value.

In this example, we use the term "category" to express different "dimensions" to categorize a movie inventory. If you were creating a movie inventory, you might come up with a movie list that looks like this.

Goals for Movie Inventory:

1. Should be easy for customers to find the movies they want.
2. Should also be easy for employees to shelve the movies when customers return them.

One way to organize the inventory is alphabetical order. Customers can find what they are looking for if they know the name of the movie. However, this method is inefficient if customers don't know the name of the movie or even what movie they want to buy.

Data Modeling for Dimensions

Another way to organize the inventory is to group the movies according to category.

Two categories could be genre and rating.

Movie Inventory			
	Rating		
Movie Genre	G	PG	R
Science Fiction	The Black Hole	Star Wars	The Terminator

Movie Inventory			
Western	True Grit	Young Guns	On a Pale Horse
Classic	Bambi	Casablanca	Psycho

The purpose of the above chart is to catalog the movie inventory. By breaking down the inventory by movie genre and by rating, customers are able to find the movies they want easily and find similar titles that might interest them. A real inventory of a video store would contain thousands of titles, more genres, and more ratings, but this examine is perfect for our purposes. For example, you might have a hundred movies under the science fiction genre and G rating category. However, for the purpose of this example, the inventory is suitable.

A human can review the movie inventory report above and determine how the information is being broken down and categorized. A computer on the other hand that processes this movie inventory data requires a system for expressing and understanding this breakdown of data.

In Microsoft Excel, users can define a set of columns to report and categorize information:

Movie Genre	Rating	Movie Inventory
Science fiction	G	The Black Hole

To report only the Movie Genre “Science fiction”, Rated “G”, Movie inventory titles, users can pivot the columns to express a dimensional view of this information, breaking the information down into "Science Fiction" movies rated "G."

Title	Genre	Rating	Number of Copies
The Black Hole	Science Fiction	G	1
The Day the Earth Stood Still	Science Fiction	G	1
Star Wars	Science Fiction	PG	2
The Matrix	Science Fiction	R	2
The Terminator	Science Fiction	R	2

Dimension Breakdown

A dimension is a way of adding context to a measure value, and can be thought of as a category. In the example below, rather than view all movie releases, we add the concept of "Movie Genre" where the movies can be viewed by grouping them under the same genre.

Movie Inventory			
	Rating		
Movie Genre	G	PG	R

Movie Inventory			
Science Fiction	The Black Hole	Star Wars	The Terminator
Western	True Grit	Young Guns	On a Pale Horse
Classic	Bambi	Casablanca	Psycho

If we looked at just the Movie Genre dimension, it would look like this. The information remains the same, but the method in which it is displayed changes. The domain-members for the Movie Genre are "Science Fiction", "Western," and "Classic." It is very easy to determine which movies belong to the "Science Fiction" Movie Genre.

Movie Inventory	
Movie Genre	
Science Fiction	The Black Hole, Star Wars, The Terminator
Western	True Grit, Young Gun, On a Pale Horse
Classic	Bambi, Casablanca, Psycho

Hypercube Breakdown

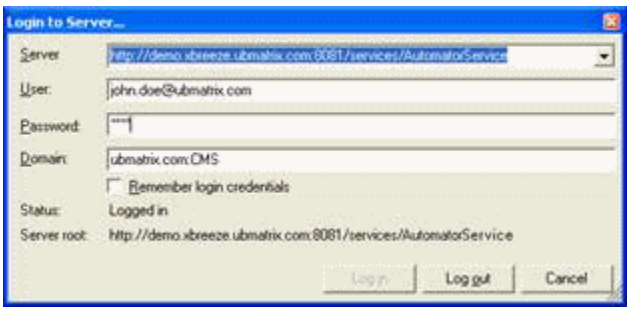
The example below is a fairly generic chart analyzing the breakdown of the Movie Inventory, by genre, by rating. Each of these breakdowns can be considered a way to view the same information. We could think of these individual breakdowns as views, slices, or dimensions. Collectively, these dimensions form a hypercube.

The human brain can examine this data and determine what the report is conveying where as a computer would not be able to determine without some knowledge and language to express the dimensions, the data, and the connection between them. People can make assumptions and connections based on past experiences and education. For example, review the chart below and know that Psycho should not be rented to a child because it is rated R. A computer knows the rating of a movie only as a value unless we provide it with context.

Movie Inventory			
	Rating		
Movie Genre	G	PG	R
Science Fiction	The Black Hole	Star Wars	The Terminator
Western	True Grit	Young Guns	On a Pale Horse
Classic	Bambi	Casablanca	Psycho

Hypercube

A hypercube is a construct designed to express a collection of dimensions. The example above could be expressed as a simple hypercube. The axis of the cube below is a graphical representation to aid your understanding of dimensions. Each dimension has its own axis. The human brain can view a spreadsheet and intuitively understand these details, but a computer needs a way of expressing dimensions.



Primary Item

The first step to modeling data using dimensions is to determine what it is we are measuring. In the above example, we are examining different views of the Movie Inventory with the filters of Genre and Rating. We are measuring the Movie Inventory; therefore we state that the Movie Inventory is the primary item.

Dimension

In the Movie Inventory example we decided to divide Sales by Products and Regions. These categories are considered to be dimensions, or categories by which we are going to analyze the information.

Domain and Domain Members

A domain is all of the domain members which are used to express a dimension. In the example above, Genre is a dimension. The domain members for Genre are Science Fiction, Western, and Classic. Collectively, these domain members form the Genre Domain.

Dimensions and XBRL Overview

The XBRL Dimensions 1.0 specification provides an optional incremental syntax to the XBRL 2.1 specification to express Dimensions in a global standardized manner.

XBRL 2.1 calculations are limited in that calculations that are expressed can only be can only be executed under the same content, and not across different contexts. Dimension aggregation calculations enable you to complete calculations across different contents. If you have a requirement to complete calculations across contexts, then it is necessary to create a business

rule (formula) to complete a calculation across different contexts, or make use of the dimensional specification. The authors of the XBRL Dimensions specification leveraged the XBRL syntax or architecture which allows dimensions to be expressed in a standardized manner; and the dimensions to be extended or restricted as necessary.

The XBRL Dimensions specification provides the context needed for financial reporting software to effectively analyze this information. To read more about this specification which is currently published as a public working draft go to: http://www.xbrl.org/SpecPWDs/

From an XBRL perspective, dimensions are the ability to express relationships between XBRL contexts. XBRL calculations are limited in that calculations can only be expressed and therefore performed within the same context. Using XBRL dimensions, relationships can be expressed across contexts.

Dimensions are expressed using an XBRL taxonomy which also supports features of the XBRL dimensions specification. Within the XBRL taxonomy, information is expressed which provides continuity between:

- Primary items
- Hypercubes
- Dimensions
- Domains
- Domain members

An XBRL processor that supports dimensions can then use this additional information to perform additional processing expressed by the dimensions.

Intermediate Hypercube

Most Dimensional Taxonomies contain one or more hypercubes with one or more dimensions and therefore are considered more complex than our current example. In this section we add additional dimensions into the movie inventory and therefore additional complexity.

Goals for Movie Inventory include:

- Cost per Movie Title
- Number of Copies of a Movie Title
- Barcode of Movie Title

A basic hypercube created from these goals might look like the example below.

Movie Inventory			
Movie Title	Cost per Movie	Number of Copies	Barcode
The Black Hole	20.00	2	2144124
Star Wars	15.00	3	4436346

Movie Inventory			
The Terminator	13.99	1	3463463

Adding in the information from the Hypercube Breakdown creates a Hypercube that looks like the example below.

Movie Inventory					
Movie Title	Genre	Rating	Cost per Movie	Number of Copies	Barcode
The Black Hole	Science Fiction	G	20.00	2	2144124
Star Wars	Science Fiction	PG	15.00	3	4436346
The Terminator	Science Fiction	R	13.99	1	3463463

This hypercube enables you to examine the movie inventory by a variety of dimensions.

Advanced Dimensions

Dimensions can be divided into explicit and implicit dimensions.

Note: These features are demonstrated in"Lesson Two: Creating a Dimension Taxonomy" on page 331.

- **Explicit Dimensions**—Explicit dimensions are dimensions whose domain-members are explicitly defined. In the Hypercube Breakdown example, the Movie Genre is an example of an Explicit Dimension because all of the domain-members are strictly defined.

Movie Inventory	
Movie Genre	
Science Fiction	The Black Hole, Star Wars, The Terminator
Western	True Grit, Young Gun, On a Pale Horse
Classic	Bambi, Casablanca, Psycho

- **Implicit Dimensions**—There are occasions where the number of domain members is known, but so large that it would be impractical to list all domain members. For example,
  - Each ZIP code in the US.
  - Longitudes, which span from 180 to -180.

    The XBRL Dimensions specification provides a mechanism to express these types of dimensions. The XBRL Dimensions specification calls implicit dimensions "typed" dimensions.

In the Intermediate Hypercube example, Cost is a good example of an Implicit Dimension. It would be impossible to define all of the possible costs as it changes based on market forces. Instead, we create an Implicit Dimension with a domain-member of all-possible-costs.

Movie Inventory					
Movie Title	Genre	Rating	Cost Per Movie	Number of Copies	Barcode
The Black Hope	Science Fiction	G	20.00	2	2144124
Star Wars	Science Fiction	PG	15.00	3	4436346
The Terminator	Science Fiction	R	13.99	1	3463463

Aggregation Overview

XBRL Dimensions 1.0 was designed to express the relationships between dimensions. The XBRL Aggregation 1.0 specification enables users to create summable values across dimensions.

Note: These features are demonstrated in "Lesson Four: Creating a Hypercube Taxonomy" on page 333 of the Basic Dimensions Tutorial.

With XBRL, it would be easy to determine the sum of all of the genres for the year 2003. However, if you wanted to determine the sum of Science Fiction across the periods of 2003, 2002, and 2001 you would use Dimensions and Aggregation.

To determine the sum value of science fiction requires the aggregation of the values for Science Fiction-2003, Science Fiction-2002, and Science Fiction 1000. XBRL alone does not allow users to sum values across dimensions.

Movie sales Analysis					
		2003	**2002**	**2001**	
Sales by Genre		32,038	35,805	32,465	
Breakdown by Genre					
Science Fiction		20,181	18,150	15,275	53,606
Wester		2,433	1,973	1,823	
Classic		6,675	6,514	5,752	
Other		2,749	9,168	9,615	

Movie sales Analysis					
	Grand Total	32,038	35,805	32,465	

Advanced Hypercube Example

The Movie Inventory is a decent start. The next step is to create a hypercube that measures Movie Sales by Genre and by Rating.

Note: These features are demonstrated in "Lesson Four: Creating a Hypercube Taxonomy" on page 333 of the Basic Dimensions Tutorial.

Goals for Movie Inventory include:

- Determine which type of movies sell better.
- Determine which movies sell poorly.

A basic hypercube created from these goals might look like the example below. This example uses the Genre and Rating breakdown from previous examples. In addition, it uses a third dimension of period to further see how sales change from year to year.

Movie Sales Analysis				
		2003	2002	2001
Sales, by Genre, by Ratings		32,038	35,805	32,465
Breakdown by Genre				
Science Fictiont		20,181	18,150	15,275
Western		2,433	1,973	1,823
Classic		6,675	6,514	5,752
Other		2,749	9.168	9,615
	Grand Total	32,038	35,805	32,465
Breakdown by Rating				
G		10,214	12,649	10,137
PG		11,901	10,374	10,396
R		5,639	4,371	3,210
Other		4,284	8,411	8,722

Movie Sales Analysis				
	Grand Total	32,038	35,805	32,465

From this example, we can determine that Science Fiction movies outsell all of the other movie genres. In addition, movies rated G or PG outsell other ratings. Inversely, Westerns seem to sell poorly so users might decide to buy less of them in the future.

Notice that the Grand Totals for the Genre and the Rating breakdowns are the same year by year. This is because the information being measured is the same in both dimensions, but the way that it is being quantified differs. XBRL Dimensions 1.0 is what enables you to manipulate this information into suitable formats while XBRL Aggregation1.0 is what enables you to create summable values across dimensions.

Dimensions Validation

Validating Dimensions and Aggregation Files

The process for validating Dimensions and Aggregation is the same process for validating taxonomies and for validating instance documents; however, dimensions and aggregation validation requires another set of validation error messages than the standard XBRL file. Users can find these errors here:

- "XBRL Taxonomy Designer Dimension Validation Errors" on page 196
- "XBRL Taxonomy Designer Instance Dimension Validation Errors" on page 198

See Also

"Reading a Calculation Trace" on page 151

XBRL Taxonomy Designer Dimension Validation Errors

The namespace `xbrldte` is defined as `http://xbrl.org/2005/xbrldt/errors`.

Table 84 XBRL Taxonomy Designer Dimension Validation Errors

Taxonomy Error	Meaning	Reference
[Dim Err, 1] xbrldte:HypercubeElementIsNotAbstractError	Hypercube element must be abstract	2.2
[Dim Err, 2] xbrldte:HypercubeDimensionSourceError	The source of the hypercube-dimension arc is not the correct type.	2.2.2.1
[Dim Err, 3] xbrldte:HypercubeDimensionTargetError	The target of the hypercube-dimension arc is not the correct type.	2.2.2.1
[Dim Err, 4] xbrldte:HasHypercubeSourceError	The source of an all, notAll arc is not the correct type.	2.3.1.1

Taxonomy Error	Meaning	Reference
[Dim Err, 5] xbrldte:HasHypercubeTargetError	The target of an all, notAll arc is not the correct type.	2.3.1.1
[Dim Err, 6] xbrldte:HasHypercubeMissingContextElementAttributeError	The all, notAll arc does not contain an xbrldt:contextElement attribute.	2.3.1.1
[Dim Err, 7] xbrldte:TargetRoleNotResolvedError	The URI content of a xbrldt:targetRole attribute cannot be resolved using a roleRef to a roleType.	2.4.3
[Dim Err, 8] xbrldte:DRSUndirectedCycleError	Within a Dimensional Relationship Set there must not be undirected cycles not allowed by the arc declaration.	2.4.3
[Dim Err, 9] xbrldte:DRSDirectedCycleError	Within a Dimensional Relationship Set there must not be directed cycles not allowed by the arc declaration.	2.4.3
[Dim Err, 10] xbrldte:DimensionElementIsNotAbstractError	Dimension elements must be abstract	2.5
[Dim Err, 11] xbrldte:TypedDomainRefError	The xbrldt:typedDomainRef attribute is displayed on an element declaration that is not a dimension declaration.	2.5.2.1.1
[Dim Err, 12] xbrldte:TypedDimensionError	The xbrldt:typedDomainRef attribute does not locate a typed dimension.	2.5.2.1.1
[Dim Err, 13] xbrldte:TypedDimensionURIError	The xbrldt:typedDomainRef attribute contains an invalid URI or does not contain a fragment identifier	2.5.2.1.1
[Dim Err, 14] xbrldte:DimensionDomainSourceError	The source of a dimension-domain arc is not the correct type.	2.5.3.1.1
[Dim Err, 15] xbrldte:DimensionDomainTargetError	The target of a dimension-domain arc is not the correct type.	2.5.3.1.1
[Dim Err, 16] xbrldte:PrimaryItemPolymorphismError	A cycle exist, the primary item cannot be a member of the domain of any of its dimensions	2.5.3.1.1 2.5.3.2.1
[Dim Err, 17] xbrldte:DomainMemberSourceError	The source of a domain-member arc is not the correct type.	2.5.3.2.1
[Dim Err, 18] xbrldte:DomainMemberTargetError	The target of a domain-member arc is not the correct type.	2.5.3.2.1
[Dim Err, 19] xbrldte:NoStandardLabelError	The dimensional element does not contain a standard label in any language.	2.7.1.1
[Dim Err, 20] xbrldte:DimensionDefaultSourceError	The source of a dimension-default arc is not an explicit dimension declaration	2.8.1.1
[Dim Err, 21] xbrldte:DimensionDefaultTargetError	The target of a dimension-default arc is not a domain member declaration	2.8.1.1
[Dim Err, 22] xbrldte:TooManyDefaultMembersError	The dimension has two or more members in its domain that MAY play the role of the default member.	2.8.1.1

XBRL Taxonomy Designer Instance Dimension Validation Errors

The namespace `xbrldie` is defined as `http://xbrl.org/2005/xbrldi/errors`.

Table 85 XBRL Taxonomy Designer Instance Dimension Validation Errors

Instance Error	Meaning	Reference
[Ins Err, 1] xbrldie:DefaultValueUsedInInstanceError	Default values of dimension must not be reported in instances.	2.8.1.2
[Ins Err, 2] xbrldie:PrimaryItemDimensionallyInvalidError	The primary item contains invalid hypercubes in all base sets	3.1.1
[Ins Err, 3] xbrldie:RepeatedDimensionInInstanceError	It is not allowed to report a value for the same dimension more then once.	3.1.4.2
[Ins Err, 4] xbrldie:TypedMemberNotTypedDimensionError	The xbrldi:typedMember element does not refer to a typed dimension.	3.1.4.4.2
[Ins Err, 5] xbrldie:ExplicitMemberNotExplicitDimensionError	The xbrldi:ExplicitMember element does not refer to an explicit dimension.	3.1.4.5.2
[Ins Err, 6] xbrldie:ExplicitMemberUndefinedQNameError	The QName value of the xbrldi:explicitMember element is not an element defined in the taxonomy schema.	3.1.4.5.3

8 Formulas

In This Chapter

What are XBRL Formulas?

Formula is the XBRL term for business rules. Business rules are the practices, processes, and polices of an organization expressed semantically so that XBRL can manage, or analyze the organization's data according to these rules. These statements define or constrain some aspect of the business which is intended to assert business structure, or to control or influence the behavior of the business.

Types of business rules might be:

- Definitions such as "Assets = Liabilities + Equity"
- Calculations such as "Total Property, Plant and Equipment = Land + Buildings + Fixtures + IT Equipment + Other"
- Process oriented such as "If property, plant, and equipment exists; then a property, plant and equipment policy must exist and property, plant and equipment disclosures must exist."
- Instructions or documentation such as "Cash flow types must be operating, financing, or investing."

Business rules are important because they are extraordinarily useful for financial reporting; some of the benefits of XBRL business rules are:

- Rather than paying programmers to update rules (which is expensive and time consuming), business users who actually understand the rules can update them, saving time and money.
- Rather than having programmers create validation for each thing they wish to validate (called one-to-one programmatic validation) a business rules engine can be used to do validation (many-to-many rules-based validation).

If the business rules are also agreed upon, then they can become an accepted global standard format. Therefore, you can exchange the rules with others. For example, a business might use the business rules to explain the data it is collecting, determine which data needs to be collected, validate the data before it is submitted, and decide which data collection forms should be used by the type or quality of entity submitting data.

All this promotes an understanding of business policies and procedures, facilitates consistent decision making, forces order to rules and policies because they are clearly expressed; all with increased flexibility because of the separation of the processing logic from the rules, the ability of the business users to control the processing logic easily without understanding programming.

Disclosure Management Formula and Function Syntax

This section explains how to use formula and function syntax.

Create a Simple Formula

➤ To create a simple formula:

1 Create a taxonomy.

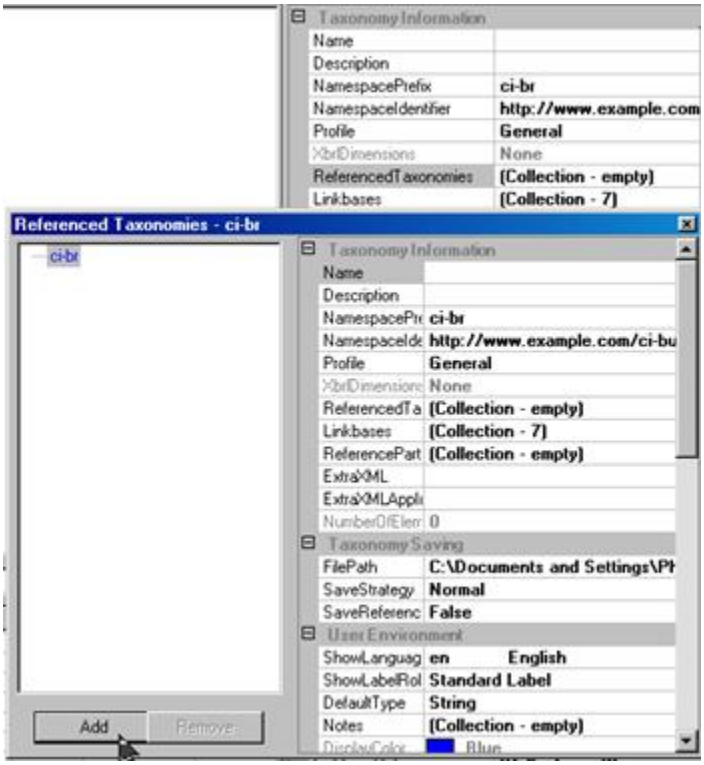
2 Select the **Taxonomy Property** window and change the **namespace prefix** to **ci-br**.

3 Change the **namespace identifier** to `http://www.example.com/ci-businessrules`.

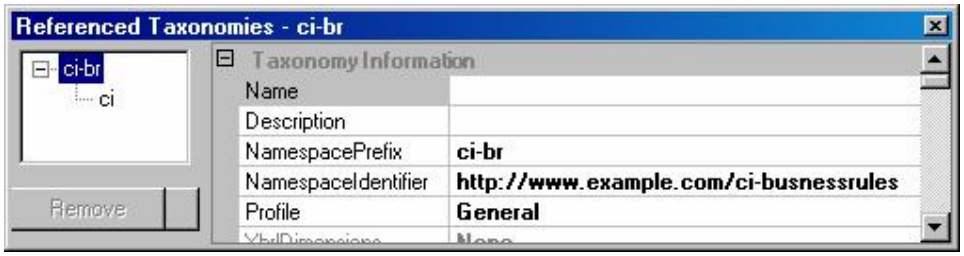
4 Select **File**, and then **Save As**.

5 Enter `CIBusinessRules.xsd`.

6 Create a reference taxonomy through the Reference Taxonomies property grid using the Basic Calculation taxonomy.



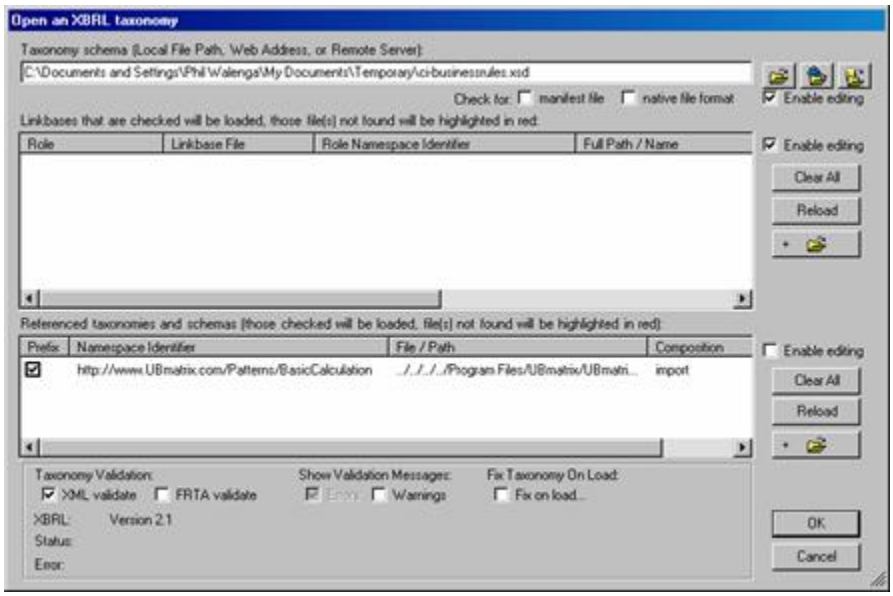
The Reference Taxonomies property grid now shows:



7 Select **Turn off**, then **View** and then **Namespace Prefix** so that when you edit concepts you are not editing the namespace prefix.

8 Select **File**, and then **Save** and close the taxonomy.

9 Open `ci-businessrules` taxonomy and notice that BasicCalculation taxonomy is now a referenced taxonomy to your ci-businessrules taxonomy.

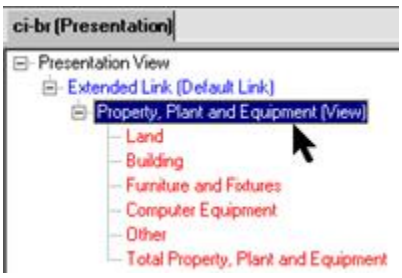


Note: If you want to use colors to distinguish between the reference taxonomy (BasicCalculation) and your taxonomy cibusinessrules, select Tools, and then Options and move colors of your choice. Save the taxonomy and reload it.

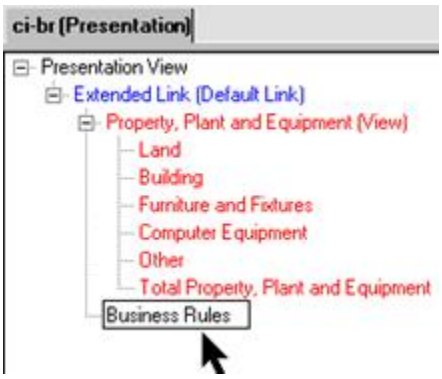
Now let's edit some elements and create elements where we can store formulas.

10 Right-click the element **Property, Plant and Equipment** to edit and add the word (View).

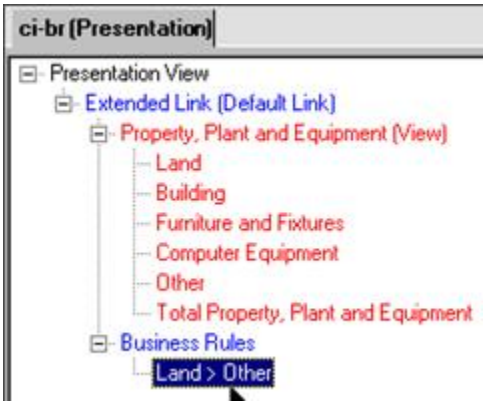
This separates the concepts from the "containers" that we create to store formulas.



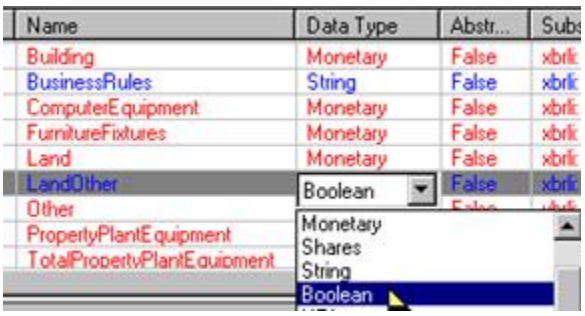
11 Now add the element "**BusinessRules** as a concept at the same level as Property, Plant and Equipment (View).



12 Add a concept under Business Rules and call it **Land > Other**.



13 Set the data type to **Boolean**.



14 Select “**Land > Other**” concept in the presentation view.

15 Select **Business Rules** and change **Abstract** to **True**.

16 Select **Formula** and then choose **[...]** from the element property grid.

17 Select **ci:Land** from the presentation relationship view, and then drag it to the formula editor.

18 Enter **>** .

19 Select **ci:Other** from the presentation relationship view, and then drag it to the formula editor.

20 Select  to validate the formula.

If the formula is valid, this message is displayed in the messages window.

Line	Type	Message ID	Message
1	Info	fmChkOk	Formula has no errors: ci:Land > Other Compiled as: ci:Land > ci:Other

If the formula is invalid, an error is displayed.

Line	Type	Message ID	Message
1	Info		Validating Formula syntax and semantics :ci:Land > Othe Compiled as: ci:Land > UnknownTerm:Othe
2	Error	fmErrFnName	The name, Othe, is not an implemented function, nor is it an unqualified element name of a loaded taxonomy

21 Select **[x]** to close the formula editor.

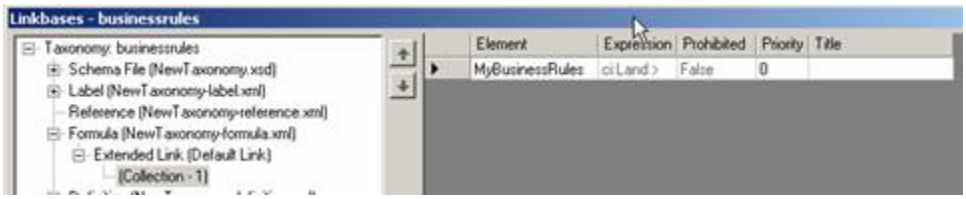
22 Select the element property window and note the formula expression.

23 Select the **Taxonomy Property** window, then **Linkbases**, and then **[...]**.

24 From the **FormulaType** list, select **formula linkbase; note.**.

The default is Disclosure Management (for the Disclosure Management propriety format), this can be changed to XBRL 1.0 (for the XBRL international formula 1.0 specification).

Note the formula linkbase that is added; note the expression that is detailed and the element that the formula is associated with.

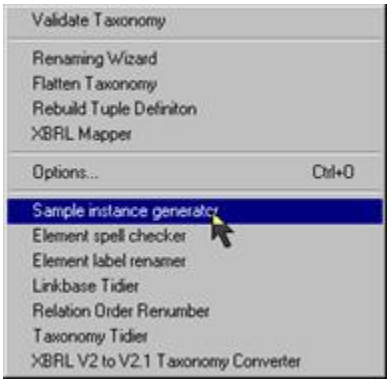


25 Select **Tools**, and then **XBRL Validation** to verify that the taxonomy is XBRL valid

26 Select **[x]**, then **File**, then **Save**, and then `CIBusinessRules.xsd`.

Create an instance document for the extended taxonomy and validate the business rules by using the Tools Taxonomy Generate Business Rules menu choice.

All elements

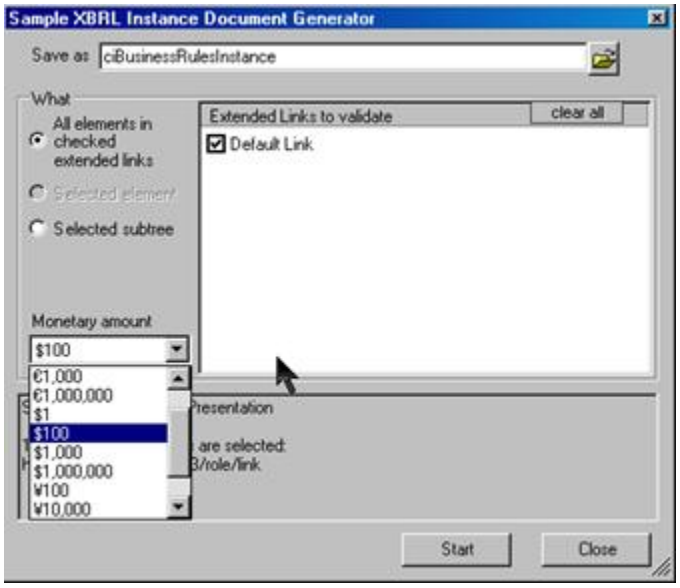


27 Enter in a name for the instance document `ciBusinessRulesInstance`.

For example, select **All Element** in checked extended links, and for **Monetary** amount, select **$100**.

A value of $100 is entered into the instance for each concept. Since there are five concepts, the total should be $500 and be reported in the instance document.

28 Select **Start** and then **Close.**



Validate the instance document and the business rule. Change some values so the validation triggers some errors.

29 **To do this, close the taxonomy and open the instance document that you created.**

30 **Run the business rule validation from the menu toolbar.**

31 **For land, enter a value to less than $100 and run the validation again.**

An XBRL element can have the attribute substitution group containing one of these default values: xbrl:item, or xbrl:tuple; or it may also substitute for another element of the same type.

Table 86 Background Information

Term	Definition
Concept	Concepts are defined in two equivalent ways. In a syntactic sense, a concept is an XML Schema element definition, defining the element to be in the item element substitution group or in the tuple element substitution group. At a semantic level, a concept is a definition of kind of fact that can be reported about the activities or nature of a business activity.
Item	An item is an element in the substitution group for the XBRL item element. It contains the value of the simple fact and a reference to the context (and unit for numeric items) needed to correctly interpret that fact. When items occur as children of a tuple, they must also be interpreted in light of the other items and tuples that are children of the same tuple. There are numeric items and non-numeric items, with numeric items being required to document their measurement accuracy and units of measurement.
Tuple	A tuple is an element in the substitution group for the XBRL tuple element. Tuples are used to bind together the parts of a compound fact. Those constituent parts are themselves, facts but they must be interpreted in light of each-other. For example, the name, age and compensation of a director of a company must be grouped together to be correctly understood.

Sample screenshots

Substitute for xbrl:Item because the element is a distinct item reported instance document.

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyItem	MyItem	Monetary	False	xbrli:item

Substitute for xbrl:Tuple because the tuple groups together other items.

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyTuple	MyTuple	Tuple		xbrli:tuple

Substitute for another element of the same type; if they are not the same type, an error occurs. This technique enables you to substitute for elements of the same type, and enables you to restrict the original concepts definition.

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyItem	MyItem	Monetary	False	xbrli:item
new	Another...	AnotherI...	Monetary	False	new:MyItem

Formula Collection Editor

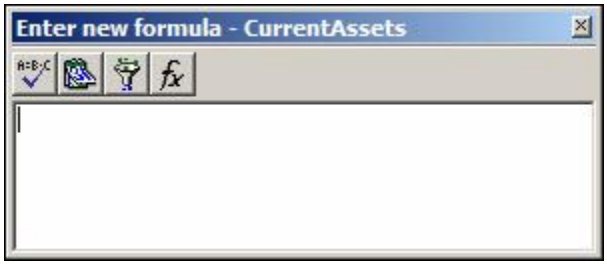


Table 87 Formula Entry Form Controls

Form Control	Description
	Checks the syntax of the formula, displaying any parser messages in the message collection, and displays the decompiled formula as it would be saved in a formula linkbase.
	Allows the entry of a named relative context.
	Allows the entry of a named absolute context
	Allows standard functions to be selected from a menu instead of remembering and typing them.
Formula	The formula itself is entered into the text box area. Element names can be dragged (from the relationships tree or elements list view) and dropped into the text box, or you can enter values into the text box. For example “ci:TotalAssets = ci:TotalLiabilitiesAndEquity“ indicates that total assets must equal total liabilities plus equity in instance documents.

The formula syntax is similar to Microsoft Excel syntax for formulas. Table 88 lists examples of formula expressions.

Table 88 Example Formula Expressions

Formula	Description
ci:ElementA = ci:ElementB	Boolean result true if ElementA equals ElementB

Formula	Description
ci:ElementA > ci:ElementA[-P1Y]	Boolean result true if ElementA is greater than the corresponding element of the previous year's context
ci:ElementA < 100000	Boolean result true if ElementA is less than 100000
ci:ElementA + ci:ElementB	Addition result of the same type as the two terms, in same unit and context as terms.

Create an XBRL instance

➤ To create an XBRL Instance

1 Select **File**, then **New**, and then **Instance**.

2 Select **File**, then **Load**, and then **Taxonomy**.

3 Enter `MyBusinessRules.xsd` and click **OK**.

4 Select the **Instance property** window.

5 Select **Entity**, and then **Add**.

6 In the Identifer field, enter "=**SAMP**.

7 For Schema, enter http://www.samplecompany.com/.

8 Select **OK**.

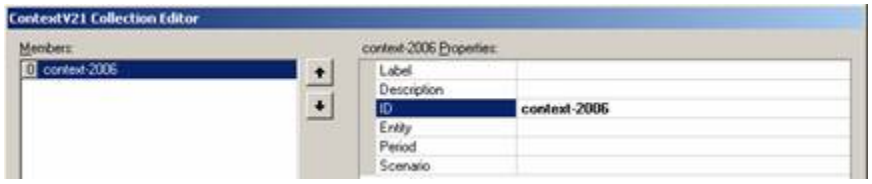
9 Select **Units**, and then **Add**.

10 Select **ID**, and then **units-monetary**.

11 Select **MeasureSimpleTerm**, then **iso4217:USD**, and then **OK**.

12 Select **Context**, and then **Add**.

13 For **ID**, enter **context-2006**.



14 Select **Entity**, and then select the **SAMP** entity.

15 Select **Period**, and then * to create a context.

16 Select **Type**, then **Instant**, enter a **date**, and then **OK**.

17 Select the **fact value grid**.

18 Select the **element field**, and then **ci:Land**.

19 Select **context** , then **context-2006** , then **D**, and then enter **INF**.

20 Select **unit**, then **units-monetary**, and then set the value to **1**.

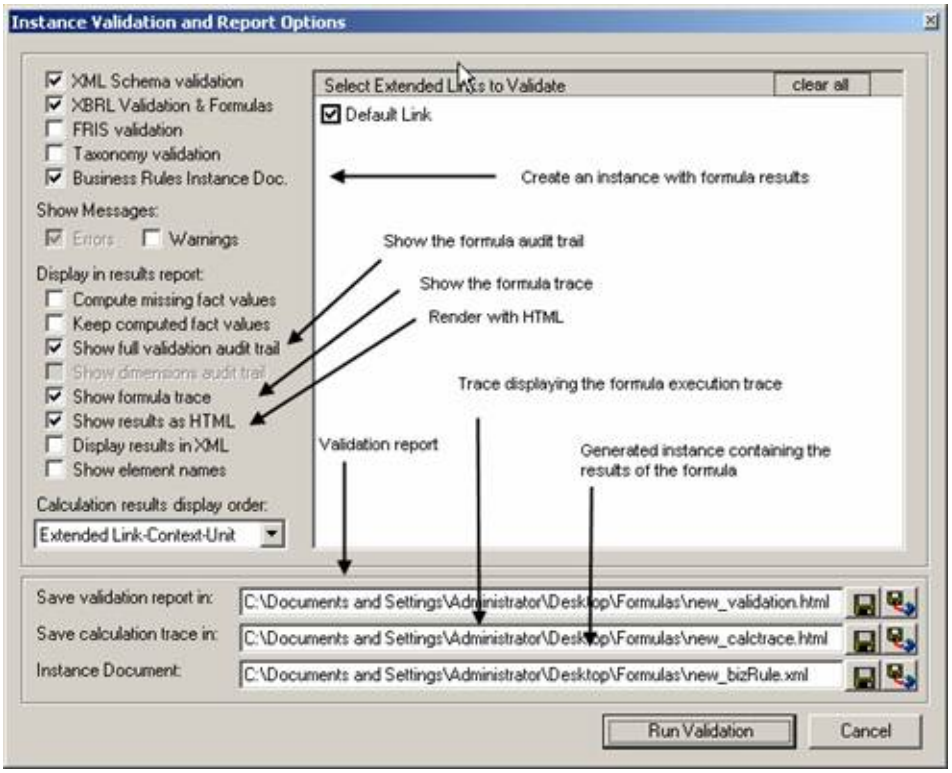
Sequence /	S	Element	Context	P	D	Unit	IsNi	Value	Effective Value
1	1	ci:Land	context-2006		0	unit-monetary		1	1
2	2	ci:Building	context-2006		0	unit-monetary		1	1
3	3	ci:ComputerEquip...	context-2006		0	unit-monetary		1	1
4	4	ci:FurnitureFixtures	context-2006		0	unit-monetary		1	1
▶ 5	5	ci:Other	context-2006		0	unit-monetary		1	1

21 Select  validate business rules, or alternatively **Tools**, and then **validate instance document**.

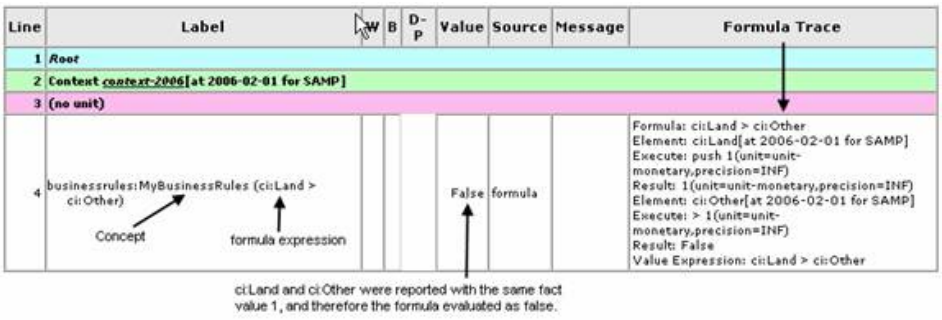
22 Select **full validation audit trail**.

23 Select **Show formula trace**.

24 Select **Run Validation**.



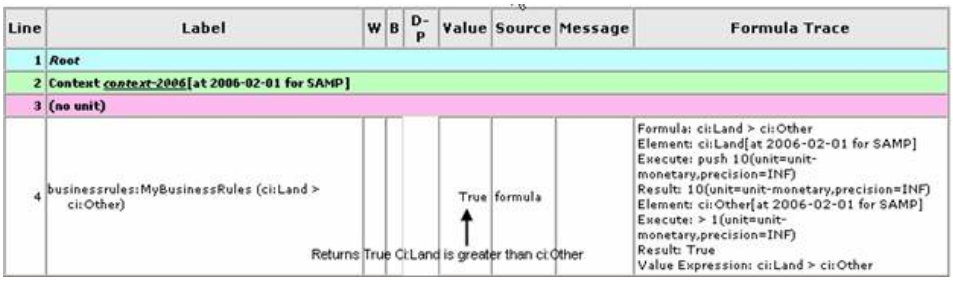
With HTML rendering is selected this report is generated and displayed.



Note that reported fact value for ci:Land is not greater than the reported fact value for ci:Other and therefore the Boolean result is false. Note also the formula was evaluated for each context that was defined. In this example only one context was defined. If there were additional contexts, the formula would have been evaluated multiple times.

25 **Select the reported fact value ci:Land, and then change the value to 10.**

26 **Validate the instance document again.**



Formula Linkbase

XBRL Taxonomy Designer creates a proprietary XBRL-based linkbase that is similar to the labels or references linkbases currently specified in by XBRL 2.1. An instance document or taxonomy refers to a formulas linkbase using the standard XBRL "linkbaseRef" element with a XLink:role attribute value. See the XBRL Specification section 4.3 which explains the linkbaseRef element.

Formulas are contained in the Disclosure Management proprietary formulaLink extended link. The pattern for the formula linkbase followed matches that defined in the XBRL specification. The formulaLink extended links may contain locators, arcs, and resources. Formula information is contained in a formula resource. Relative context (period offset specifications) and absolute context (period, entity, and other filtering criteria) are also contained in the formulaLink resources.

The formulaLink Element

The formulaLink element is an extended link. Its generic syntax is documented in the XBRL specification . It contains relationships between concepts and formulas and the business rules associated with the concepts. The XML Schema constraints for the formulaLink element are:

```
<schema targetNamespace="http://www.oracle.com/2004/XLink/xbrllinkbase/formula"
xmlns:xl="http://www.xbrl.org/2003/XLink" xmlns:flink="http://www.oracle.com/2004/XLink/
```

```
xbrllinkbase/formula" xmlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:XLink="http://www.w3.org/1999/XLink"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <import namespace="http://www.xbrl.org/2003/linkbase" schemaLocation="http://
www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd"/>
 <import namespace="http://www.xbrl.org/2003/XLink" schemaLocation="http://www.xbrl.org/
2003/xl-2003-12-31.xsd"/>
 <import namespace="http://www.w3.org/1999/XLink" schemaLocation="http://www.xbrl.org/
2003/XLink-2003-12-31.xsd"/>
<element name="formulaLink" substitutionGroup="xl:extended">
 <annotation>
 <documentation>
 formula extended link element definition
 </documentation>
 </annotation>
 <complexType>
 <complexContent>
 <extension base="flink:formulaExtendedType">
 <choice minOccurs="0" maxOccurs="unbounded">
 <element ref="flink:absoluteContext"/>
 <element ref="flink:relativeContext"/>
 </choice>
 </extension>
 </complexContent>
 </complexType>
</element>
<complexType name="formulaExtendedType">
 <complexContent>
 <restriction base="xl:extendedType">
 <choice minOccurs="0" maxOccurs="unbounded">
 <element ref="xl:title"/>
 <element ref="link:documentation"/>
 <element ref="flink:loc"/>
 <element ref="flink:formulaArc"/>
 <element ref="flink:formula"/>
 </choice>
 </restriction>
 </complexContent>
 </complexType>
```

The formula Element

The formula element is an Xlink resource that uses the standard xml:lang attribute to specify the language used, and an optional classification of the formula purpose using a role attribute.

The XML Schema constraints for the formula element are:

```
<element name="formula" substitutionGroup="xl:resource">
 <annotation>
 <documentation>
 Definition of the formula resource element.
 </documentation>
 </annotation>
 <complexType mixed="true">
 <complexContent mixed="true">
 <extension base="xl:resourceType">
```

```
<sequence>
<any namespace="http://www.w3.org/1999/xhtml"
processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
<attribute name="select" type="string" form="unqualified"/>
</extension>
</complexContent>
</complexType>
</element>
```

These are example formula resource elements:

```
<formula XLink:type=”resource” XLink:role=”http://www.oracle.com/2003/role/formula”
XLink:label=”ci_LandDepreciates_formula” select=”ci:Land < ci:Land[-P1Y]” />
```

The xml:lang attribute

All formula resources must contain an xml:lang attribute identifying the language used for the content of the formula. The value of the xml:lang attribute must conform to XML rules.

The formulaArc Element

The XML Schema constraints on formulaArc elements are shown below:

```
<element name="formulaArc" type="xl:arcType" substitutionGroup="xl:arc">
<annotation>
<documentation>
Concrete arc for use in formula extended links.
</documentation>
</annotation>
</element>
```

This arc role value is for use on a formulaArc from a concept locator (“loc” element) to a formula element. It indicates that the formula conveys human-readable information about the concept.

The formulaArc elements cannot describe cyclic relationships between concepts because they are only relate concepts to formula resources, not other concepts. For this reason, no restrictions on cyclic formulaArc networks are prescribed.

```
<loc XLink:type=”locator” XLink:href=”ifrs-ci-2—3-10-15.xsd#currentAssets”
XLink:label=”B”/>
<formulaArc XLink:type=”arc” XLink:from=”B” XLink:to=”A” XLink:arcrole=”  http://
www.Ubmatrix.com/2003/arcrole/concept-formula”/>
<formula XLink:type=”resource” XLink:label=”A” XLink:role=”http://www.UBmatrix.com/2003/
role/formula”
xml:lang=»en» select=»ci:Land < ci:Land[-P1Y]» />
The xml:lang attribute
```

All formula resources must contain an xml:lang attribute identifying the language used for the content of the formula. The value of the xml:lang attribute must conform to XML rules. (For now this defaults to the “showLanguage” taxonomy property and is not user-settable.)

The Select Attribute

The formula element has a "select" attribute which is a string. The value of the "select" attribute is an Excel-style formula expression. Every variable which is displayed in the "select" attribute string must match a taxonomy element. (These elements are expressed as fact values within XBRL instance documents.)

A relative context expression or relative context name maybe be in "[ ]" following the element, or an absolute context where a "$" character precedes the element name (per Excel convention). Constants and formulas also follow the Excel grammar. Tuple results follow array expressions of the excel grammar.

The data type of the Excel formula expression must match the data type of the content of the schema element for which it is a resource. If the schema element is a tuple, then the formula expression evaluates to an Excel formula array result corresponding to the tuple element values.

(Note that a leading "=" character is used by Excel to identify that the contents of a spreadsheet cell are a formula, just as a leading apostrophe indicates a character string is provided. The leading "=" is not part of the formula grammar or expression. It is optional, and ignored if it is the very first character.)

Example	Explanation
"Hello, World!" or 'Hello, World!'	The string Hello, World! A select without any variables is valid. Notice how the string is quoted.
Abc:assets = (abc:liabilities + abc:equity)	A Boolean valued Excel formula expression using three variables. No context is specified in the formula (though there may be contexts on calculation arcs to the formula), so as shown this formula pertains to all occurrences of these element items, at once to all contexts, of an instance document.
abc:netIncome / abc:commonSharesOutstanding	A monetary or decimal valued Excel formula expression using two element items. The usual Excel arithmetic operators are preserved ("/" instead of Xpath "div").

Formulas

The grammar of a formula is as follows:

Formula ::= ["="] <a combination of element item references, formula references, operators, functions, numbers and other primitives.> (The leading "=" is allowed but not required, for compatibility with Excel's use of a "=" to indicate a cell has a formula, instead of a character string or other.)

Formulas can refer to element items from an instance documents.

ElementItemReference ::= [taxonomyPrefix:]elementName["["absoluteContext | relativeContext | relativeContextExpression"]"]

A taxonomyPrefix is required when the taxonomy prefix is not the default namespace of the element item (which is the usual case). A prefix provided is that of the taxonomy, not that of an instance document, and at execution time the taxonomy prefix is correctly translated, by namespace identifier matching, to the instance's xmlns-specified namespace prefixes.

Formulas can refer to results of other formulas in the current document by the taxonomy prefixed name of the element which has that formula. The formula is an IDREF in the namespace of the taxonomy, not the instance document. (More T.B.D. for formula references to formula results.)

For business rules, formulas generally belong to extension taxonomy elements, producing fact values to be saved in an instance document. Generally their elements do not participate in calculation linkbase relations. These formulas are executed after any calculation roll-up computation of missing values (if requested). The order of the formula execution is determined by a dependency analysis and an attempt to recycle if needed to produce results where cross-context terms preclude static dependency analysis.

For data validation, where the use of a formula is appropriate instead of matching a roll-up weighted sum to an instance value, a formula can be provided. In this case, the calculation tree provides a definite execution order to determine the generation of weighted-sum intermediate values, which may be accessed as terms of the formula by their element names. Formulas used for data validation are matched to the instance values found in source data and matching errors are indicated in the calculation trace report. Formulas of numeric elements participating in calculation relations are executed during the calculation roll-up processing. Calculation of fact values of elements unrelated on the calculation tree are ordered by dependency analysis and re-cycled if needed to utilize results from other formulas.

In the future a formula may refer to an external formula or element item reference, of another instance document, by the Excel workbook syntax, "[" href "]" before element item reference within that other instance document. An example of such a reference might be "[http://myco.com/myInstDoc.xml]abc:def[-P1Y]" for the element def, in prefixed namespace abc, of instance document myInstDoc.xml, by Web site http://myco.com, offset so that any context receiving this data obtains it from the previous year's context in the source.

Terms of expressions contain element item references styled to Excel's RC style. An element name has any needed "context" in square brackets. However, Excel style also permits square bracketed terms in front of an element item reference, where it means the external origin of an instance document (file).

Tuples are expressed as array results. Arrays are enclosed in curly brackets.For example, "{" multiple results "}" places these multiple results in an array which can be assigned to element items within a tuple. BNF

BNF for the formula expression is:

<expression> ::= <term> [<addop> <term>]*

<term> ::= <factor> [<mulop> <factor>]*

<factor> ::= <number> | (<expression>) | <cellRef> | <function>

<function> ::= <functionName> ([expression [, expression]*])

Built-in operators

The operators BNF is:

Operator ::= ComparisonOperator | ArithmaticOperator | ConcatinationOperator | OtherOperator

ComparisonOperator ::= “<=” | “>=” | “<” | “>” | ”“<>” | “=” (also “=>” allowed for “>=”, and “=<” for “<=”)

ArithmaticOperator ::= “+” | “-“ | “*” | “/” | “%” | “^”

ConcatinationOperator ::= “&”

OtherOperator ::= ”“,” (union – combining references) | “[…]” (relative-reference)

Operator preference for order of evaluation of multiple operators in one formula is:

, Union

Negation

% Percentage

^ Exponentiation

or / Multiplication or Division

+ or – Addition or Subtraction

& Text Concatenation Operator

= < > <= >= <> Comparison Operators

Arithmetic operators perform basic mathematical operations such as addition, subtraction, or multiplication; combine numbers; and produce numeric results:

Table 89 Arithmetic Operators

Arithmetic Operator	Meaning	Example
+ (plus sign)	Addition	3+3
- (minus sign)	Subtraction	3-1
-	Negation	-1
* (asterisk)	Multiplication	3*3
/ (forward slash)	Division	3/3
% (percent sign)	Percent	20%
^ (caret)	Exponentiation	3^2

Comparison operators compare two values and then produce the Boolean value TRUE or FALSE.

Table 90 Comparison Operators

Comparison Operator	Meaning	Example
= (equal sign)	Equal to	A1=B1
> (greater than sign)	Greater than	A1>B1
< (less than sign)	Less than	A1<B1

Comparison Operator	Meaning	Example
>= (greater than or equal to sign)	Greater than or equal to	A1>=B1
<= (less than or equal to sign)	Less than or equal to	A1<=B1
<> (not equal to sign)	Not equal to	A1<>B1

The string operator "&" combines one or more string values to produce one string result:

Table 91 String Operator

Text Operator	Meaning	Example
& (ampersand)	Connects, or concatenates, two values to produce one continuous text value.	"Disc" & "closure" produces "Disclosure".

Reference operators combine ranges of cells for calculations. Note Microsoft Excel reference operators are only useful for spreadsheet cells (the range, ":" and intersection " ") and are not included here.

Table 92 Reference Operators

Reference Operator	Meaning	Example
, (comma)	The Union operator combines multiple references into one reference (here, suitable to be contents of an array expression assigned to a tuple).	name, address, phone
[(relative reference)]		The Relative Reference operator designates a relative context expression, or a name of an absolute or relative context resource to apply to this element.

Operator Precedence

The operator precedence allows you to code familiar expressions without parentheses, for example, use 3 + 4 * 5 to produces the result 23 (customary) instead of 35 (not usual). All expression parsing is left to right given precedence considerations. If operators have the same precedence, they are handled in lexical order, for example, left to right. The calculation trace of Callisto's instance document validation feature enables you to see how the terms are parsed and ordered for execution.

Table 93 Operator Precedence

Precedence	Operators
Highest	Infix operators, '-' (negate), '%' (percent)
	Exponentiation: '^'
	Multiply, divide: '*', '/'
	Add, subtract: '+', '-'

Precedence	Operators
	Concatenate: '&'
	Comparison: '>=', '<=', '>'. '<'
	Equality: '=', '<>'
	Xor
	And
	Or
Lowest	Union, parentheses, function terms: '(', ',', ')'

The operand conversion allows you to mix unlike operands in formula execution. Unlike operands are "coerced" into the more general of the two. For example adding an integer and a double floating number is handled by converting the integer to double floating and then adding both in double floating form. During the formula execution of fact values, the decimals or precision aspects, and units, of each numeric fact are maintained. The precision of a result becomes the least of operand pairs.

In general, strings containing numbers in arithmetic expressions convert the number to floating point. Booleans are converted to zero or one (false or true) if mixed in arithmetic expression terms.

Date-time expressions may be compared to each other, but do not convert into arithmetic units (one must use a function to extract arithmetic-suitable days, hours, etc, from a date-time field).

Table 94 Logical Functions

Name	Number of Input Parameters	Description	Syntax
Logical functions			
If	3	If the first argument, as a Boolean, is true, returns the second argument (any type), otherwise it return the third argument (any type).	If(condition, value_if_true, value_if_false) condition is the value to test. value_if_true is the value that is returned if condition evaluates to TRUE. value_if_false is the value that is return if condition evaluates to FALSE
TRUE	0	The Boolean true result	TRUE
FALSE	0	The Boolean false result	FALSE
Nothing	0		
Quit	0	This result is the equivalent of a formula failing to fire, it returns no result.	Quit, Quit()

Name	Number of Input Parameters	Description	Syntax
Choose	1, <1 or more>	The Choose function returns a value from a list of values based on a given position.	Choose (position, value1, value2, ... value_n) position is position number in the list of values to return. value1, value2, ... value_n is a list A value can be any one of these: a number, a conceptname, a formula/ function, or a text value.

Table 95 String Functions

Name	Number of Input Parameters	Description	Syntax
Len	1	The Len function returns the length of the specified string. (in, or converted to, string type).	Len(<text or Value>)
Left	2	The Left function enables you to extract a substring from a string, starting from the left-most character.	Left(text, number_of_characters) text is the string that you wish to extract from (string conceptname, function that returns a string) number_of_characters indicates the number of characters that you wish to extract starting from the left-most character.
Right	2	The Right function extracts a substring from a string starting from the right-most character.	Right(text, number_of_characters) text is the string that you wish to extract from (string conceptname, function that returns a string) number_of_characters indicates the number of characters that you wish to extract starting from the right-most character.
Concatenate	2	The Concatenate function enables you to join two strings together. It is similar to “&”.	concatenate(string1, string2)
Mid	3	The Mid function extracts a substring from a string (starting at any position).	Mid(text, start_position, number_of_characters) text is the string that you wish to extract from. start_position indicates the position in the string from where you begin extracting. The first position in the string is 1. number_of_characters indicates the number of characters that you wish to extract
Trim	1	The Trim function returns a text value with the leading and trailing spaces removed.	Trim(text) text is the text value to remove the leading and trailing spaces from.
Replace	3 or 4		

Name	Number of Input Parameters	Description	Syntax
Substitute	3 or 4	The Substitute function replaces a set of characters with another.	Substitute(text, old_text, new_text, nth_appearance) text is the original string to use to perform the substitution. old_text is the characters to replace. new_text is the characters to replace old_text with. nth_appearance is optional. It is the nth appearance of old_text that you wish to replace. I f this parameter is omitted, then every occurrence of old_text is replaced with new_text
Char	1	The Char function returns the character based on the NUMBER code.	Char(ascii_code)
Search	2	The Search function returns the location of a substring in a string. The search is NOT case-sensitive.	Search(text1, text2) text1 is the substring to search for in text2. text2 is the string to search.
Exact	2	The Exact function compares two strings and returns TRUE if the values match. Otherwise, it returns FALSE.	Exact(text1, text2 text1 and text2 are the values to compare. The Exact function is case-sensitive
Upper	1	The Upper function converts all letters in the specified string to uppercase. If there are characters in the string that are not letters, they are unaffected by this function.	Upper(text1)
Lower	1	The Lower function converts all letters in the specified string to lowercase. If there are characters in the string that are not letters, they are unaffected by this function.	Lower(text1)
Proper	1	The Proper function sets the first character in each word to uppercase and the rest to lowercase.	Proper(text1)
Value	1	The Value function converts a text value that represents a number to a number.	Value(text1) text1 is the string that represents number

Table 96 Math Functions

Name	Number of Input Parameters	Description	Syntax
Abs	1	The Abs function returns the absolute value of the argument.	Abs(number1)
Ceiling	1	The Ceiling function returns a number rounded up	Ceiling(number1)
Floor	1	The Floor function returns a number rounded down	Floor(number1)

Name	Number of Input Parameters	Description	Syntax
Round	1	The Round function rounds to nearest integer	Round(number1)
RoundDown	1	same as Floor	
Roundup	1	Same as Ceiling	
Truncate	1	truncates the number	Truncate(decimalnumber1)
Trunc	1	same as Truncate	
Int	1	Converts the input to a string. And also truncates if required.	Int('1234') = 1234
Sum	<1 or more>	Adds all numbers together.	Sum(n1, n2, n3, ... Nn)
SumSquares	<1 or more>	Adds all squares of the numbers together	SumSquares(n1, n2, n3, ... Nn)
Sqrt	1	Returns the Square root of the argument with (base 2).	Sqrt(number)
Min	2	Returns the minimum of the two numbers.	Min(number1, number2) returns number1 if number1 < number2
Max	2	Returns the maximum of the two numbers.	Max(number1, number2) returns number1 if number1 > number2
Sign	1	Returns -1 if the number is negative, else returns 1.	Sign(number1)
Power	2	The Power function returns the result of a number raised to a given power.	Power(number, power) number is a base number. power is the exponent used to raise the base number to.
Avg	<1 or more>	The Avg Functions returns the avg of the numbers provided as arguments	Avg(n1, n2, n3, ... Nn)
Rational	1		

Table 97 Datetime Functions

Name	Number of Input Parameters	Description	Syntax
DateTime	0	Returns the current date-time (for example, "Now"), as a dateTime	
DateNow	0	Returns the date portion of current date-time, as a dateTime.	
TimeNow	0	Returns the time-of-day portion of current date-time, as a duration.	

Name	Number of Input Parameters	Description	Syntax
DateOf	1	Returns the date portion of the date-time type argument, as a dateTime.	
YearOf	1	returns the time-of-day portion of date-time argument, as a duration.	
MonthOf	1	returns the year portion of the date-time type argument, as an integer.	
DayOf	1	returns the month portion of the date-time type argument, as an integer.	
IsEndOfMonth	1	returns the day portion of the date-time type argument.	
HourOf	1	returns true if date-time type argument is last day of month.	
MinuteOf	1	returns the hour portion of the date-time type argument.	
TimeOf	1	returns the minute portion of the date-time type argument.	
Offset	2	returns the date-time of the first argument (date-time) offset by the second argument (duration).	
DateToString	2	returns the string formatted date as specified by the second argument (C# date format)	DateToString(Date, DateFormat) For example: DateToString(Context.Period.EndDate, 'yyyyMMDD')
DateValue	1	returns the serial number of a date.	

Name	Number of Input Parameters	Description	Syntax
Date	3	returns the date	Date(year, month, day) year is a number that is between one and four digits. If the year is between 0 and 1899, the year value is added to 1900 to determine the year. If the year is between 1900 and 9999, the Date function uses the year value as the year. If the year is greater than 9999, the Date function returns the #NUM! error. month is a number representing the month value. If the month value is greater than 12, the Date function adds that number of months to the first month of the year specified in the year parameter. day is a number representing the day value. If the day value is greater than the number of days in the month specified, the Date function add that number of days to the first day in the month specified in the month parameter.
Day	1	returns the day of the month (a number from 1 to 31) given a date value.	Day(date_value) date_value is a valid date.
Month	1	The Month function returns the month (a number from 1 to 12) given a date value.	Month(date_value) date_value is a valid date.
Year	1	The Year function returns a four-digit year (a number from 1900 to 9999) given a date value.	Year(date_value) date_value is a valid date.

Table 98 Trigonometric Functions

Name	Number of Input Parameters	Description	Syntax
Acos	1	returns the angle whose trig argument is specified	
Asin	1	returns the angle whose trig argument is specified	
Atan	1	returns the angle whose trig argument is specified	
Atan2	2	returns the angle whose trig argument is specified	
Sin, Cos, Tan	1	returns the trig function based on the radians angle argument.	
Cosh	1	returns the hyperbolic trig function of the argument	
Degrees	1	returns the radians argument converted to degrees (arg * 180 /)	
Ln	1	returns the logarithm, natural log, or base 10 logarithm of the argument	
Log	1	returns the logarithm, natural log, or base 10 logarithm of the argument	

Name	Number of Input Parameters	Description	Syntax
Log10	1	returns the logarithm, natural log, or base 10 logarithm of the argument	
Pi	0	the maximum resolution of (3.14...).	
Radians	1	returns the radians value of the degrees argument (arg * / 180).	
Tanh	1	returns the hyperbolic trig function of the argument	

Table 99 Pre-binding Functions

Name	Number of Input Parameters	Description	Syntax
Exists	1	returns True/False based on the pre-binding value of the instance document fact for the current context. Returns True when the Fact value is nil only if the Nilable attribute is set true	Exists(prefix:<conceptname>)
ExistsNonNil	1	eturns True/False based on the pre-binding value of the instance document fact for the current context. Returns True only if the Fact has a reported value regardless of the XBRL Nilable attribute	
ExistsNonEmpty	1	returns True/False based on the pre-binding value of the instance document fact for the current context. Returns True only if the Fact has a reported value and is non empty	ExistsNonEmpty (TotalAssets)
IsNil	1		
ExistingOf	<1 or more>	returns the first found value in the list of arguments. If no arguments bind the function returns no value and the expression fails to bind	ExistingOf(TotalAssets, '0')

Table 100 Context Functions

Name	Number of Input Parameters	Description	Syntax
`Context.Period.Type`	0	the current context's type (for example, "Instant"), as a string.	
`Context.Period.StartDate`, `Context.Period.EndDate`	0	the current context's start and end dates (or instant date), as a date-time type.	

Absolute Context Resources (Filters)

Absolute contexts define constraints that allow an expression term, and the formula result, to be restricted to certain contexts. (This is currently adapted from a May 2002 V2.1 proposal.) Only a formula linkbase MAY contain elements to define absolute contexts. An absoluteContext element contains one sub-element for each of the types of constraints that can be expressed: entity, segment, scenario and period. If a sub-element does not appear in an absoluteContext, then the calculation is not constrained to match any value regarding that aspect of its context. An empty or missing absolute context means that the calculationArc can be applied across all contexts having any entity, any period, and any scenario.

```
<element name="absoluteContext">
 <complexType>
 <sequence>
 <element name="periodConstraint" minOccurs="0" maxOccurs="unbounded">
 <complexType>
 <sequence>
 <element name="instantConstraint" minOccurs="0" maxOccurs="unbounded">
 <complexType>
 <attribute name="year" type="string" form="unqualified"/>
 <attribute name="month" type="string" form="unqualified"/>
 <attribute name="day" type="string" form="unqualified"/>
 </complexType>
 </element>
 </sequence>
 </complexType>
 </element>
 </sequence>
 <attribute name="id" type="string" form="unqualified"/>
 <attribute ref="XLink:title" use="optional"/>
 <attribute ref="XLink:type" use="optional"/>
 </complexType>
 </element>
```

An Excel-style expression term utilizes an absolute context filter by placing its name in square brackets following the qName of the element. If a term with such a filter can't find a value in the fact values of the instance, relative to an execution context of the formula's owning taxonomy element, then that context of the formula does not execute, as if it were Nil or absent.

Table 101 Absolute Context Examples

Example	Matches

`<absoluteContext id="after2003">`
`<periodConstraint>`
`<startDateConstraint minimum="2003-01-01"/>`
`</periodConstraint>`
`</absoluteContext>`
	Any context with a startDate on or after 1st January 2003.

`<absoluteContext id="payday">`
`<periodConstraint>`
`<instantConstraint day="15 30"/>`
`</periodConstraint>`
`</absoluteContext>`
	Any context describing an instant falling on the 15th or 30th of any month.

`<absoluteContext id="any-year">`
`<periodConstraint>`
`<durationConstraint year="1"/>`
`</periodConstraint>`
`</absoluteContext>`
	Any context with a period whose start date and end date differ by one year.

Example	Matches
<absoluteContext id="calendar-year"> <periodConstraint> <startDateConstraint monthDay="01-01"/> <durationConstraint offset="P1Y"/> <endDateConstraint monthDay="12-31"/> </periodConstraint> </absoluteContext>	Any context with a period whose start date and end date differs by one year and starts on 1 January and ends 31 December of that year.
<absoluteContext id="birth"> <entityConstraint> <identifier scheme="http://www.state.us">NM</identifier> </entityConstraint> <periodConstraint> <instantConstraint day="19 20" month="10" year="1998"/> </periodConstraint> </absoluteContext>	Any context in which the state of New Mexico is the entity and the date is the 19th or 20th of October 1998.
<absoluteContext> <scenarioConstraint><my:budget/></scenarioConstraint> </absoluteContext>	Matches contexts in which the scenario element contains a "budget" element.

An entityConstraint is used to restrict calculations to apply only to facts pertaining to a entity or entities. Multiple identifiers mean that multiple entities MAY match. Absence of an entityConstraint element means that all entities match.

Any number of segmentConstraint elements can be used to further restrict applicability of the arc. Multiple segment constraints means that either MAY match. Absence of the segmentConstraint element means that all segments match.

A scenarioConstraint element restricts applicability of the arc to contexts with scenario elements. Any empty portion is taken to match any segment.

A periodConstraint element restricts applicability of an arc to contexts whose periods match features. Any empty portion of the periodConstraint is taken to match any period. When multiple elements appear, a context MAY match any or all of them.

The periodConstraint sub-elements contain attributes that allow individual parts of the date to be constrained, and allow minimum and maximum date values to be set. Except for the maximum and minimum attributes, the attributes allow multiple values to be specified as a whitespace-separated list of numbers.

Table 102 More Absolute Context Examples

Example	Matches
<absoluteContext id="monthStart"> <periodConstraint> <instantConstraint day="01"/> </periodConstraint> </absoluteContext>	Any context that is the beginning of a month.

Example	Matches
<absoluteContext id="thirteenMonths"> <periodConstraint> <startDateConstraint day="22" year="1962" month="10"/> <endDateConstraint day="22" year="1963" month="11"/> </periodConstraint> </absoluteContext>	The period of 13 months starting 22 October 1962.
<absoluteContext id="afterPublicLaunchOfXBRL"> <periodConstraint> <startDateConstraint minimum="2000-04-06"/> </periodConstraint> </absoluteContext>	Any period starting on or after 6th April 2000.
<absoluteContext id="endOfFiscalQuarter"> <periodConstraint> <instantConstraint monthDay="03-31 06-30 09-30 12-31"/> </periodConstraint> </absoluteContext>	Any instant falling at the end of any fiscal quarter.
<absoluteContext id="idesOfMarch21stCentury"> <periodConstraint> <instantConstraint yearMonth="2001-03 2002-03 2003-03" day="15"/> </periodConstraint> </absoluteContext>	The Ides of March that occurred in the current millennium, so far.

Relative Context Expressions (Offsets)

Relative context expressions are an in-line term constant to express a relationship between the current execution context of a formula and a term's appropriate context (for example, to get a term from a prior year period), without use of a Relative Context Resource. A relative context expression offsets a term context after application of any (possibly multiple) absolute and relative context resources. A forever period is unaffected, an instant period is offset, as are both start and end dates of a duration period. The format of the relative context expression is to place it in square brackets separate from any square brackets containing absolute and relative context expressions. Regardless of order, absolute and relative context expressions perform filtering and relating first, and the expression is applied last.

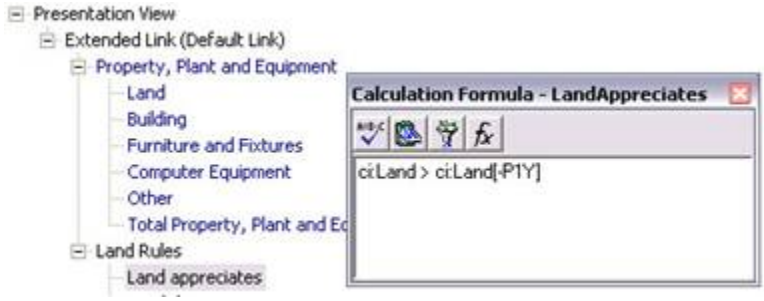
[(relative reference)]	Relative Reference operator, which designates a relative context expression.	ci:Land[-P1Y] is the value of the ci:Land fact of the preceeding-year's context

Relative Reference (Cross-context) Expressions

There are cross-context expressions of three types: (1) relative period-offset expressions, (2) named relative context resource, and (3) named absolute context resource.

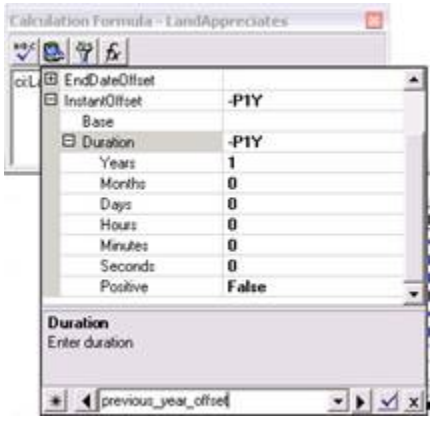
An excel-syntax relative reference expression is in square brackets following the prefixed element name.

Relative period-offset expressions use the Xbrl/Xml period expression. The expression ci:Land[-P1Y] means that the formula executes if the ci-prefixed Land-named element has a fact value one year earlier than the formula's execution context, and utilizes the previous year's value for this expression term.



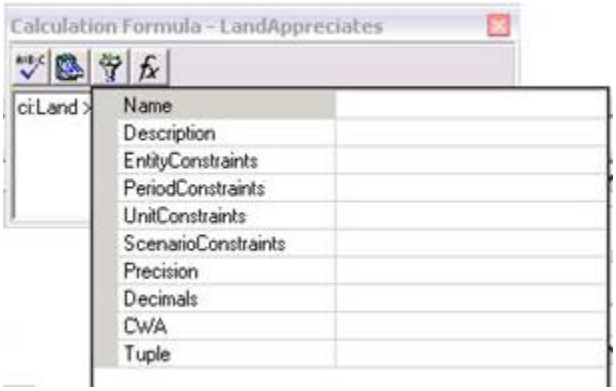
In this example, the formula executes for each context where ci:Land exists in the current period and in another like context one year earlier. If both values are not present, the formula is not "executed" and no value or test results.

Relative period-offset expressions based on an end-of-month instant date yield an end-of-month relative offset date. January 31 offset +P1M produces February 28 (or 29 in a leap year).



Named relative context resources are entered, reviewed, and selected by a formula editor panel; they allow specification of start-date, end-date, and instant-date offset, based on the start or end of an interval period. Named relative context resources can be entered once and used many times in the formulas of a given formula linkbase. They are stored in the formula linkbase.

Named absolute context resources are also entered, reviewed, and selected by a formula editor panel. An absolute context resource is a filter (hence the "funnel" icon). It provides constraints on entity, period, scenario, and tuple, and in the V2.0 version also unit, precision, decimals, and CWA.



Variables and Constants

Every variable and constant comes from an instance document. This allows for the chaining together of formula computations into a series.

Multiple layers of intermediate variables can be provided in the business rules (extension taxonomy) situation by incorporating one (or more) extension taxonomies. The outermost extension taxonomy has its formula-result fact values saved in an instance document file. Intermediate extension taxonomies can be provided as fact values collections for common sub-expressions, for constant data such as ranges and factors, and other purposes as useful to the analyst.

Constants assume their most natural form. Thus 1 or 1001 are integers, 1.0 or 1.001e3 are double precision floating point, and Rational("33 1/3") is a rational number (assumable by a fractionItemType fact value of 100/3 or assignable to any other fact value as double precision floating point value 33.333333). "abc" and "123" are character strings.

Values Returned

Formulas produce a result which is assigned to the formula's XBRL taxonomy element fact values. The reason for this is (1) for the business analysis case, extension taxonomy formulas, to allow an XBRL instance document to be created as a result of calculation a formula, and (2) for the data validation case, to allow formula produced fact values to be compared to instance document fact values for source data validation. Some values returned may be "true" of "false" to indicate if a formula passes or fails, or even descriptive character strings. Or, values can be returned and instance documents created. Or, values can be computed and inserted in an instance document. Formulas can be used to achieve each of these results. These features allow for the chaining together of a series of instance documents.

Formula Examples and Use Cases

Table 103 provides a summary of formula examples and use cases which explain how formulas can be used.

Table 103 Formula Examples and Use Cases

Contents of Select Attribute	Explanation
ci:Land + ci:Building	The value returned is the value of "ci:Land" plus "ci:Building".
Ci:Land * .80	The taxonomy element "ci:Land" times .80
ci:Land >= ci:Land[-P1Y]	The taxonomy element "ci:Land" value of the current period is greater than or equal to the value of "ci:Land" for the prior period.
If((ci:Land + ci:Building) > ((ci:Furniture + ci:Fixtures) * .8), 'Approved', 'Denied')"	If the value of "ci:Land" plus "ci:Building" is greater than 80% of the value of "ci:Furniture" plus "ci:Fixtures", then the character string value returned is "Approved". Otherwise, the value returned is "Denied".
Ci:PropertyDiameter / Pi	The value returned is the element "ci: PropertyDiameter" divided by the constant "Pi" (3.14....).
(ci:Land > 100000) And (ci:Building > 100000)	The value of "ci:Land" must be greater than 100000 AND the value of "ci:Building" must be greater than 100000, or the value of False is returned.
(ci:Land > 50000) Or (ci:Building > 50000)	The value of "ci:Land" must be greater than 50000 OR the value of "ci:Building" must be greater than 50000.
Ci:Land + ci:Building + ci:FurnitureFixtures + ci:ComputerEquipment + ci:Other	The value returned is the total of "ci:Land" plus "ci:Building" plus "ci:FurnitureFixtures" plus "ci:ComputerEquipment" plus "ci:Other". This works much like a weighted roll up of subordinate fact values of elements in a calculation linkbase tree structure.
Ci:Land[-P1Y] + ci:LandChanges [ForPeriod]	The value returned is the value of "ci:Land" with a context of one year less than the current context plus "ci:LandChanges" with a context, which is provided in the formula, less one day.
Ci:Land[-P1Y] + ci:LandChanges [ForPeriod] = ci:Land	The value returned is True if the value of "ci:Land" in the prior period plus the value of "ci:LandChanges" for the context beginning one day after the –P1Y context equals the value of "ci:Land" for the current context.

Formula Execution

Here is a summary of the formula processing model. Formulas execute:

- **Case 1**— In an instance's own taxonomy, for validation of data or generation of fact values not in source data. Generally provided by the producer of data, in the source taxonomy. Example: movement analysis development of closing balance on accounts, as instant period context, from opening balance, as instant period context, plus changes during year as duration contexts.
- **Case 2**— In an analytical taxonomy, for analysis of a subject instance (data and taxonomy) to produce derived data in the form of instances containing only the derived data. Generally provided by the consumer of data, in an extension taxonomy for analysis and evaluation of a producer's data. Example: checking a balance sheet for liquidity ratio changes over recent periods, and for minimum liquidity ratio.

In Case 1, the execution model for calculation validation checks calculation linkbase roll-up weighted sums against data appearing in an instance document. Where formulas are provided and values appear in the instance document, the formula result and instance document data are checked in the same way a roll-up weighted sum is compared to the instance data. For example,

a computed closing balance checks against any closing balance in instance data, and calculation trace reports mismatches as errors.

In Case 2, every formula which "fires" results in an fact value for an element in the analytical taxonomy. These fact values are saved as an instance document currently called a business rules results file.

When Do Formulas Execute?

In case 1, they execute when their element would be rolled-up if on the calculation linkbase tree. If the element owning the formula has no relationships, and is at the root level on the calculation tree, it executes last, after calculation roll ups finished.

In case 2, they execute last, separately from any calculation tree roll ups in the subject instance being analyzed.

What Does It Mean for a Formula to "Fire"?

A formula executes when its element would be found in walking the calculation tree (or at root to the tree). It executes by trying to produce a result for every context in the fact value set. For each context it tries to get each term (element reference) according to whether that term is in the same context as the formula is executing or offset by relative context or filtered by absolute context.

If a formula finds all of its terms for a given context, it is said to have "fired". The calculation trace clearly shows each formula in each context attempting to gather its data and trying to complete, and when it gives up if a term it requires lacks a corresponding fact value.

A relative context is an offset for getting a term (and if property grids restored, also an offset for the result context). The relative context can grab a duration term for an instant context result and vice versa (for example, movement analysis example). If the formula is producing an instant context result, and the relative context has start and/or end terms, the term offset by that relative context provides a relative context fact value.

Conversely a duration context formula result can obtain an instant context fact value by basing the instant property of the relative context on the start or end of the period, plus date offset. Offsets can be positive or negative, and each term can likewise be positive or negative.

For example, for a result context of 12/31/02, the relative context to grab data from duration context 1/1/02 – 12/31/02 requires a start offset of minus one year (back to 12/31/01) and plus one day (forward to 1/1/02). Month end dates go back or forward to their respective month end dates. 2/28/01 plus one month is 3/31/01. 3/31/04 back 1 month is 2/29/04. Any other date (month start, month middle) just counts months, so there is no clean way to go from the middle date of February to middle date of March without complex coding.

An absolute context is a filter for getting a term from another context. For example to grab a term from a subsidiary entity and use it in a formula whose result context is the parent entity.

The formula expression for trivial contexts uses the XML duration notation, for example, P1Y means 1 year forward, -P1Y means 1 year backward, and from Dec 31 to Jan 1 would be –P1Y-1D.

The formula expression for nontrivial relative contexts is to use the GUI menu collection editor to select or create such a context. These relative contexts (and following absolute contexts) which are entered to the GUI are saved as resources in the formula linkbase. Their names are local to the formula linkbase (though at one time the names were supposed to be taxonomy, not linkbase, specific).

The GUI enters a relative context when the cursor is positioned after a prefixed element name, and pressing the relative context button (looks like an old fashioned desk calendar with page leaves for date). You can scroll through contexts, or enter a relative context (star button), or delete unneeded contexts (brush-stroke x button). Pressing the check-mark button closes this collection editor and places the name, after parentheses, where the formula text cursor was.

The GUI enters an absolute context in the same manner, using the absolute context button (a funnel, depicting filtering out data). It behaves in the same manner.

Relative Context Resources (Offsets)

Relative contexts express a relationship between the current execution context of a formula and a term's appropriate context (for example, to get a term from a prior year period), or to offset the result of a formula from the current formula context to another (such as to establish the opening balance for next period). A relative context uses a context as a base and produces a matching context. The formula linkbase MAY contain resources that define relative contexts. The content of the relativeContext element are sub-elements that express functions of the sub-elements and attributes of the item context. Any missing element or attribute is copied from the base context. Therefore, an empty relativeContext means that there are no differences between the base context and the output context; the two must share s-equal contexts.

```
<element name="relativeContext">
 <complexType>
 <sequence>
 <element name="periodOffset" minOccurs="0" maxOccurs="unbounded">
 <complexType>
 <sequence>
 <element name="instantOffset" minOccurs="0" maxOccurs="unbounded">
 <complexType>
 <attribute name="base" type="string" form="unqualified"/>
 <attribute name="offset" type="string" form="unqualified"/>
 </complexType>
 </element>
 </sequence>
 </complexType>
 </element>
 </sequence>
 <attribute name="id" type="string" form="unqualified"/>
 <attribute ref="XLink:title" use="optional"/>
 <attribute ref="XLink:type" use="optional"/>
 </complexType>
 </element>
```

An Excel-style expression term utilizes a relative context by placing its name in square brackets following the qName of the element, and optionally comma-separated from any preceding absolute context (filter). The relative context of the term is an offset to the context of the

formula's owning taxonomy element. If there is no source data for a term's relative context, then that context of the taxonomy element's formula does not execute (as if it were Nil or absent).

A formula offsets its result fact value by a relative context by placing the relative context resource id in the relative context of the property grid for the calculation arc to its parent calculation element.

Table 104 Examples of Relative Contexts

Context	Description
`<relativeContext id="same"/>`	An empty relative context means that the output context is s-equal to that of the input.
`<relativeContext id="period-end"> <periodOffset> <instantOffset base="end"/> </periodOffset> </relativeContext>`	Given an input item's context, this produces a matching context that is s-equal in all respects except that its period is the instant that marks the end of the input period.
`<relativeContext id="period-start"> <periodOffset> <instantOffset base="start"/> </periodOffset> </relativeContext>`	Given an input item's context, this produces a matching context that is s-equal in all respects except that its period is the instant that marks the start of the input period.
`<relativeContext id="prior-year-end"> <periodOffset> <instantOffset base="end" offset="-P1Y"/> </periodOffset> </relativeContext>`	Produces that matching context which is s-equal in all respect except that its period is an instant whose year is 1 less than the endDate of the input period.
`<relativeContext id="next-year"> <periodOffset> <startDateOffset base="end" offset="-P1D"/> <endDateOffset base="end" year="P1D"/> </periodOffset> </relativeContext>`	The output context has a period that starts on the day before the input period ends, and ends one year later.

The content of the periodOffset element in relativeContext is similar to the element content of the period element in contexts. The sub-elements of period contain attributes that allow individual parts of the relative period to be offset from portions of the base period. The sub-elements of relativeContextinclude are relativePeriod, instantOffset, startDateOffset, and endDateOffset.

The elements relativeInstant, endDateOffset, and startDateOffset appearing in relativeContext contain an optional attribute offset whose content is a duration that specifies a relative period, and an optional attribute base that specifies whether the period is relative to the start or end of the input context. The offset attribute defaults to the zero length duration P0D.

A context with the period forever matches any output context. The base, relativeStartDate, endDateOffset and instantOffset elements are all ignored in that case.

Creating a Business Rule Using the Properties Grid

In the steps above, we used the Properties Grid to work with a property of the taxonomy called the Referenced Taxonomies property.

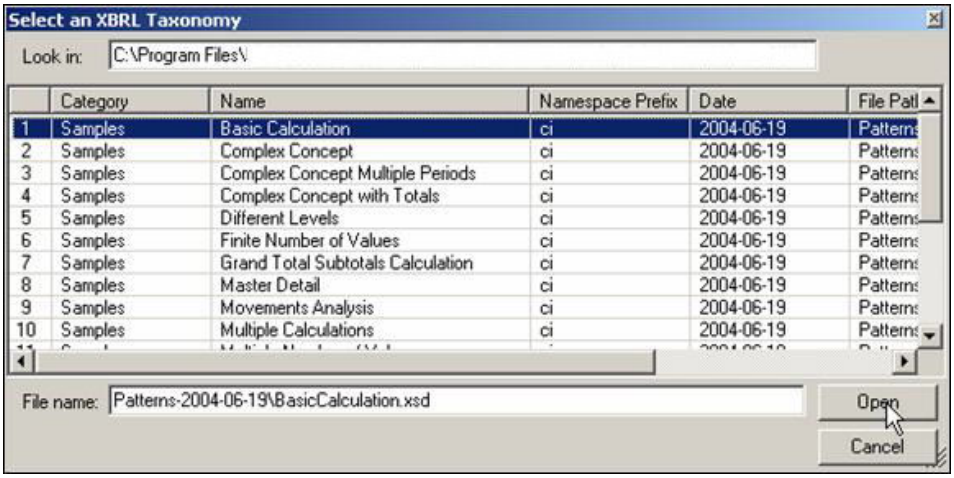
Now we use the Properties Grid to work with a property associated with one element within the taxonomy .

➤ To do this:

1 Select **File**, then **Open**, and then **Taxonomy**.

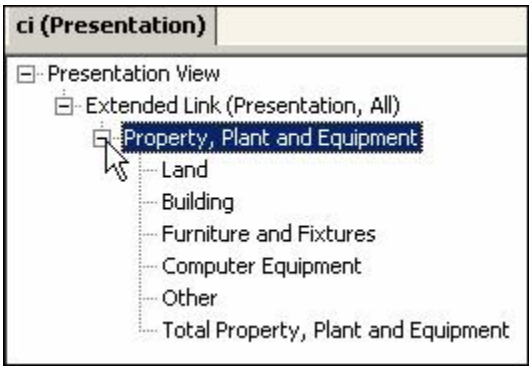
2 Within the **Open Taxonomy** window, select the Open icon to open the local taxonomy directory.

3 In the **Open XBRL Taxonomy** pop-up window, select the **Basic Calculation** taxonomy and click **Open**.



4 Click **OK** to load the taxonomy and close the **Open Taxonomy** window.

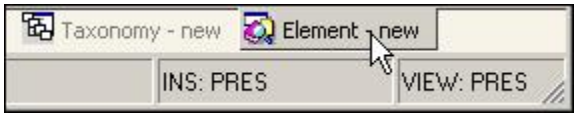
5 In the **Relationships View** pane (Presentation), click to expand the elements under Property, Plant, and Equipment.



6 Click to select the element **Property, Plant, and Equipment**.

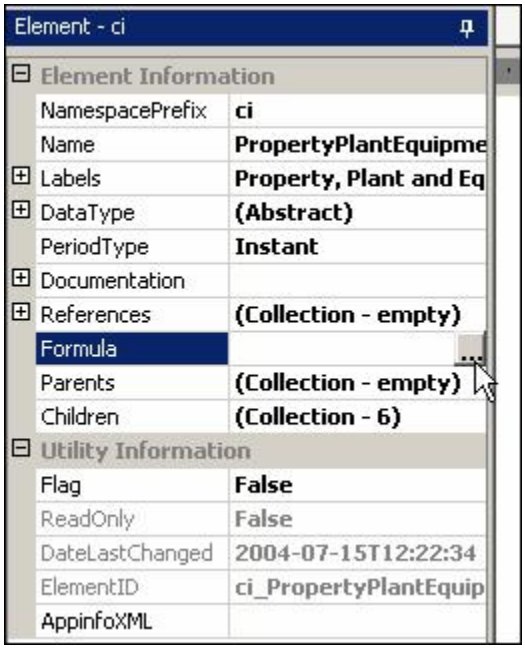
We now add a business rule associated with this element. This rule checks the formula associated with this element to give a True or False response.

7 At the bottom of the **Properties Grid** (with the element selected), click the **Element** tab to display element properties in the grid.

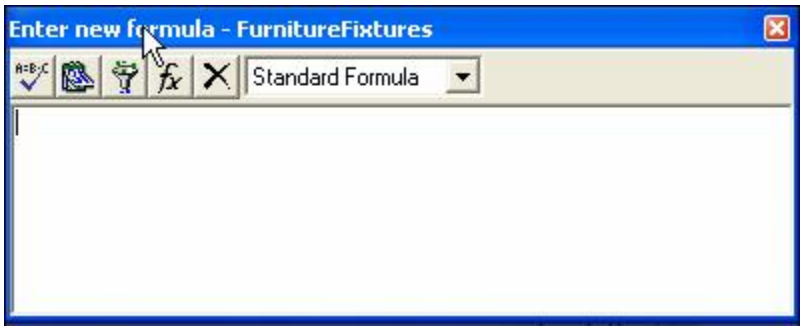


8 Select the **Formulas** property from the Elements Property Grid.

An ellipsis button is displayed in the associated field.



9 Click the button.



10 From the **Relationships View** pane, left-click the element Land from within the taxonomy tree and hold the left mouse button down while you drag it into the **Enter new formula** window.

11 Release the mouse button to drop the **Land** element into the formulas form.

12 Enter **>** (greater than symbol) into the formula.

13 Next, drag and drop the element **Other** from the **Relationships View** pane.

Your formula should look like this:

14 Click **Check Formula** to validate the formula.

15 Click the **X** in the upper right hand corner to close the **Enter new formula** window.

16 Select **Property, Plant, and Equipment.**

Note that there is a formula in the Element Properties Grid, indicating that a business rule is associated with this element.

17 Click **Save** and then **Close.**

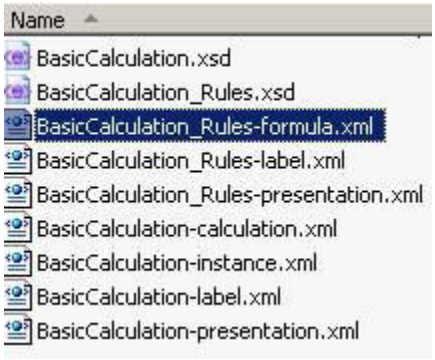
Creating Formulas with XBRL Taxonomy Designer

Formulas (or Business Rules) can be expressed using the Disclosure Management propriety formula format, and saved in the propriety format or the XBRL international formula 1.0 specification format. The XBRL 1.0 formula specification is currently published as a candidate recommendation (February 1, 2006) and is still a work in progress. Each formula format is implemented as a resource within a dedicated linkbase that is similar to other XBRL linkbases.

To read more on the XBRL formula 1.0 specification go to this URL: http://www.xbrl.org/SpecRequirements/

Formulas are associated with a taxonomy concept and allow a user to test the validity of data reported in an instance for one concept or other concepts, or for one element relative to another element, and other such calculations.

Formulas are stored in formula linkbase:



A Tutorials

In This Appendix

The XBRL Taxonomy Designer tutorials are divided into these sections.

- "Basic XBRL Taxonomy Designer Tutorial" on page 238 covers the basics of using XBRL Taxonomy Designer. This includes taxonomy basics, elements, relationships, reference taxonomies, and instance documents.
- "Formulas Tutorial: Creating Simple Formulas to Validate Data" on page 348 covers the basics of using formulas with XBRL Taxonomy Designer.
- "Dimensions Tutorial" on page 327 covers the basics of dimensions and using dimensions with XBRL Taxonomy Designer.
- "Mapping Tutorial" on page 284 the basics of learning to mapping. Mapping enables you to map from a non-xbrl source to an XBRL document and then convert that information into XBRL.
- "Importing Tutorial" on page 269 explains how to import and export data with XBRL Taxonomy Designer to create taxonomies and instance documents.

Before You Begin

This section provides definitions of some basic XBRL Taxonomy Designer terminology, as well as screen shots of XBRL Taxonomy Designer two main editor screens and their components. Take a moment to review the terms and use this section for reference as needed.

Table 105 Basic XBRL and XBRL Taxonomy Designer Terminology

Term	Description
Taxonomy	An XBRL taxonomy is a collection of elements, linkbases, role definitions, reference part definitions, and data type definitions. It incorporates referenced taxonomies, which contain all these parts also, as well as reference XML schemas (type definitions, other definitions).
Linkbases	Linkbases contain extended links and — depending on linkbase type — relationships, labels, formulas, or references.
Element	Each term or text entry for building a taxonomy can be viewed as one element. The Elements List pane shows elements in a dictionary style.
Extended Links	A taxonomy is a collection of extended links which contains arcs (relationships) between one element in a DTS and another element in a DTS, or a resource (label, formula, reference). Each relationship view (presentation, calculation, definition) has a minimum of one extended link, or it may contain any number of extended links. These extended links contain relationships between elements. Extended links are independent networks of relationships.
DTS (Discoverable Taxonomy Set)	The DTS is a collection of all of the taxonomies and linkbases.
Multiple Taxonomy DTS	A loaded taxonomy may reference other taxonomies explicitly through import constructs in its .xsd file or implicitly by finding references to roles, linkbase locs, or reference part definitions. Referenced taxonomies are loaded as identified and discovered, and form a DTS. The DTS is represented as a tree, viewable and editable as the Referenced Taxonomies Collection of the opened (base) taxonomy.
Relationship View	In XBRL Taxonomy Designer, the Relationships View pane shows a tree view of links or relationships between elements. There are three types of relationship views in XBRL Taxonomy Designer: Presentation, Calculation, and Definition.

Introduction to the Tutorials

These tutorials are designed to introduce you to the key steps in using XBRL by using XBRL Taxonomy Designer taxonomy developer and instance document tool.

It is important that users take time to go through these tutorials in the order they are presented, because data entered often builds on previous steps. The example files referenced are included with your installation of XBRL Taxonomy Designer, or on CD provided for the course; directions on how to locate these are given within the tutorials.

Sample Files

Below is a list of sample files used in the tutorials:

Table 106 Sample Files Used in the Tutorials

File Name	Extension	Type
Basic Calculation	.xsd	Schema
SampleCompany-Taxonomy	.xsd	Schema

File Name	Extension	Type
COREP taxonomy _import_all_taxonomies	.xsd	Schema

XBRL Taxonomy Designer includes these sample XBRL files for educational purposes in the `samples/xbrl` folder:

- acme
- FLIPA
- IFRS-GP (International Financial Reporting Standard - General Purpose)
- patterns
- usgaap

File Extensions

During these tutorials, you work with three types of files—XBRL schemas, XML linkbases and XBRL instance documents. The file extensions for each are:

Table 107 File Extensions

Extension	File Type
.xsd	XBRL Schema
.xml	An XBRL linkbase or XBRL instance document. Note that linkbases often include a name for the type of linkbase, for example, ca-gaap-pfs-presentation, ca-gaap-pfs-label. Instance documents can have a `.xml` or `.xbrl` file extension.
.xbrl	Instance document

Launching XBRL Taxonomy Designer

To begin, launch XBRL Taxonomy Designer.

➤ To launch XBRL Taxonomy Designer, click **Start**, then **All Programs**, and then **Disclosure Management XBRL Taxonomy Designer**.

(This assumes the product was installed in the default location.) The XBRL Taxonomy Designer application opens.

Note: Menu choices differ for creating and editing taxonomies versus creating and editing instance documents.

Shortcuts

Table 108 lists the general shortcut keys for XBRL Taxonomy Designer.

Table 108 General Shortcut Keys

Shortcut Key	Description
F1	Opens the XBRL Taxonomy Designer Online Help menu.
Ctrl+Z	Undoes the last action taken by XBRL Taxonomy Designer.
Ctrl+I	Inserts a sibling element.
Ctrl+N	Inserts a child element
Ctrl+E	Edit the selected cell
Ctrl+X	Cut elements and relationships
Ctrl+D	Delete elements and relationships.
Ctrl+P	Open the print dialog box.
Ctrl+S	Save or Save As the document.
Ctrl+O	View the Options menu.
ALT+F4	Exit XBRL Taxonomy Designer

Basic XBRL Taxonomy Designer Tutorial

The basic XBRL Taxonomy Designer tutorial is divided into these sections:

- "Taxonomy Basics" on page 238 teaches about the opening, editing, creation, and editing of taxonomies.
- "Elements Basics" on page 249covers how to create and edit elements within a taxonomy.
- "Relationships Basics" on page 257shows how concepts relate to each other and how to create and edit these relationships.
- "Reference Taxonomy Basics" on page 261 reviews how to view, load, and remove a reference taxonomy.
- "Instance Documents Basics" on page 266 covers the opening, creation, and editing of instance documents.

Taxonomy Basics

This tutorial walks you through the basics of using the XBRL Taxonomy Designer Taxonomy Editor.

- "Lesson One: Opening a Taxonomy" on page 239 teaches how to open a taxonomy in XBRL Taxonomy Designer.
- "Lesson Two: Creating a Taxonomy" on page 243 explores how to create a taxonomy.

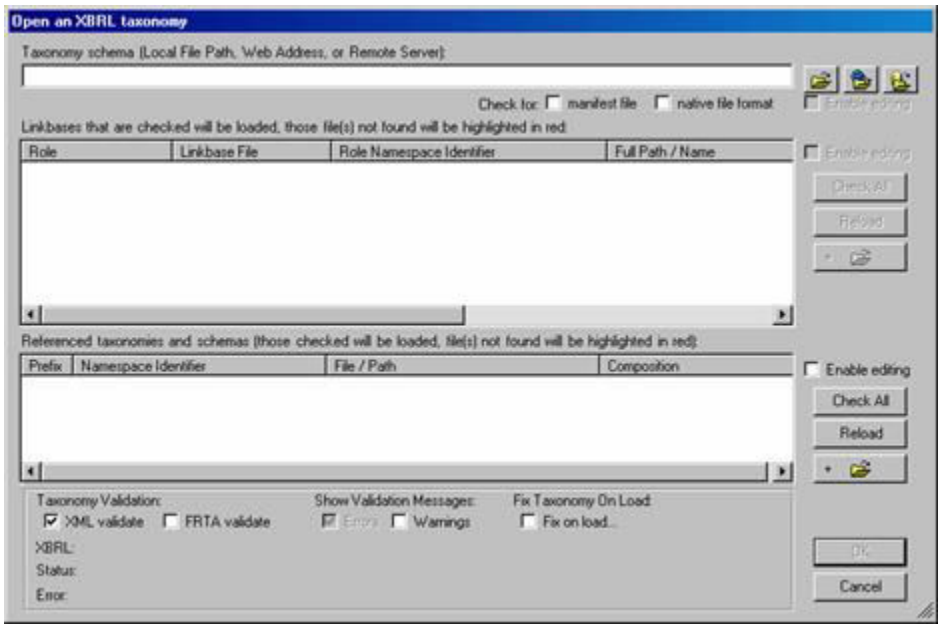
- "Lesson Three: Taxonomy Identification and Description" on page 244 explains how to create namespaces for taxonomies.
- "Lesson Four: Saving a Taxonomy" on page 245 teaches how to save a taxonomy.
- "Lesson Five: Validating a Taxonomy" on page 247 teaches how to validate a taxonomy.
- "Lesson Six: Printing a Taxonomy" on page 247 covers printing a taxonomy.

Lesson One: Opening a Taxonomy

➤ To open a taxonomy from your local system:

1 **From within the Taxonomy Editor, select File, then New, and then Taxonomy, or use the drop-down icons.**

The Open an XBRL Taxonomy window is displayed.



Three icons in the top right of the window enable you to open taxonomies.

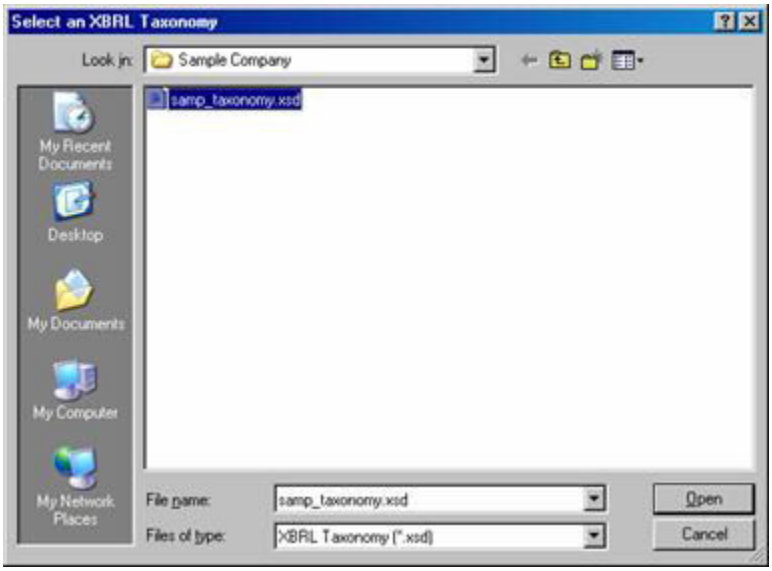
-  enables you to browse to any accessible location on your computer and select the appropriate taxonomy file.
-  enables you to access taxonomies that are stored in a Taxonomy Directory in a user-defined Web location.
-  enables you to access taxonomies that are stored in a local Taxonomy Directory.

2 **Click the on the  icon to open the Select an XBRL taxonomy browse window and navigate to the location of a taxonomy file.**

For this example, navigate to the folder in which the XBRL Taxonomy Designer application was installed, and open the `Samples` folder. For a typical default installation of XBRL Taxonomy Designer follow the file path:

```
C:/Programs Files/Disclosure Management/sclosure Management XBRL Taxonomy Designer,
XBRL Professional Studio (version)/ Samples,/ Tutorial/Sample Company
```

3 **Locate the `samp_taxonomy.xsd` file from the folder.**

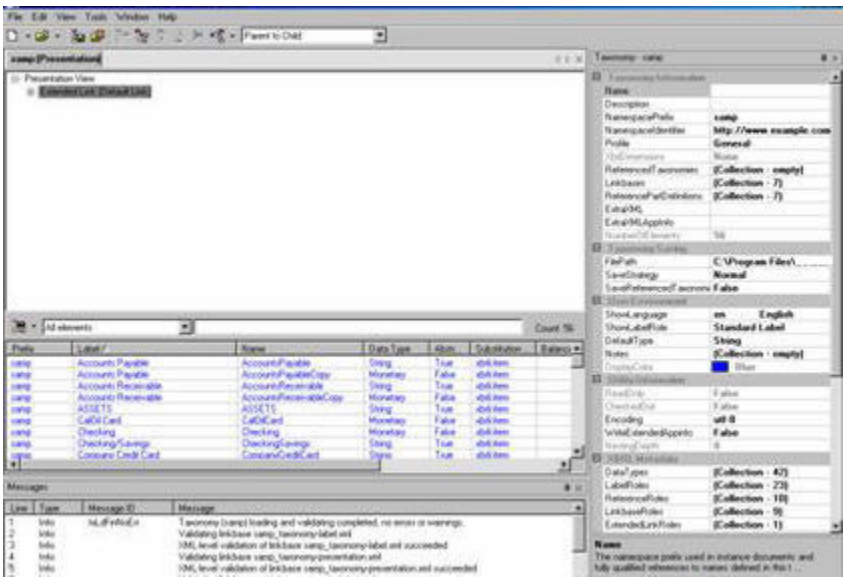


4 **Double-click the `samp_taxonomy.xsd` file or select it and click Open.**

The browse window closes. Linkbase information for the taxonomy displays in the Open Taxonomy window. Before you load the load taxonomy you may select Enable Editing . In this case do not. After the taxonomy is loaded you turn on editing using the File menu.

5 **Click OK at the bottom of the window to load the selected taxonomy and close the Open Taxonomy window.**

The Taxonomy Editor screen is displayed with the selected taxonomy loaded.



6 Select **View**, and then **Expand** to see the complete presentation view of the taxonomy.

7 Select **File**, and then **Close** to close this taxonomy.

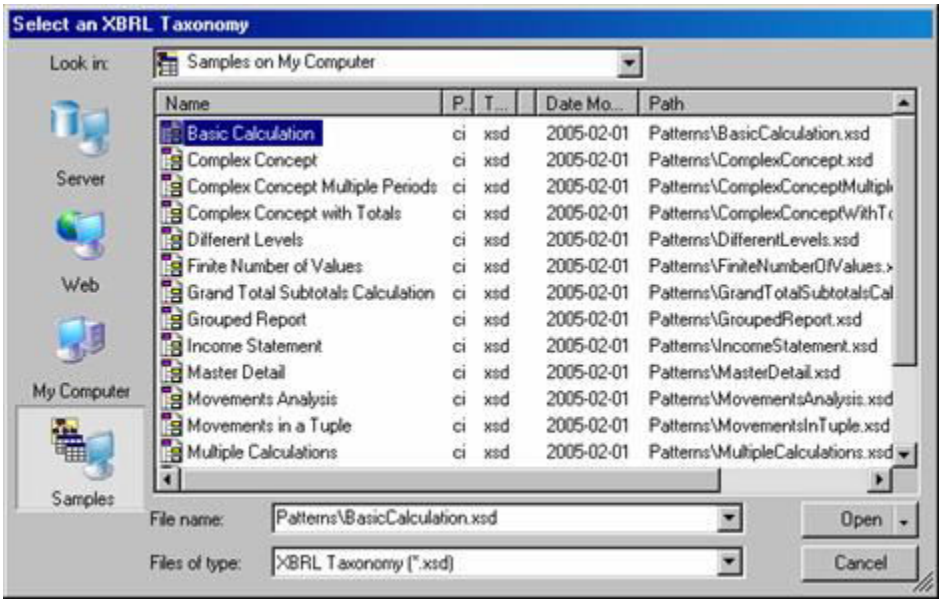
➤ To open a taxonomy from local taxonomy directory:

1 From the XBRL Taxonomy Designer main screen, select **File**, then Open, and then **Taxonomy**.

The Open Taxonomy window is displayed.

2 Click  .

The local Taxonomy Directory opens in another window, listing the taxonomy files stored within it.



Note: Taxonomy Directories are XML files that display pertinent and descriptive information about the taxonomies listed. The URL in the Look in field can be changed so that any Taxonomy Directory using the prescribed format can be accessed. For more on Taxonomy Directories, see the section Taxonomy Directories in the Taxonomy Builder chapter of this guide.

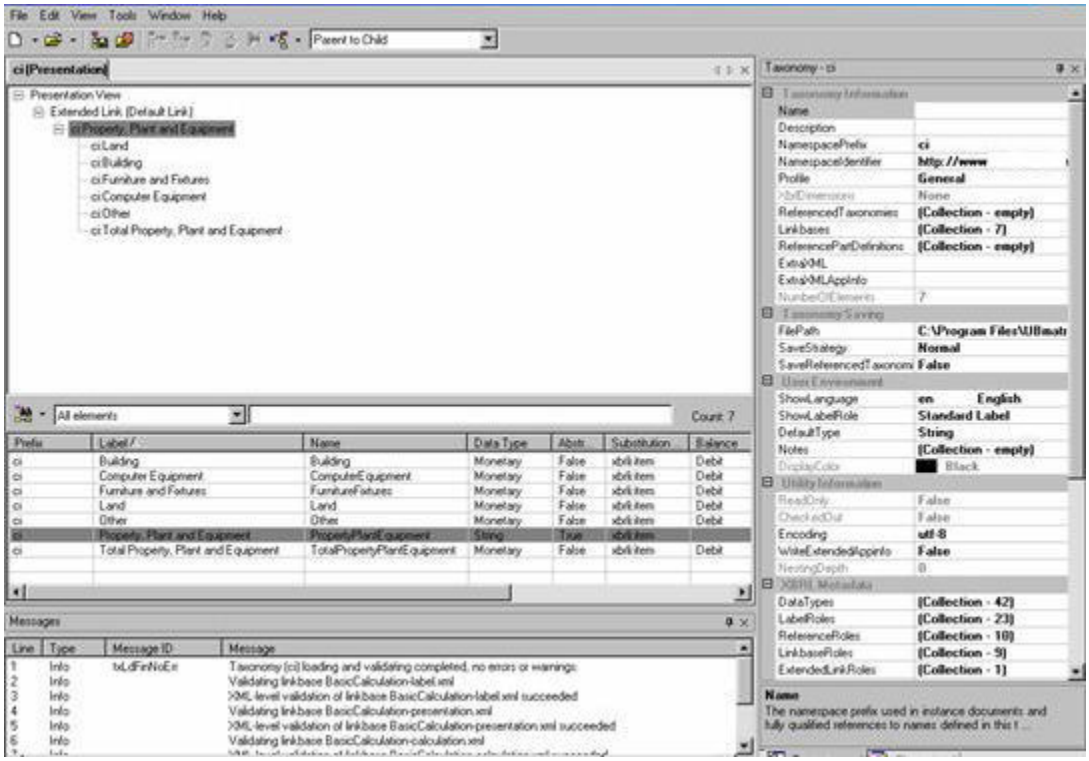
3 **Select the Basic Calculation file, and click Open.**

The Select an XBRL taxonomy window closes and Linkbase information for the taxonomy displays in the Open Taxonomy window.

4 **Click OK at the bottom of the window to load the taxonomy from the Internet into the XBRL Taxonomy Designer application.**

Do not close the taxonomy because we use it in the next step.

The Open Taxonomy window closes and the Taxonomy Editor screen is displayed, ready for edit.



You completed Taxonomy Basics Lesson One. Proceed to "Lesson Two: Creating a Taxonomy" on page 243.

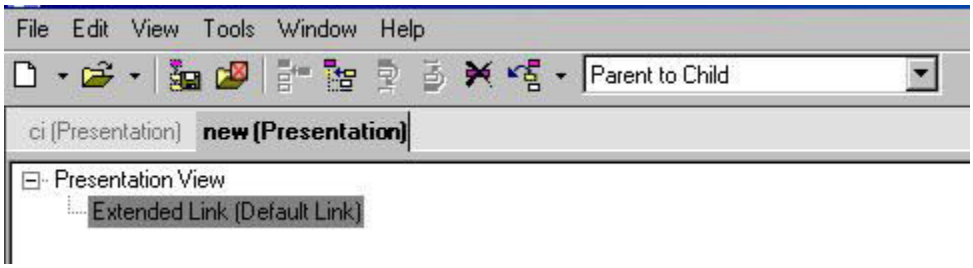
Lesson Two: Creating a Taxonomy

➤ To create a taxonomy:

1 **In the Taxonomy Editor, select File, then New, and then Taxonomy.**

A blank Taxonomy Editor screen is displayed.

Because we did not close the previous taxonomy, it now is displayed as an additional tab at the top of the Relationships Tree View pane, allowing you to work with the taxonomies simultaneously.



For the purposes of this tutorial, let's close the first taxonomy for now. To do this:

2 **Click the first tab to select ci(Presentation).**

3 **Select File, and then Close.**

Keep the taxonomy open for now; save it in the next section.

Before you save the taxonomy, we must complete a few steps in the next lesson. Creating a taxonomy provides us with a blank taxonomy document and we must provide some additional identification for this taxonomy besides its file name.

You completed Taxonomy Basics Lesson Two. Proceed to "Lesson Three: Taxonomy Identification and Description" on page 244.

Lesson Three: Taxonomy Identification and Description

We must provide the identification and descriptive information for the taxonomy in the taxonomy properties grid, on the upper right hand side of the XBRL Taxonomy Designer screen.

➤ To enter the descriptive information:

1 **Optional: Enter a name.**

The name which, if provided, is stored in the local taxonomy directory, to provide a user-friendly name for the taxonomy.

For example: International Financial Reporting Standards General Purpose (IFRS-GP)

2 **Optional: Enter a description.**

The description if provided, is stored in the taxonomy native file and local taxonomy directory, to provide a user-friendly description of the taxonomy

For example: This Taxonomy is intended to allow Profit-Oriented entities to prepare XBRL-based interim and annual financial statements according to IFRS. This includes, but is not limited to, consolidated publicly listed entities, parent entity financial statements, and non-consolidated entities.

3 **Enter the namespace prefix.**

The required namespace prefix is used to distinguish base and reference taxonomies. Use all lowercase letters.

For example: ifrs-gp

4 **Enter the namespace identifier.**

The required namespace identifier, also known as the target namespace, uniquely identifies this taxonomy. By convention this is a phantom URL (that is, no Web location need exist, but the character string must be the same wherever this taxonomy is referenced).

For example: http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15

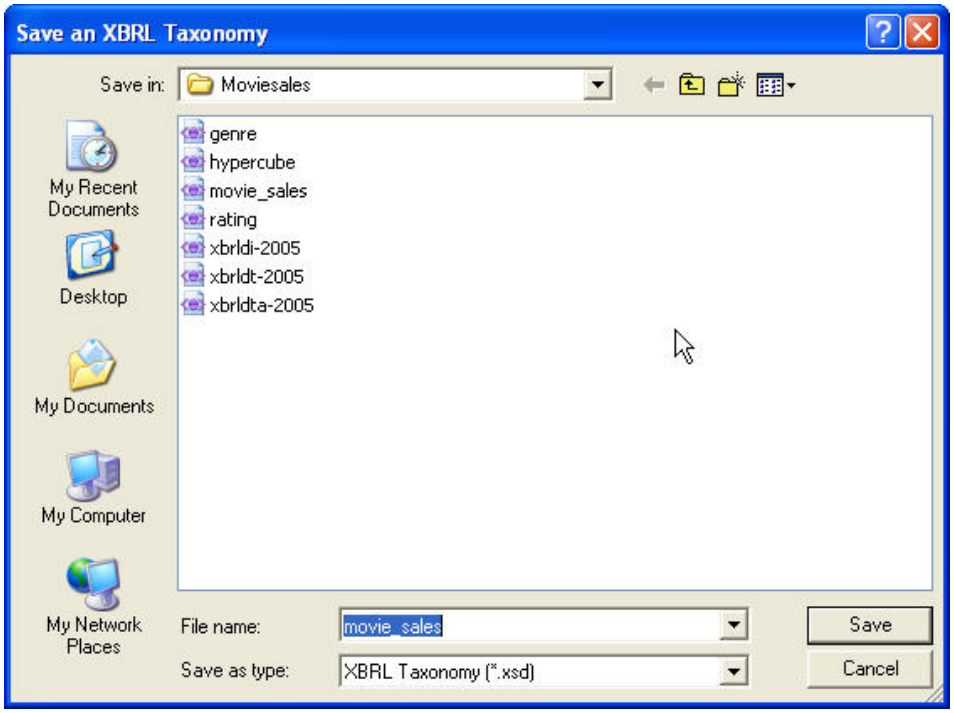
You completed Taxonomy Basics Lesson Three. Proceed to "Lesson Four: Saving a Taxonomy" on page 245.

Lesson Four: Saving a Taxonomy

➤ To save a taxonomy:

1 Enter a **namespace prefix** and **namespace identifier** (see Lesson Three).

2 Select **File**, and then **Save**.



3 In the Save dialog box, change the file name to `MyTaxonomy.xsd`.

4 Click **Save**.

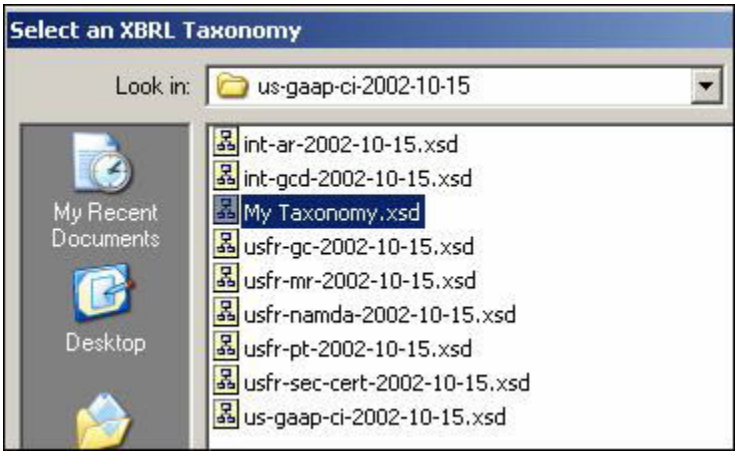
5 Select **File**, and then **Close** to close your taxonomy.

To confirm that your taxonomy saved as directed:

6 Select **File**, then **Open**, and then **Taxonomy**.

7 Click  .

The Select an XBRL Taxonomy window is displayed.



8 **Select File, and then Close.**

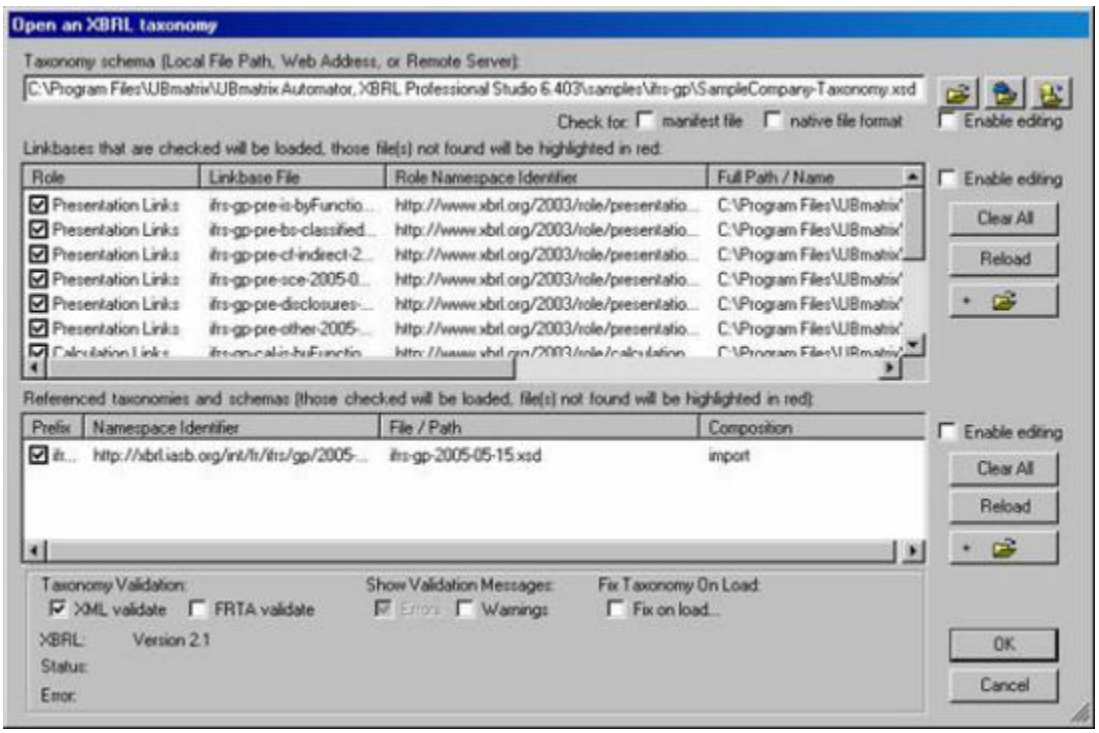
➤ To prepare for the next section of the tutorial:

1 **Select the `SampleCompany-Taxonomy.xsd` using this path and click Open.**

```
C:/Programs Files/Disclosure Management/Disclosure Management XBRLTaxonomy Designer/
XBRL Professional Studio (version)/ Samples/ifrs-gp/ SampleCompany-Taxonomy.xsd
```

2 **In the Open Taxonomy window, click OK to load the taxonomy.**

Note the number of linkbases that are checked and loaded. More discussion follows about the types of linkbases. Also note the referenced linkbase in the second window.



You completed Taxonomy Basics Lesson Four. Proceed to "Lesson Five: Validating a Taxonomy" on page 247.

Lesson Five: Validating a Taxonomy

Note: If you did not load the `SampleCompany-Taxonomy.xsd` file, do so.

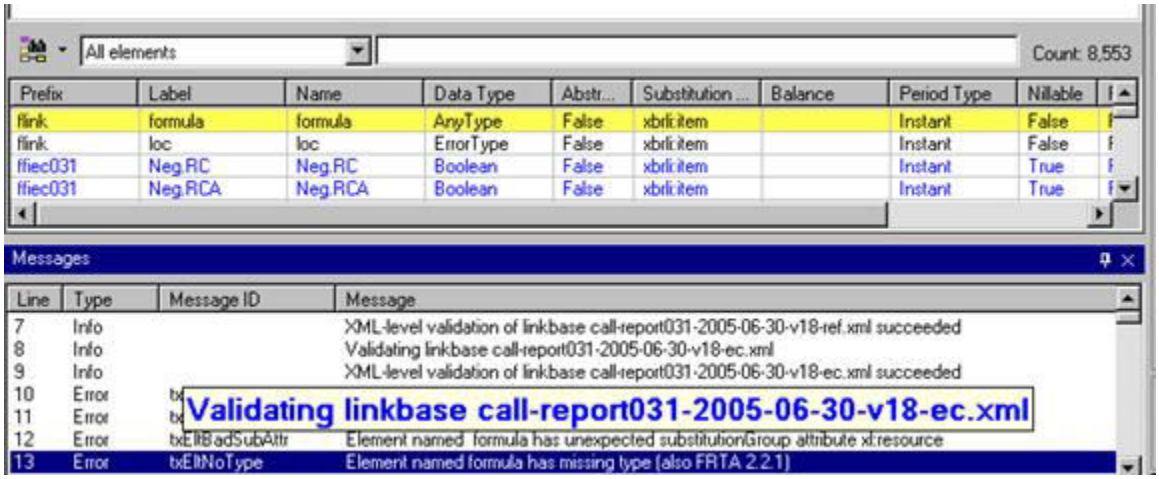
➤ To validate a taxonomy:

1 **Select Tools, and then Validate Taxonomy.**

The Taxonomy Validator window is displayed. XML Schema Validation and XBRL Validation is selected by default. You can select and/or deselect as appropriate. We use the default settings.

2 **Click Run Validation.**

When complete, the status and details of the validation are listed in the Messages pane on the bottom of the screen. If an error with a concept is shown, double-click it and the concept in question are highlighted in yellow.



For detailed information on taxonomy validation, see "Validating Taxonomies" on page 51.

You completed Taxonomy Basics Lesson Five. Proceed to "Lesson Six: Printing a Taxonomy" on page 247.

Lesson Six: Printing a Taxonomy

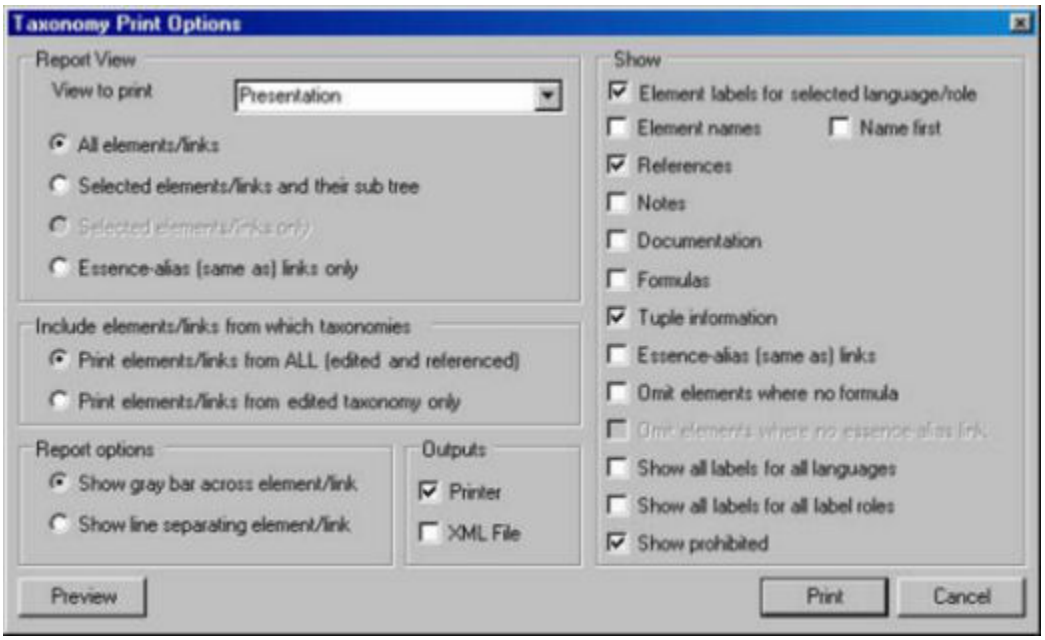
➤ To print a taxonomy:

1 **Open the** `SampleCompany-Taxonomy.xsd`.

2 **Select File, and then select Print.**

The Taxonomy Print Options window is displayed. Note the View to print option and the number of ways a taxonomy can be printed. Also note Show check boxes on the right side, including formulas.

Outputs include print to a printer or a XML file. XML file output can be later converted to any type of file, for example, PDF. This can be exchanged with other people collaboratively developing the taxonomy or it can be used for documentation purposes.



3 **Click Preview at the bottom of the window.**

The Print Preview window is displayed allowing you to see a printable report of the taxonomy. For this taxonomy there are over 100 pages of documentation.

4 Click **Close** to close the Print Preview window and return to the Taxonomy Editor screen.

You completed Taxonomy Basics Lesson Five and the Taxonomy Basics Tutorial.

Elements Basics

This tutorial walks you through the basics of elements. An XBRL element is a fact or piece of information described by an XBRL taxonomy. For example, an element with the name "Total NonInterestIncome" is the USFR-FST taxonomy's XBRL element name for the financial statement disclosure fact for "Total Noninterest Income".

- "Lesson One: Working with the Elements List" on page 249 teaches how to use the elements list.
- "Lesson Two: Editing Element Properties Using the Properties Grid" on page 254 explains how to edit an element using the Properties Grid.
- "Lesson Three: Editing Elements Attributes" on page 256 covers edited attributes of an element.
- "Lesson Four: Moving Elements in a Taxonomy" on page 256 explains how to move elements inside of a taxonomy.
- "Lesson Five: Searching for Elements" on page 256 explains how to search through the elements.
- "Lesson Six: Deleting Elements in a Taxonomy" on page 257 covers how to delete an element.

Lesson One: Working with the Elements List

Before we start to create your first taxonomy, you must know a few basics.

When you are creating a XBRL taxonomy, consider these questions. Note that this is an iterative process.

- What are you trying to model in the taxonomy?
  - Concepts on a financial disclosure?
  - Concepts in a piece of accounting literature?
- If you want to do a disclosure, such as the balance sheet, collect many samples for the industry of choice.
- Put all titles of the concepts into a spreadsheet, one company per column, and use one row for each common concept. If many companies share a concept (even if the title changes slightly), then that concept should be in your taxonomy.
- If you are working on a piece of the literature, read it carefully and copy out all nouns and noun phrases. The nouns or noun phrases that are defined or explained by the statement are the ones you want in your taxonomy.

- These approaches give you the list of elements in your taxonomy, then you can enter and assign attributes like balance attribute and period type. These can be entered in the XBRL Taxonomy Designer tool or in an Excel spreadsheet and imported in a batch process.
- Next create a sample instance document (automatically generate it with XBRL Taxonomy Designer) to validate your ideas. Then start adding labels, references, and other linkbases.
- There is a limited list of how financial information is commonly presented. The Patterns training in this course shows how to create the most common ones identified.
- As more public taxonomies reach recommendation stage from XBRL International you will likely create a taxonomy to extend them.

There are number of short cut (Ctrl) key functions that can be used when working with elements.

Table 109 Shortcut Keys for Working with Elements

Shortcut Key	Description
Ctrl+I	Inserts a sibling element.
Ctrl+N	Inserts a child element
Ctrl+E	Edits the selected cell
Ctrl+X	Cuts elements and relationships
Ctrl+D	Deletes elements and relationships.

Adding Elements

There are a number of ways to add elements to a taxonomy. We'll use every method in this exercise and you can choose which you prefer. A more advanced method for a large number of elements is to create a list in Microsoft Excel and import them. We do this later in the course.

➤ To add an element using the shortcut menu:

1 **Using what we've learned so far, create a taxonomy in the Taxonomy Editor.**

2 **Save the taxonomy with the name in this format:**

```
YYYY-MM-DD-MyOrganization-Taxonomy.xsd
```

3 **Replace the date characters with the current date and MyOrganization with the name of your organization.**

Note: Save your taxonomy often.

4 **Enter the namespace prefix and ID for this taxonomy.**

For example "dc" as the prefix for Dow Chemical and an ID like: http://www.DowChemical.com/FinRep/FixedAssets

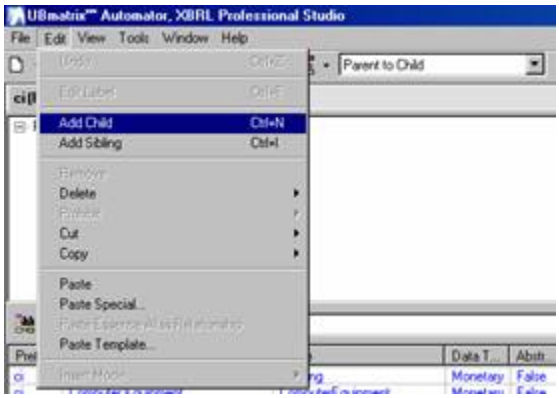
5 **In the Relationships pane, insert a child element like Property, Plant and Equipment by right-clicking on Extended Link (Default Link).**

6 To edit the element name, enter the label of the element and press **Enter**.

➤ To add an element using the edit menu:

1 In the Relationships pane, select **Property, Plant and Equipment**.

2 Select **Edit**, and then **Add Child**.



➤ To create an element using the toolbar method:

1 In the Relationships pane, select an element.

2 On the Taxonomy Editor toolbar, click the  **Add Sibling** icon.

Note that you more toolbar choices are available; for example: insert child, move element up or down, prohibit and undo.

3 Name the element (your choice from the list).

➤ To create elements using shortcut keys:

1 Select any element and press **Ctrl+N**.

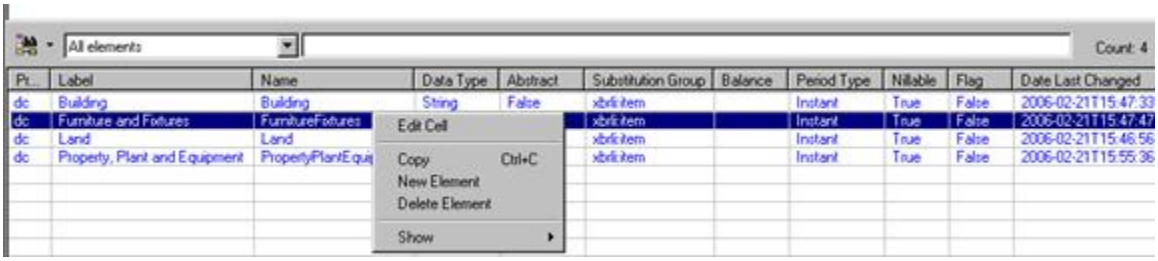
A child element is created.

2 **Select another element and press Ctrl+I.**

A sibling element is created.

➤ To add an element using the Elements List (located just below the Relationships pane).

1 **From anywhere in the Elements list, right-click and select New Element.**



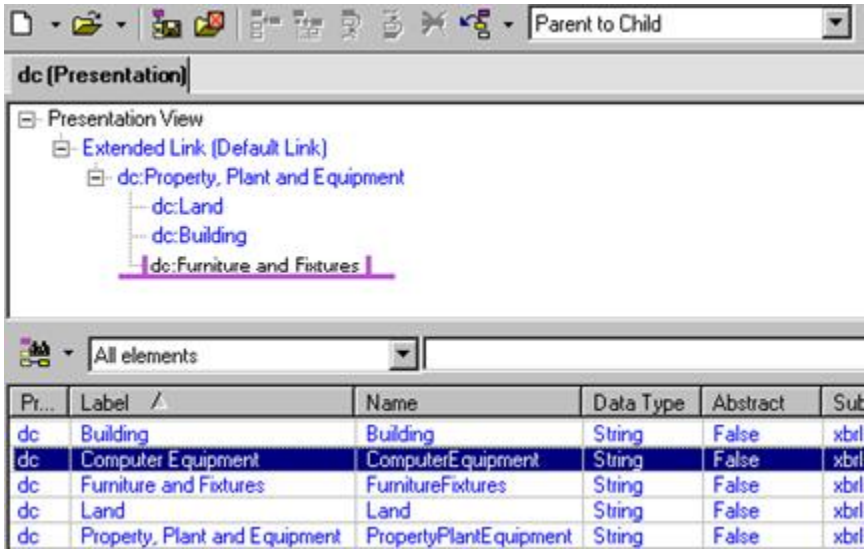
The New Element Form window is displayed for completion.

2 **Add a label, if appropriate.**

3 **When finished, click Close.**

The element is added to the Elements list.

4 **To form a relationship for the element, drag the element from the Elements list to the appropriate location on the relationship tree.**



5 **Release the left-click button to drop the element and form the relationship.**

6 **Enter all elements in the list using the method of your choice.**

Element Attributes

Some attributes for each element must be updated. Table 110 describes the attributes in the Element list.

Table 110 Element Attributes

Attribute	Description
Prefix (Namespace Prefix)	Namespace prefix of the taxonomy. This is set up when you create your taxonomy. See the section in this tutorial on Taxonomy Identification and Description.
Name	Name of the element. Elements can be Auto-Named, by automatically assigning element names based on labels. For example — CashAndCashEquivalents. Though this property can be edited, if automatic naming options are active for the element's taxonomy, any changes to the element's default label (for taxonomy's shown language and shown role) are reflected by automatic generation of a name.
Labels	Collection of labels of the element. The default label is shown here. For example — Cash and Cash Equivalents for the English label and the "Standard" label role. An ellipsis button activates the Labels Collection Editor. Alternatively, a plus box expands the labels collection in the property grid for direct editing of labels, and for alternative entry of labels. If a label's text is entered and role chosen, and a subsequent label is appropriate, double-click the plus/minus expansion box of the Add new property.

Attribute	Description
Data Type	Data type of the selected element. For example, Monetary. Changing the data type redisplays the property grid with only applicable properties for that data type. Data types include: ● Decimal ● Integer ● Monetary ● Shares- stock ● String- pure text ● Boolean ● URI ● Dates ● Token ● Tuple
Abstract	This is a True or False selection. True is used for those elements that are containers like Assets with no fact value, a reportable value associated with it. All other elements with values are children of these elements.
Substitution Group	The substitution group attribute is defined on every element. By default, it must be an item or a tuple. It can also hold the value of other elements, which are derived from items or tuples. An item is the default substitution group for a concept that is reported in an instance document with a value. A tuple is used to group of items in an instance document.
Balance Type	Balance Type is a property of monetary elements. Options allowed for Balance Type: Unspecified (blank), Debit, and Credit. Selecting Debit or Credit reflects the most common state for an element, for example, a revenue concept is often a debit balance.
Period Type	Period type is a required value for all but tuple elements in V2.1 taxonomies. The value for period type can be “Instant” or “Duration”. Instant is most often used for balance sheet type elements and duration for profit and loss.
Nillable	Nillable is an optional attribute which can be assigned to non-abstract elements. The value of nillable is set to “False” and may be changed to “True” to indicate that in instance documents the element may appear with an attribute nil=”true” and empty content.
Flag	
Date Last Changed	The exact date and time is tagged when an element was last changed. This is useful for version control.

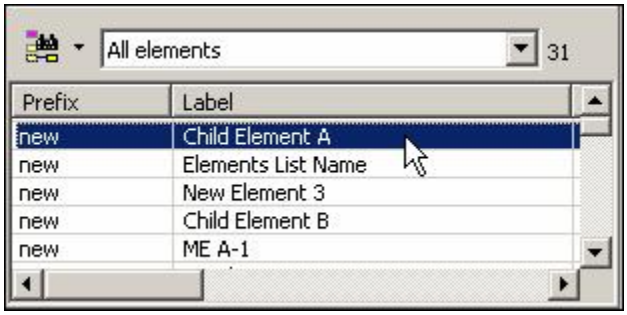
You completed Elements Basics Lesson One. Proceed to “Lesson Two: Editing Element Properties Using the Properties Grid” on page 254.

Lesson Two: Editing Element Properties Using the Properties Grid

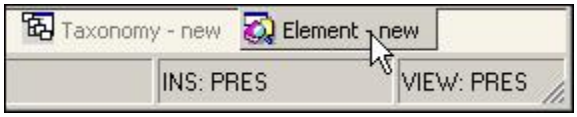
➤ To edit an element property:

1 **Select an element from the Relationships Pane or the Element List.**

For this example, select Child Element A from the Elements List. (If this element no longer exists in your taxonomy, select any element.)

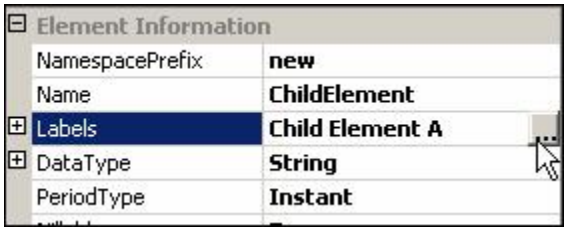


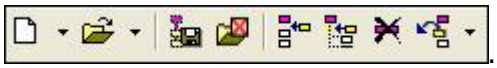
2 **Click the Element – new tab at the bottom of the Properties Grid.**



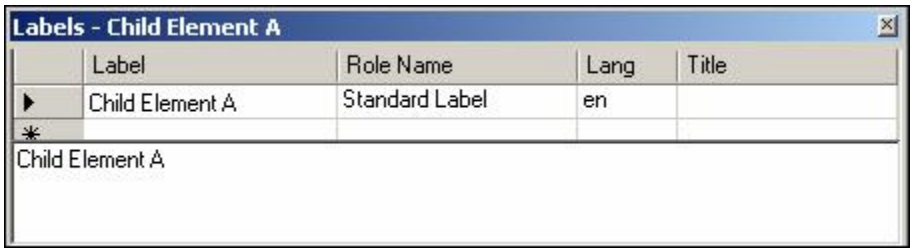
3 **Select Label as the property to edit.**

An ellipsis button is displayed in the associated field.



4 **Click**  .

The Referenced Taxonomies – new window is displayed.



5 **In the Label column, select Child Element A.**

6 **Enter a label and click X to close the window.**

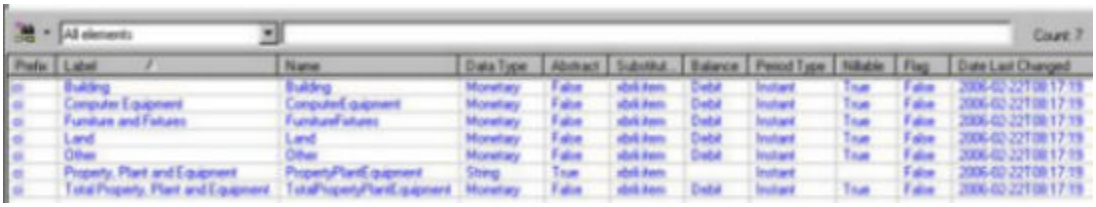
Note how the label for the element has changed in the Relationships pane, the Elements List and the Element Properties grid.

You completed Elements Basics Lesson Two. Proceed to "Lesson Three: Editing Elements Attributes" on page 256.

Lesson Three: Editing Elements Attributes

➤ To edit the attribute of an element in the Element List:

1 In the Element List, right-click a cell and select **Edit Cell**.



2 Using the copy of the screen above, edit the **Data Type**, **Abstract**, **Balance**, **Period Type**, and **Nillable** attributes for a few elements.

3 Make your edits and click **Enter**.

4 Select **File**, then **Save**, and then **Close**.

You completed Elements Basics Lesson Three. Proceed to "Lesson Four: Moving Elements in a Taxonomy" on page 256.

Lesson Four: Moving Elements in a Taxonomy

➤ To move elements within the Relationships pane:

1 Select an element in the taxonomy tree view.

2 Press and hold the left mouse button.

Notice how the cursor changes as you move over elements of the taxonomy tree. Also note how the status bar at the bottom of the screen describes the action which would be performed if the left mouse button were released at that point.

3 Release the left-click button to move the element.

➤ To move elements from within the Elements List into the Relationships pane:

1 Select an element in the Element List.

2 Press and hold the left-click button and drag the element to the Relationships pane.

3 Drop (release the left-click button) the element within the taxonomy tree.

You completed Elements Basics Lesson Four. "Lesson Five: Searching for Elements" on page 256.

Lesson Five: Searching for Elements

The Elements List view can be filtered or searched. This becomes helpful when you are working with taxonomies with thousands of elements. The element list view has several filter and search options to help you locate elements.

➤ To search for elements, use one method:

- In the Element List pane, click the drop-down arrow associated with the **Find and Show Elements** icon to display the list of find and show choices.

  You can choose criteria to Filter the dictionary view of the taxonomy elements.

- Use the ALL Elements drop-down menu to search for elements.

You completed Elements Basics Lesson Five. Proceed to "Lesson Six: Deleting Elements in a Taxonomy" on page 257.

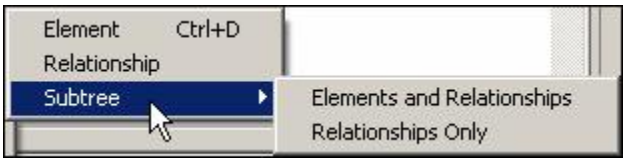
Lesson Six: Deleting Elements in a Taxonomy

➤ To delete elements in a taxonomy:

1. **Load the SampleCompany-Taxonomy.xsd file and enable editing.**
2. **Select the upper most taxonomy element of the tree branch (below the Extended Link).**
3. **Right-click and select Delete.**

   Three options are provided:

   - **Element**—Deleting an element deletes the element from the dictionary (Element List Pane), its labels, references, formulas, and all relationships to and from that element.
   - **Relationship**—Deleting a relationship deletes only the selected relationship.
   - **Subtree**—Deleting a subtree displays a submenu which enables you to specify deleting the subtree and all relationships and elements, or delete only the relationships. If the element has no subtree, this option is not available.



You completed the Element Basics Tutorial.

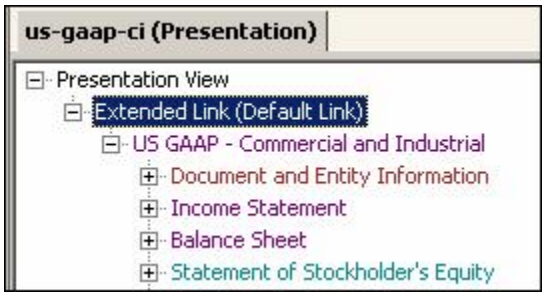
Relationships Basics

This tutorial walks you through the basics of relationships between elements.

Lesson One: Working with Relationships

Many times you want to view two portions of a taxonomy simultaneously. Or, you may want to view the same taxonomy elements using different perspectives, such as Presentation and Calculation. In XBRL Taxonomy Designer, you can add and change relationship views and perspectives using the Relationships pane.

In the Relationships pane of the Taxonomy Editor, the perspective of the displayed view is noted on the taxonomy tab at the top of the pane.



Taxonomies can be viewed from three perspectives or views— Presentation, Calculation, and Definition.

You completed Relationships Basics Lesson One. Proceed to "Lesson Two: Changing the Relationship View" on page 258.

Lesson Two: Changing the Relationship View

➤ To change the taxonomy view:

1 **Load the EU bank capital adequacy taxonomy COREP taxonomy by loading the file `_import_all_taxonomies` in the Taxonomy Editor screen.**

2 **Right-click in the Relationships pane.**

A shortcut menu is displayed.

3 **Select View.**

Three perspectives are listed. A checkmark indicates the views currently displayed.

4 **Select a view that is not selected.**

For this example, select Calculation. Note how the perspective of the taxonomy changes in the Relationships pane.

5 **Click to expand the Extended Link for the Calculation view of the taxonomy.**

You completed Relationships Basics Lesson Two. Proceed to "Lesson Three: Adding Views to the Relationships pane" on page 259.

Lesson Three: Adding Views to the Relationships pane

➤ To view the taxonomy from two or three perspectives simultaneously:

1 **Right-click in the Relationships pane.**

2 **From the shortcut menu, select Show, and then Add Another Relationships pane.**

Note that in this submenu, no checkmarks are associated with the displayed perspectives. This is because when you add multiple views, you may want to add multiple views of the same perspective.

For this example, select **Presentation**.

A second pane is created and displays in Presentation view. Notice that two relationship views are now visible.

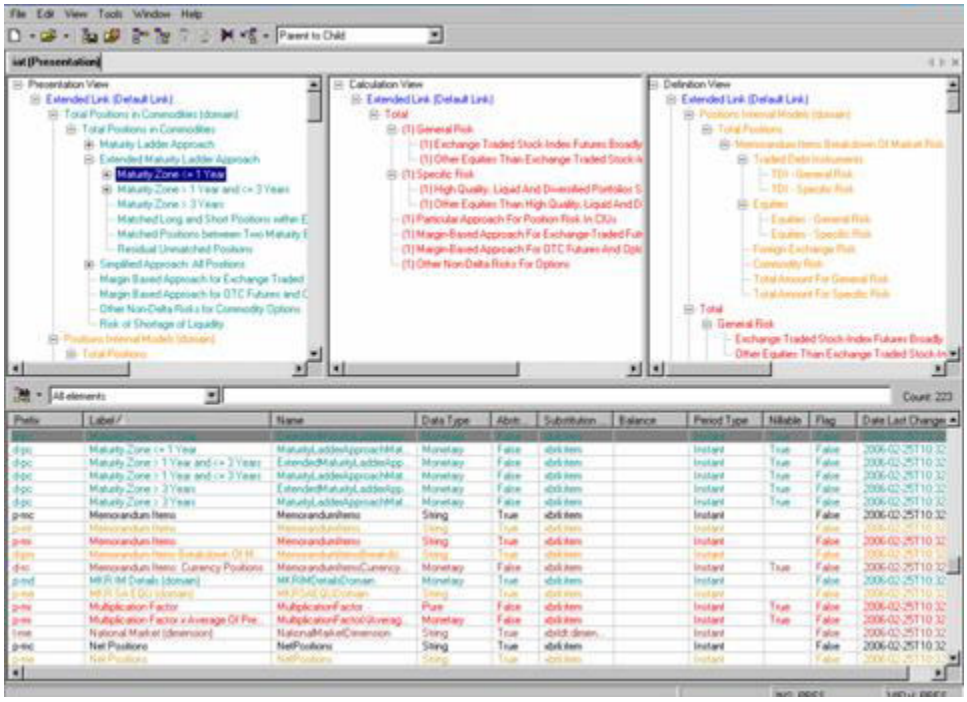
➤ To add another view using the View menu:

1 **Select View, and then Show.**

Note that the menu options are similar to those in the shortcut menu.

2 **Select Add Another Relationships Pane, and then Definition.**

Three views are now displayed in the Relationships pane.



You completed Relationships Basics Lesson Three. Proceed to "Lesson Four: Removing a View from the Relationships Pane" on page 260.

Lesson Four: Removing a View from the Relationships Pane

➤ To remove a view from the Relationships pane:

1 **Select the view to remove by clicking anywhere within it.**

Note that the active view is indicated on the bottom of the Taxonomy Editor screen on the far right of the status bar.

2 **Use the View menu or the shortcut menu to select Show, and then Remove Relationships Pane.**

Using the Sync Feature

When working with a taxonomy, you may want to examine elements of that taxonomy from multiple perspectives (multiple views) simultaneously. Instead of scrolling through each view to find the element, you can use the XBRL Taxonomy Designer sync feature to automate this process.

➤ To view data using multiple views:

1 **Make sure you have at least two views of the taxonomy in the Relationships pane.**

2 **Select a view by clicking it.**

3 **Use the shortcut menu or the View menu to open the Show menu.**

4 **In the Show menu, select Sync to Other Panes.**

5 Repeat step 2 through step 4 for all views.

6 From the Elements List (subjacent to the Relationships pane), scroll to locate any element, and then click to select it.

Note how all Relationship views refresh to display the selected element within their view.

7 Click an element (branch) in one of the Relationship views.

If that element exists in the other views, the other views sync up to present the selected element as well.

Disabling the Sync Feature

➤ To disable the syncing feature:

1 In the Relationship view, right-click and select **Show**, and then select **Sync to Other Panes** to deselect it.

Note that selecting elements in the Relationship view no longer affects the other views.

2 Select **File**, and then **Close** to close the taxonomy.

Reference Taxonomy Basics

This tutorial walks you through the basics of reference taxonomies.

Lesson One: Referenced Taxonomies

XBRL Taxonomy Designer provides full support for working with an unlimited depth of referenced taxonomies. Referenced taxonomies are often public or standard private taxonomies that can be used by another taxonomy that extends the structure and resources provided. The taxonomy using the referenced taxonomy is often called an "extended taxonomy." A referenced taxonomy is often called a base taxonomy. By extending a base taxonomy a company can leverage thousands of hours of collaborative work and the value of using a standard.

An extended taxonomy may reference a number of other taxonomies. The extended taxonomy combined with its referenced taxonomies is referred to as the DTS.

➤ To edit a taxonomy property:

1 Create a taxonomy and add a namespace prefix and ID.

2 Add a referenced taxonomy to this taxonomy: Select the **Referenced Taxonomies** property.

An ellipsis button is displayed in the associated field.

3 Click  .

4 In the Referenced Taxonomies - New window, click **Add**.

5 Select Referenced Taxonomies.

The Open Taxonomy window is displayed.

6 Use what you've learned to select and load a referenced taxonomy.

The namespace prefix in the tree below reflects the namespace prefix you just entered and the namespace prefix of the referenced taxonomy you loaded.

Note that the referenced taxonomy is now shown in the tree view pane of the Referenced Taxonomies.

7 Click the **X** to close the window.

In the Taxonomy Properties grid, the Referenced Taxonomy property indicates that there is one referenced taxonomy in the collection.

8 Select **File**, and then **Save**.

9 Click the **X** in the upper right of the Relationships View pane to close the taxonomy.

You completed Reference taxonomy Lesson One. Proceed to "Lesson Two: Viewing Referenced Taxonomies" on page 262.

Lesson Two: Viewing Referenced Taxonomies

➤ To view a large taxonomy with multiple reference taxonomies:

1 From the XBRL Taxonomy Designer main screen, select **File**, then **Open**, and then **Taxonomy**.

The Open Taxonomy window is displayed.

2 Click the on the Open icon to open the **Select an XBRL taxonomy** browse window.

3 Navigate to and select to open the `SampleCompany-Taxonomy.xsd` file from the `Samples` folder.

The browse window closes. All taxonomies that are referenced by the IFRS-GP taxonomy are listed in the lower part of the screen.

4 Click **OK** at the bottom of the window to load the selected taxonomy and close the Open Taxonomy window.

The Taxonomy Editor screen is displayed with the selected taxonomy loaded. Referenced taxonomies are color-coded. The authoring of reference taxonomy scenarios brings up some more complex issues of dealing with XBRL. Most of these issues are automatically handled by the application, enabling you to design taxonomies instead of coding XML/XBRL.

About Editing Referenced Taxonomies

To edit elements or the structure of taxonomies that are being referenced by the base taxonomy that is being edited, there are two choices:

- If one is an extender (consumer) of the referenced (standard) taxonomy, to which you have no change rights, the editor of the base taxonomy may change the structure of referenced taxonomies by prohibiting relationships between their elements in the referenced taxonomies and patch them with relations that are defined as part of the namespace of the top-level taxonomy.
- If one is the producer (author) of the referencing and referenced taxonomies, then checking the Enable editing check box to the right of the referenced taxonomies within the Open Taxonomy window enables direct editing of referenced taxonomies (instead of prohibiting relationships and overriding labels).

For the case in which one is the consumer of a standard taxonomy by an extending taxonomy, prohibit arcs block the referenced taxonomy's relationship arcs to prohibit the traversal of a relationship in referenced (standard) taxonomies. Prohibition conceptually puts a "do not traverse" sign on prohibited relationship arcs. The author is free to define a relationship between the elements in the edited top level taxonomy. Prohibited arcs show on the tree view with their labels in a strike-out font. An example of such prohibited arcs is automatically created when a user moves an element of one of the referenced taxonomies to another place on the relationship tree. You cannot move the element to any relationship view tree.

For example, assume that the author wants to move a subtree in the "Document and Entity Information" section. The Web site entry needs to be moved under the entity contact information. This is not the top-level taxonomy and you cannot change anything in this taxonomy, only prohibit the old link and create a link.

It is now clearly indicated that the Entity Web Site element is made a child of the Entity Contact element. The strikethrough on the Entity Web Site element indicates that this relationship is now prohibited and not traversed. XBRL relationship arcs for prohibition of the original and traversal of the relationship are generated by XBRL Taxonomy Designer.

You completed Reference Taxonomy Lesson Two. Proceed to "Lesson Two: Viewing Referenced Taxonomies" on page 262

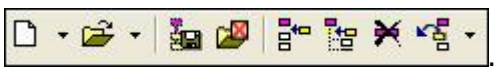
Lesson Three: Removing a Referenced Taxonomy

➤ To remove a referenced taxonomy from the base:

1 **With the base taxonomy loaded in the Taxonomy Editor, ensure that the Taxonomy tab is selected at the bottom of the Properties Grid pane.**

2 **Select the Referenced Taxonomies property.**

An ellipsis button is displayed in the associated field.

3 **Click**  .

The Referenced Taxonomies window is displayed.

4 **In the left tree view pane, select the referenced taxonomy you wish to remove.**

This activates the Remove button.

5 **Click Remove.**

The referenced taxonomy is removed from the base taxonomy. The Referenced Taxonomy window refreshes to show this.

Lesson Four: Exchanging Taxonomy Information

A taxonomy, its elements, relationships, labels, references, and formulas, can be exported for use with other tools and software. XBRL Taxonomy Designer supports export to Excel 2003 (or newer), Access, Databases, XML and CSV files. You can also import from Word.

For detailed information regarding importing and exporting taxonomies in XBRL Taxonomy Designer, see Chapter 5, "Importing and Exporting."

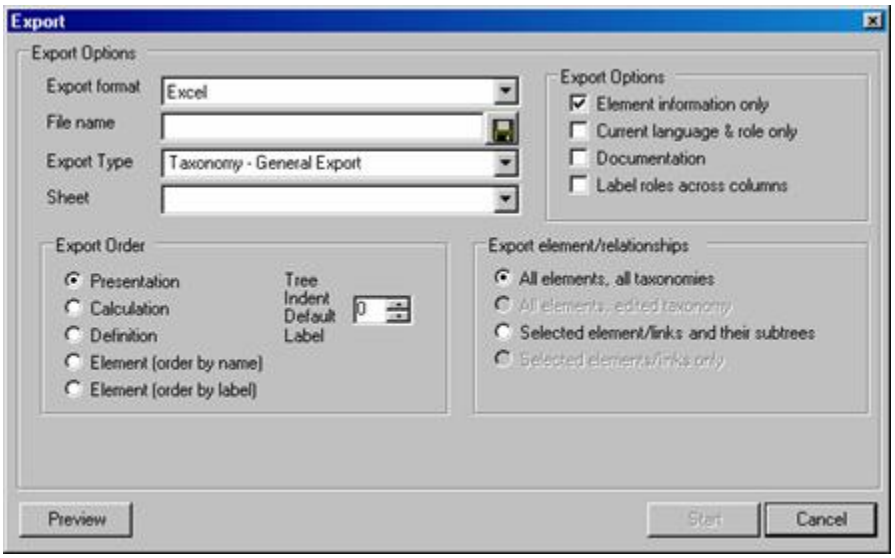
Exporting Taxonomy Information

Start by taking a taxonomy in XBRL Taxonomy Designer and exporting it to create an Excel file.

➤ To export a file to Excel:

1 **With the Basic Calculation taxonomy loaded in the Taxonomy Editor, select File, and then Export.**

The Export window is displayed.



2 **In the Export format field, select Excel.**

3 **In the File name field, enter a file name or browse to select a file to replace.**

4 **In the Export Type field, select Taxonomy – General Export.**

Keep the defaults for the rest of the Export window options.

5 Click **Preview** to view the data to be exported.

6 Click **Start** to export the data to Excel.

7 Minimize the XBRL Taxonomy Designer application and open the export file you created.

Notice it contains your taxonomy information.

8 Make at least one minor edit to this data.

9 Save and close the export file.

Re-Importing Taxonomy Information

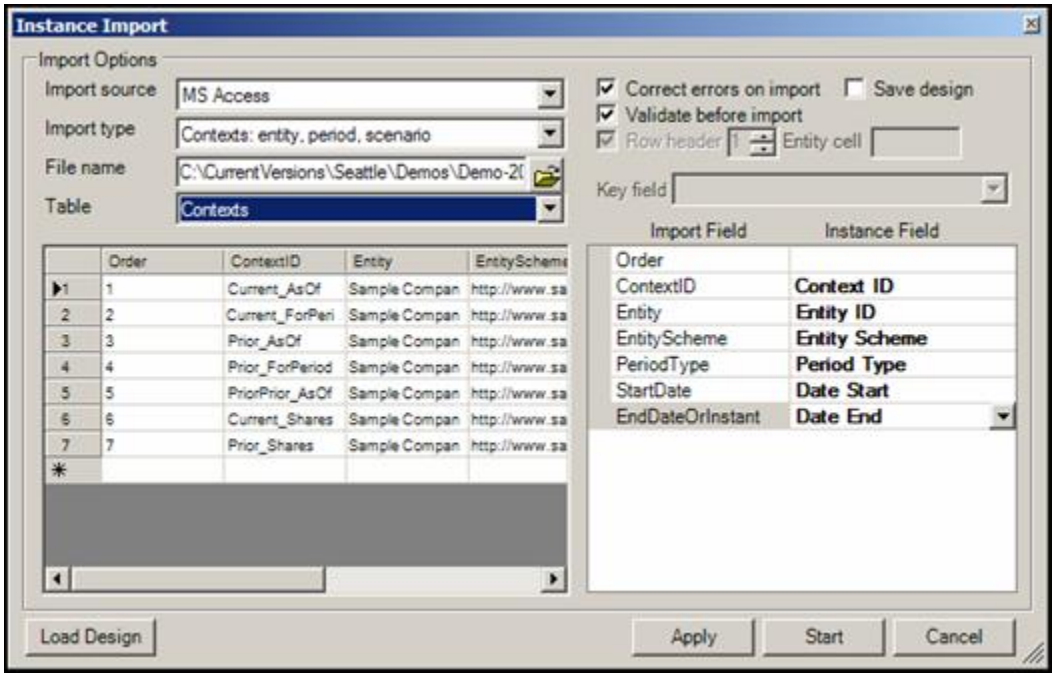
To re-import the data you just edited, we must load the exported taxonomy in your Taxonomy Editor in XBRL Taxonomy Designer.

➤ To re-import taxonomy information:

1 Maximize the XBRL Taxonomy Designer application.

2 Within the Taxonomy Editor, select **File**, and then **Import**.

The Taxonomy Import window is displayed.



3 Select **Excel** as the import source.

4 In the **File name** field, select the file that you just exported and edited.

A preview of the imported data is provided with a field name mapping.

Fields input need not necessarily match taxonomy fields in the XBRL Taxonomy Designer memory model, but if they do then they are automatically mapped to the internal field names. This flexibility enables using data sources with different field or column names, different languages, and many accommodations.

5 Click **Apply** and select **Yes** or **No** to the question.

6 Click **Start** to re-import the data with the file updates.

You completed Reference Taxonomy Lesson Four. Proceed to "Lesson Five: Creating a Calculation View (Linkbase)" on page 266.

Lesson Five: Creating a Calculation View (Linkbase)

Calculations provide one of the important benefits of XBRL—the ability to provide a consistent way of arithmetically validating data. They are effective in expressing relationships between financial reporting concepts and useful in ensuring accuracy of instance document data. XBRL calculations can handle only simple addition and subtraction and only with one context (point in time.) Calculations sometimes produce results that inconsistent from what is expected.

The use of business rules or formulas provides an additional tool to validate data (introduced later).

For this exercise, use the taxonomy you created earlier for fixed assets.

➤ To create a calculation view:

1 Load your taxonomy that had a filename format of **YYYY-MM-DD-MyOrganization-Taxonomy.xsd**.

2 Expand the view of the presentation view.

3 Use **Show Another Relationship Pane** to view the calculation view.

4 If the taxonomy loaded has a calculation linkbase, delete the relationships, not the elements.

5 Drag the **Total Property, Plant and Equipment** concept to the calculation view and when the color turns pink drop it.

   Note that you drag the total concept to the calculation view as the parent and the other concepts rolls up to it.

6 Do the same with the other elements only make them children of the Total Property, Plant and Equipment concept.

7 Save the file and validate it later.

You completed the Reference Taxonomy Tutorial.

Instance Documents Basics

This tutorial walks you through the basics of instance documents.

- "Lesson One: Creating Instance Documents" on page 267
- "Lesson Two: Validating Instance Documents" on page 268

Lesson One: Creating Instance Documents

A taxonomy becomes fruitful when it yields an instance document. The Sample instance document generator will automatically generate a template of an instance document, which can then be loaded into the instance document editor, validated, printed, formatted, The creation of instance documents utilizes a taxonomy and imports, edits, or keys in instance contexts, units, and values. Typically the design of the taxonomy will have been completed but some minor updates may be needed as the instance document is developed. XBRL Taxonomy Designer provides a convenient means to access and modify the taxonomy while authoring instance documents.

➤ To generate a sample instance document.

1 **For this example, load the SampleCompany-Taxonomy taxonomy into the Taxonomy Editor.**

This file location is in the `ifrs-gp` folder:

2 **Select a part of the taxonomy from which to generate a sample instance document.**

This could be a balance sheet subtree, or an entire Extended Link. For this example the following Balance Sheet subtreeis selected.

3 **On the Tools menu, select Sample instance generator.**

The Sample XBRL Instance Document Generator window is displayed.

4 **Specify a Save as file to generate the instance document.**

5 **In the What section choose Selected subtree.**

In this example, our sample instance document is a subtree.

6 **Select the Monetary Amount drop down menu and select $100.**

7 **Click Start to generate an instance document.**

8 **Review the instance document by selecting File, then Open, and then Instance Document.**

9 **Use the Open Instance Document window to browse and locate your newly-created instance document xml file. Double-click on the file, or select and click Open.**

10 **Click OK in the Open window to load the instance document.**

11 **Check the fact values and other collections. You may also examine the instance document by opening it with Internet Explorer, in raw XML form.**

➤ To create an instance document:

1 **On the File menu, select New Instance.**

A blank Instance Document Editor screen is displayed. The top pane is a tree view of the instance, the bottom is a data grid of the fact values, and the right side has the instance properties grid.

2 **Load the taxonomy by selecting the File menu and selecting Load a Taxonomy.**

The Open Taxonomy window is displayed.

For this example, select the BasicCalculation.xsd taxonomy. After loading, the taxonomy an additional pane is added to the screen —the Taxonomy Finder Element pane — displaying the taxonomy on the lower right of the screen.

We are now ready to enter fact values for the instance document. This is done in the Fact Values Data grid. As they are entered, contexts, entities, periods, and units are also entered using choice lists and collection editors.

3 **Click on the first blank line of the Fact Values Data grid.**

A drop-down button is displayed.

4 **Click to expand the drop-down list and select the element, Land. It populates the Element Name column.**

Note that this field has an incremental-search capability. As you start to type the prefix part of the name, the drop-down scrolls to that element and the whole prefix first matching is highlighted in the field. You can right-arrow past the colon and start typing the initial characters of the name.

5 **Click the Tab key to move to the Context column or click within the field.**

6 **Select Add/Choose.**

The Contexts window is displayed.

7 **Click the New button for a new record.**

You can click on the entity and period fields to enter them (by a chooser-editor which allows using an existing or new entry).

8 **When done, click the Okay Save button.**

Now this fact value has the newly-entered context. Also, the context value will be in the combo box drop-down list for the next record.

9 **The next column (needed for numeric entries, such as for Land value) is Unit. Select the collection editor and choose USD (U.S. Dollars) as its measure.**

10 **In the Value column, enter any number (123,000, for example).**

11 **Try another line, but if you wish to re-use the context and unit, they can be selected from the initial drop-down lists instead of activating the chooser-editor for the collection.**

12 **Save your instance document file by selecting File à Save or File à Save as.**

The result of entering the fact values is displayed in the Instance tree view pane where each entered fact value will be displayed in the view order. To change the view order:

13 **Click on the View, then Tree order, and then select an order to view the instance fact values.**

Lesson Two: Validating Instance Documents

After you create an instance document, receive a file containing an instance document, or import instance data from another application, you can make sure there are no errors in the documents you create or receive by performing a validation. The data in the instance document is validated to ensure it "foots and cross-castes" relative to the calculation links expressed in a taxonomy. For example, if Total Property and Equipment must equal Land plus Building plus Fixtures within a context, then the instance document is tested to ensure these values are accurate.

Importing Tutorial

This section explains how to import XBRL Taxonomies and Instance Documents.

Importing Tutorial Overview

These lessons teach you about Importing data to create XBRL Taxonomies and Instance Documents:

- "Lesson One: Importing to Create a Taxonomy" on page 269 teaches how to import data from a Microsoft Excel spreadsheet to create a taxonomy.
- "Lesson Two: Importing to Create an Instance" on page 273 explains how to import data from a Microsoft Excel spreadsheet to create a taxonomy.

Lesson One: Importing to Create a Taxonomy

➤ To create a taxonomy:

1 **Create a taxonomy.**

Next, we must define the Namespaces of the taxonomy

2 **In the Taxonomy Property Grid, set the NamespacePrefix to samp and NamespaceIdentifier to http://www.example.com/samp.**

3 **Select File, and then Save As to save the taxonomy.**

Note: For ease of use, we recommend that you save the taxonomy in the `samples` folder as `Sample Taxonomy`.

4 **Select File As File, then Close to close the taxonomy.**

This file is used in later lessons, so keep track of where you save it.

➤ To import indented labels to automatically create linkbases:

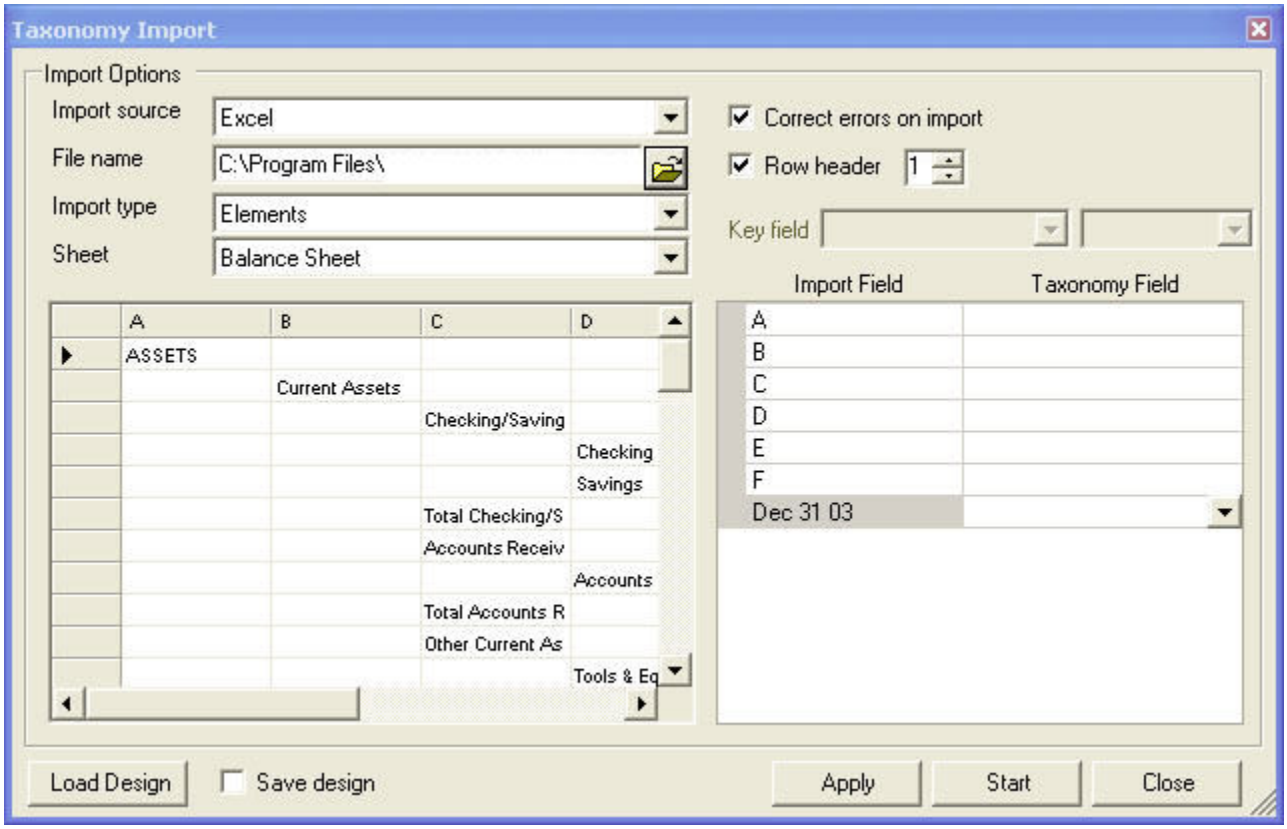
1 **Begin the taxonomy import by selecting File, and then Import.**

This opens the Import Taxonomy window.

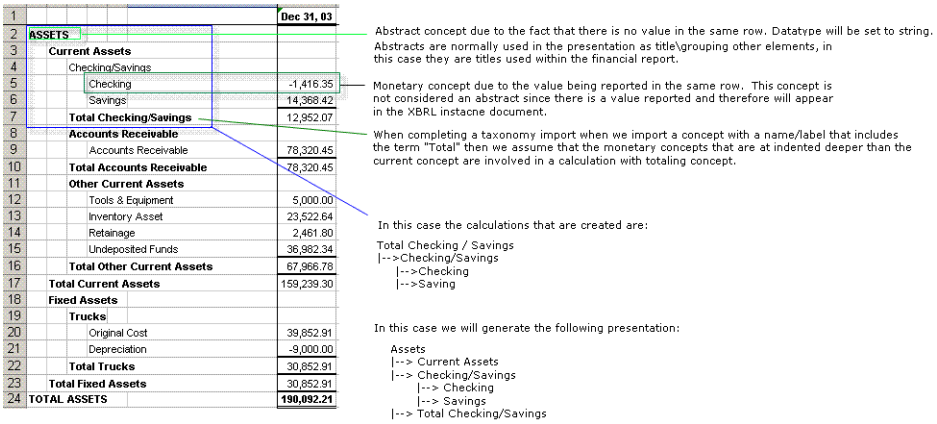
We want to import data from a Microsoft Excel document so we must set XBRL Taxonomy Designer to accept a Microsoft Excel document.

2 **From import source, select Excel.**

3 **In the File name field, click Browse to browse to `samples/Tutorial/Sample Company` folder and select the `sample-balancesheet.xls` file.**

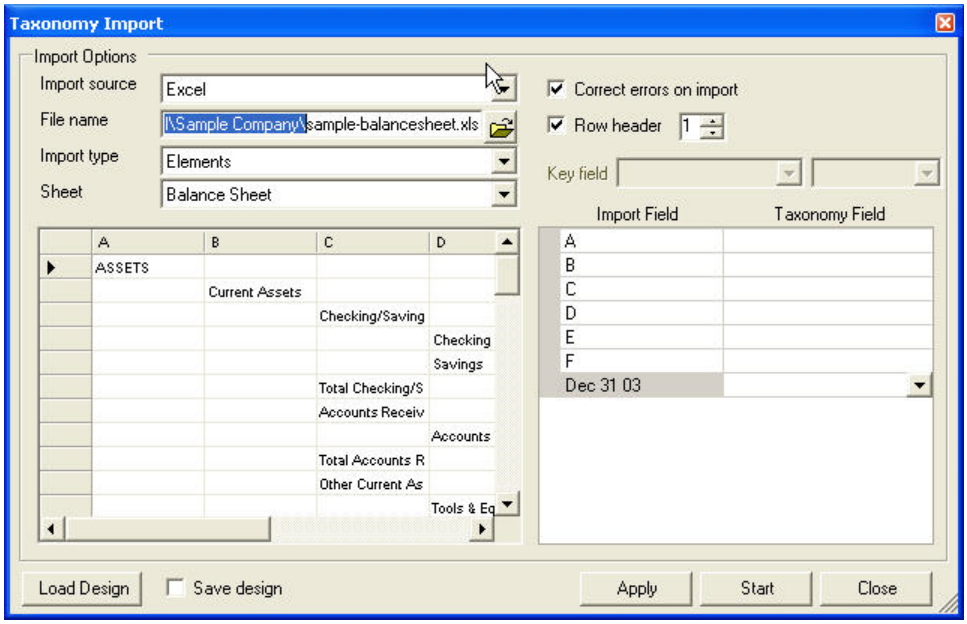


The sample company spreadsheet contains the labels for the concepts that we wish to import into our taxonomy. We generate the XBRL taxonomy data (schema, presentation, and calculations) expressed in this source by importing these labels. To start, we import the XBRL taxonomy elements using the import type "Elements". Note that we also generate elements from labels as labels were used in the source. See "Taxonomy Import Overview" on page 153.



4 From **Sheet**, select **Balance Sheet**.

This tells XBRL Taxonomy Designer that you are importing elements from a balance sheet.See "Taxonomy Import Overview" on page 153.



The Import Field to Taxonomy Field mapping grid has no entries, so nothing is recognized (based on column headings) for mapping. We must tell XBRL Taxonomy Designer how we want it to map information from `sample-balancesheet.xls` to the taxonomy.

5 **Click the drop-down button in Import Field A's Taxonomy Field column, and then select Label.**

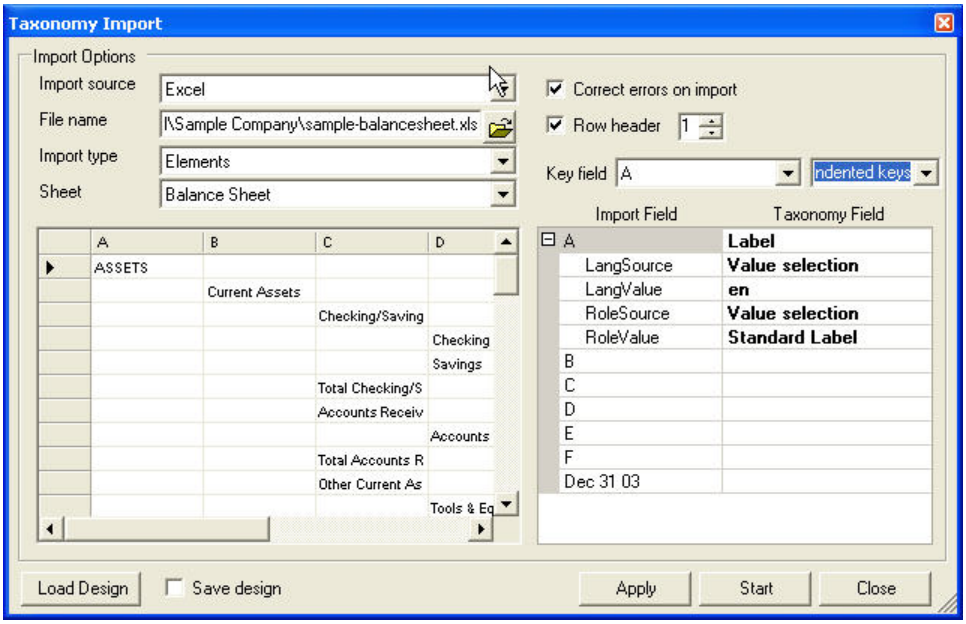
This sets XBRL Taxonomy Designer to define how to identify elements in a taxonomy when imported data is intended to update elements. In this case the key field specifies whether to match on Name, Label, and so on. See "Taxonomy Import Overview" on page 153.

There are indented labels in columns A to F (6 columns). XBRL Taxonomy Designer uses the indentation nesting to establish presentation relationships working down the indent level, and calculation relationships based on the word total appearing in some cell legends.

These two example financial reports are expressed in a Microsoft Excel spreadsheet. One report is expressed with an indented structure across multiple columns which indicate levels of indentation. The flat structure in the second example has no indentation and therefore we cannot infer the XBRL calculation and presentation relationships as can be done with the indented structure.

6 **Click the next drop-down button and select Indented key.**

This sets XBRL Taxonomy Designer to look for indented information. This needs to be done because sample-balancesheet.xls has indented the data and XBRL Taxonomy Designer does not know to look for the indented data until notified. See "Taxonomy Import Overview" on page 153



7 Click **Start.**

Line	Type	Message ID	Message
1	Info		Import to SAMP completed: 1 errors, 0 warnings
2	Info		Rows read: 57, skipped: 1, imported: 56
3	Info		Check for duplicates will be based on: Label
4	Info		No elements in taxonomy, all elements will be newly added from Excel
5	Error	txImpEltNoLabel	Row 58 indented label not found, row ignored

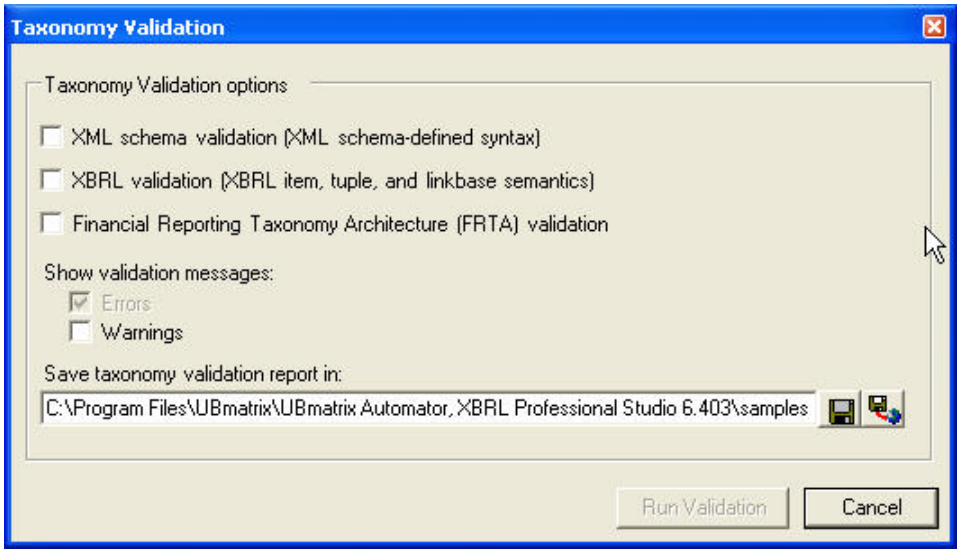
The message pane should look like the above example. The messages report that 57 rows were read and 56 were imported. As you might recall from Step 6, the Row Header is set to 1. This skips the first row as all of the information on row 1 is header information. XBRL Taxonomy Designer reads each row starting with row 2. It stops on row 57, because that is the last row of information. XBRL Taxonomy Designer continues to try to find information with row 58, but finds no information and so reports that as the error seen above.

8 **Display Presentation and Calculation Relationships Views Relationships Views side by side.**

This can be done by selecting View, then Show, then Add Another Relationship Pane and then selecting Calculation. Look in the Elements List pane to verify that monetary elements are monetary and presentation abstract elements are abstract.

Next, we must validate the taxonomy.

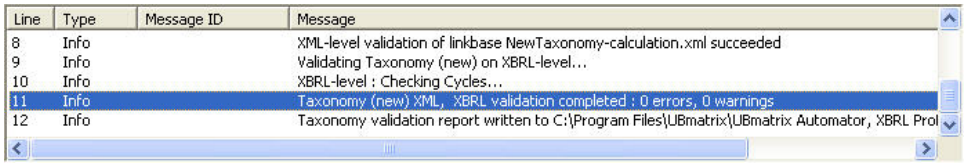
9 Select **Tools**, and then Validate **Taxonomy**.



10 **Select XML schema validation and then XBRL validation.**

These features set XBRL Taxonomy Designer to validate the taxonomy to make sure that the XML and XBRL are valid.

11 **Click Run Validation.**



Line	Type	Message ID	Message
8	Info		XML-level validation of linkbase NewTaxonomy-calculation.xml succeeded
9	Info		Validating Taxonomy (new) on XBRL-level...
10	Info		XBRL-level : Checking Cycles...
11	Info		Taxonomy (new) XML, XBRL validation completed : 0 errors, 0 warnings
12	Info		Taxonomy validation report written to C:\Program Files\UBmatrix\UBmatrix Automator, XBRL Prof

If you followed the directions correctly, you should receive the result as above.

You completed Importing Lesson One. Proceed to "Lesson Two: Importing to Create an Instance" on page 273.

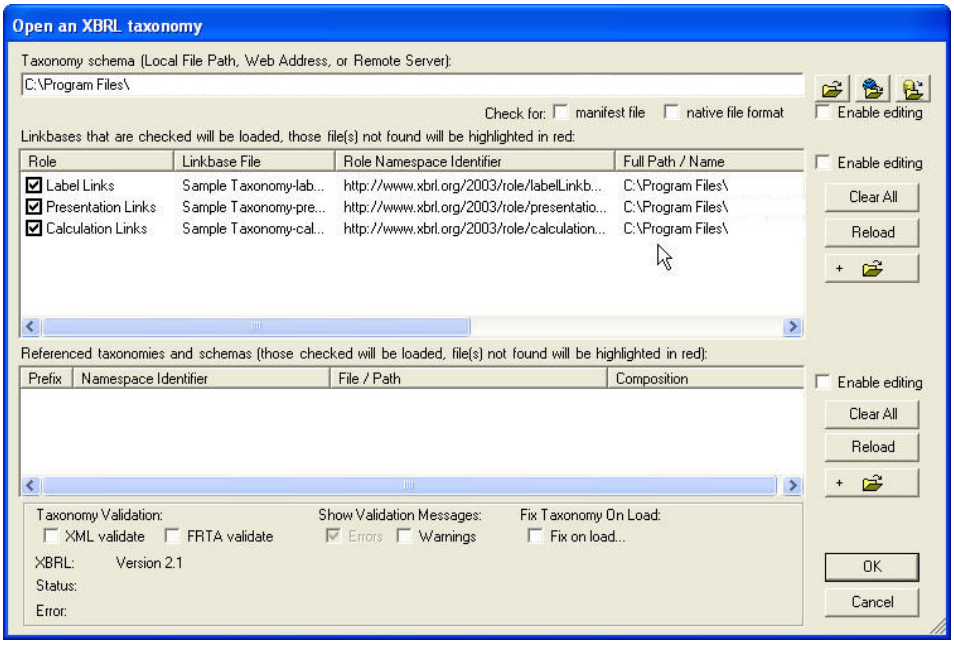
Lesson Two: Importing to Create an Instance

➤ To import an instance document:

1 **Create an instance document.**

2 **Select File, then Load, and then Taxonomy and click the icon in the Open Taxonomy window.**

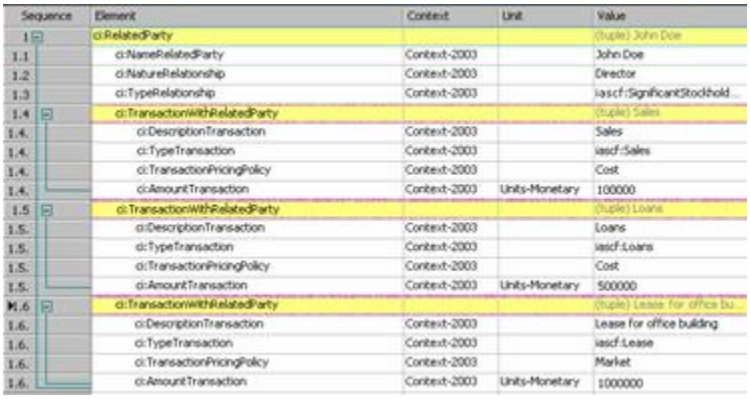
3 **Select the newly-created Sample Taxonomy file from Lesson One, and then click Open.**



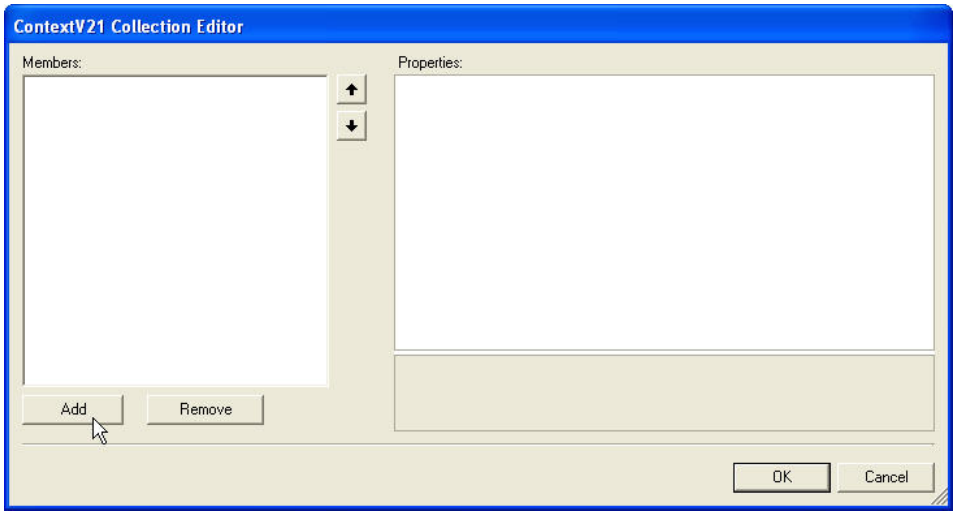
4 **Click OK in the Open Taxonomy window to load the taxonomy and close the window.**

Because the instance data is labeled with Sample Taxonomy labels and we are directly matching, we import without mapping. The next step is to create a context that we can map data into.

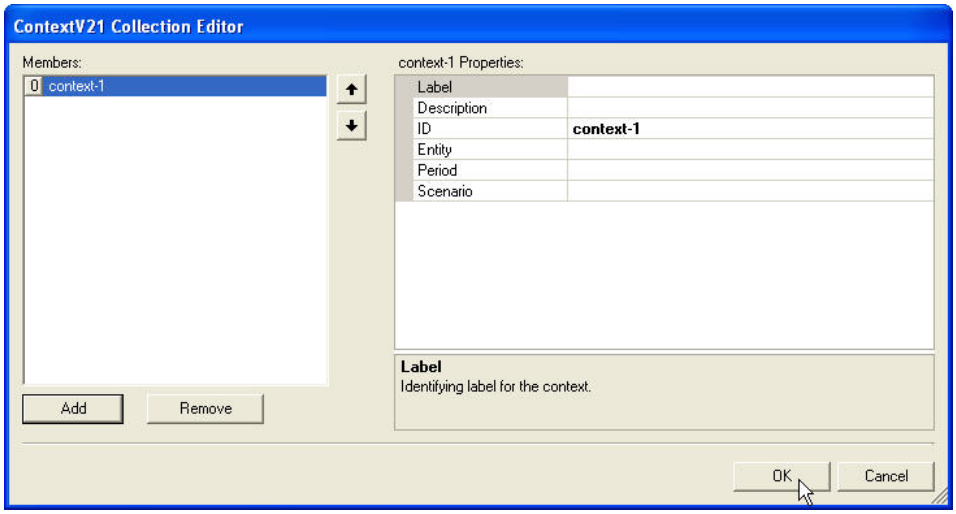
5 **Ensure that the Instance tab is active in the Instance Property Grid ("Instance Document Properties Grid" on page 121).**



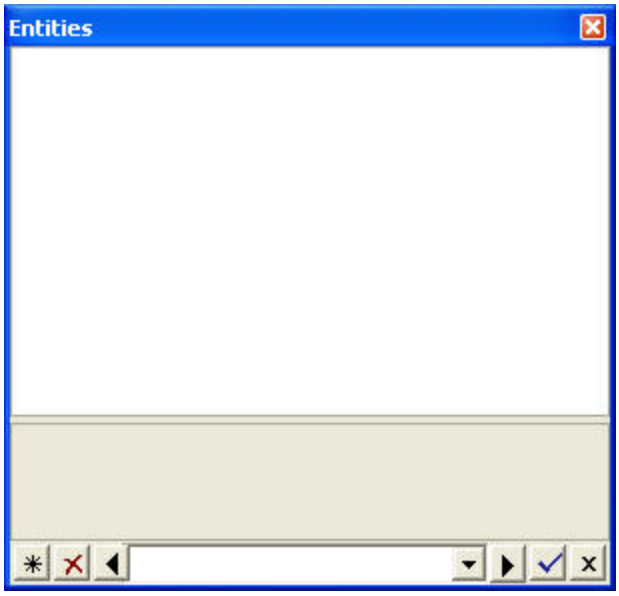
6 **Select the Contexts property in the Instance Property Grid and click the ellipsis (...) collection editor button.**



7 Click **Add** in the lower left of the collection editor.

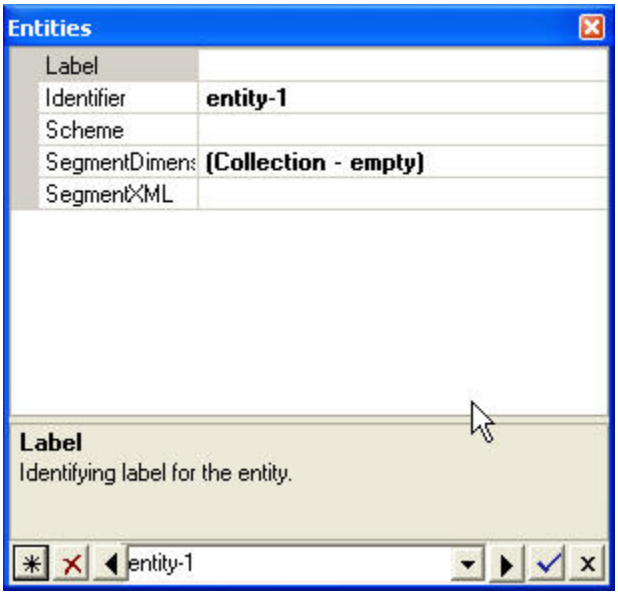


8 Click **Entity** and then select the pull-down menu.



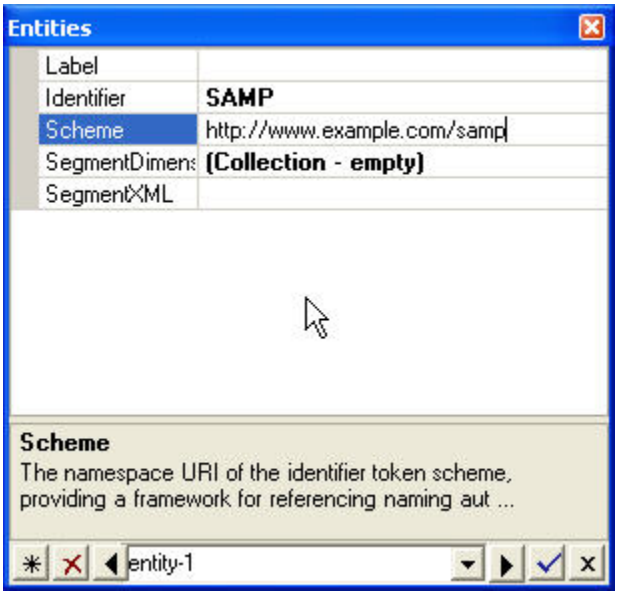
This is the Entity collection editor (see "Using Entities" on page 131).

9 Click * to add an entity.



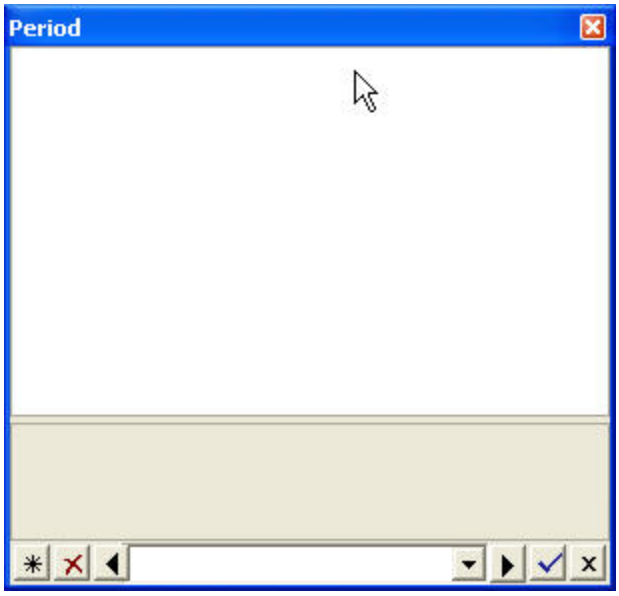
10 Change the identifier name to **SAMP**.

11 Add a scheme **http://www.example.com/samp** and then click the check button to accept the changes.

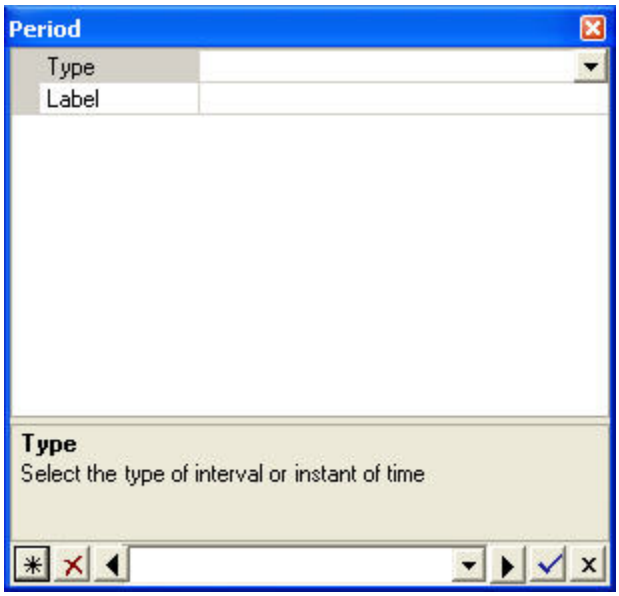


12 Click **Period** and then select the pull-down menu.

This is the Period collection editor.

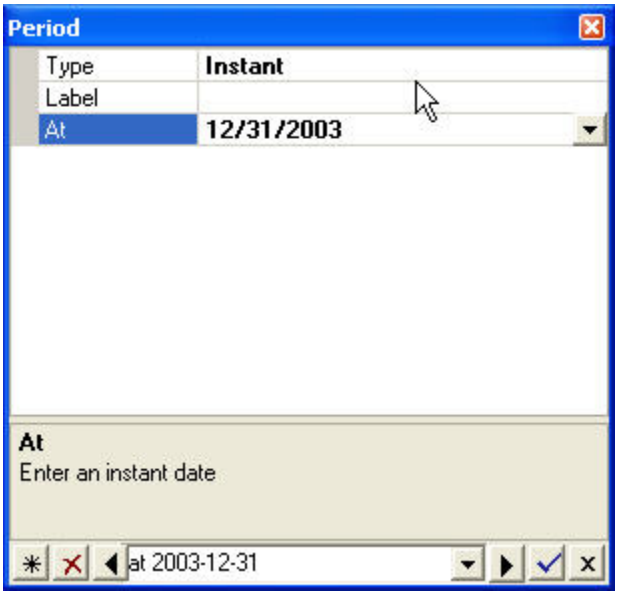


13 Click **New Period** (*).



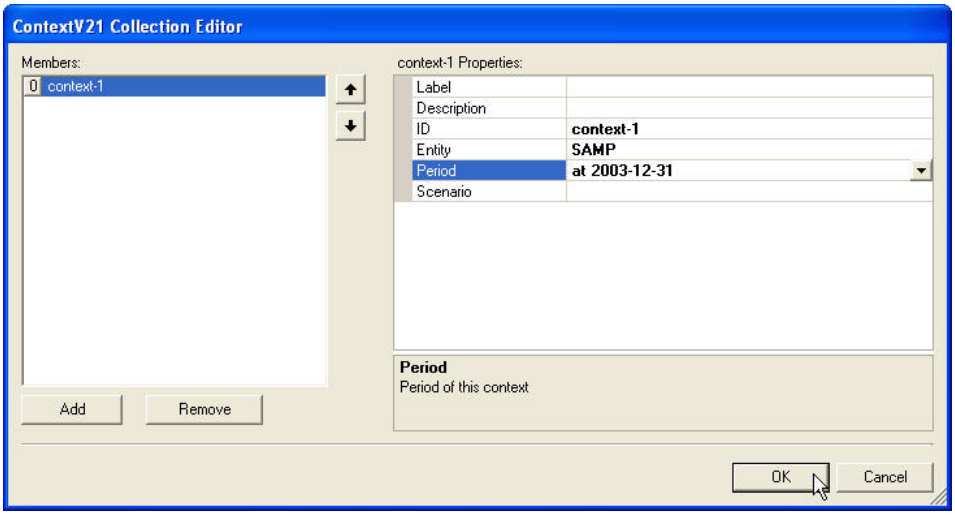
14 Select **Instant** from Type.

15 Select an At date of **December 31, 2003** from the calendar.



16 Click **Accept** (4) to save and exit the Period pop-up window.

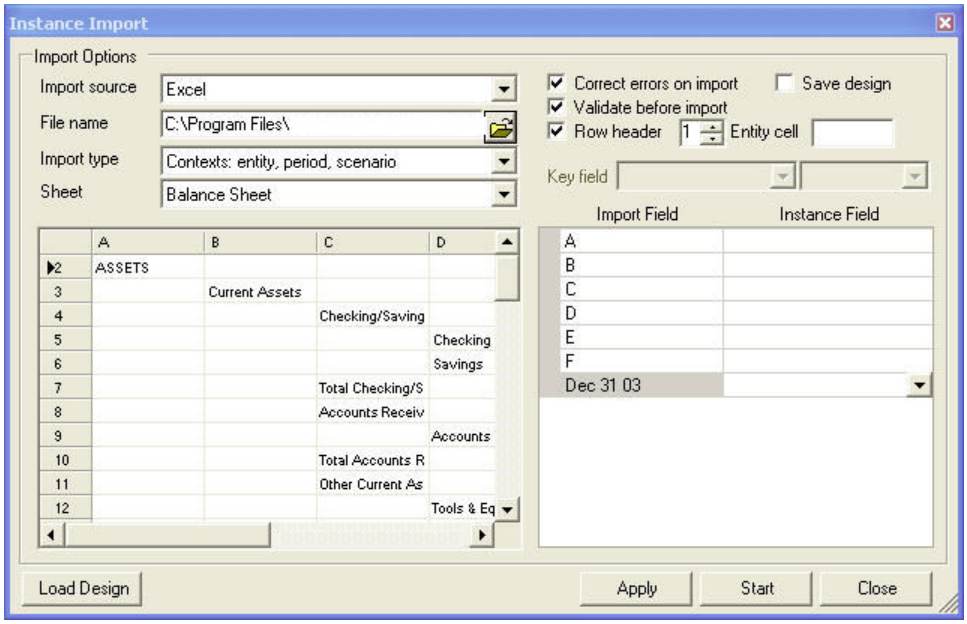
17 Click **OK** to exit the ContextV21 Collection Editor.



18 Before beginning the next step, click inside the Instance View pane to ensure the correct menus and toolbar buttons are activated.

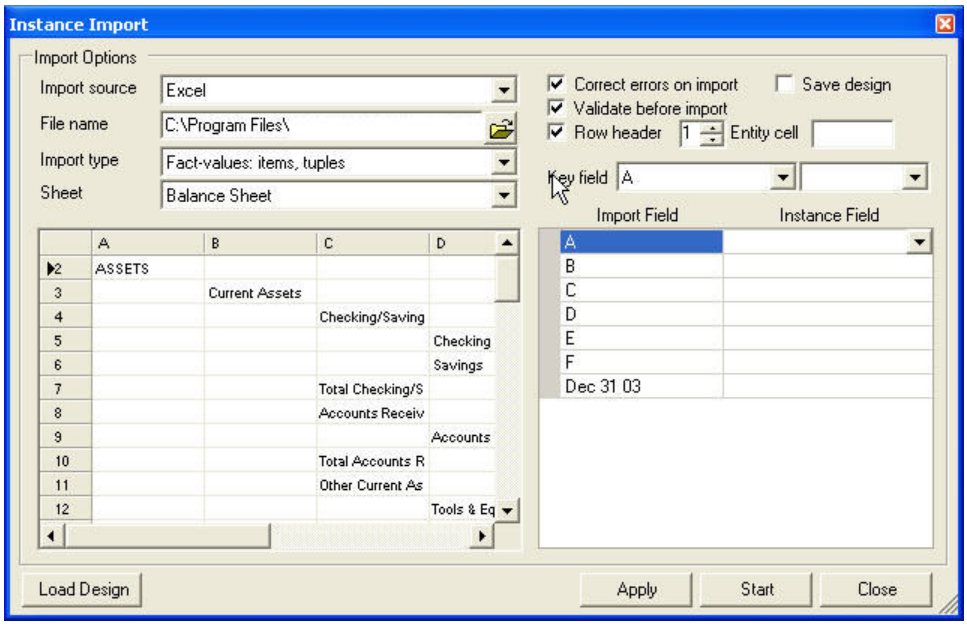
19 Select **File**, and then **Import**.

20 Browse to the `sample-balancesheet.xls` file.



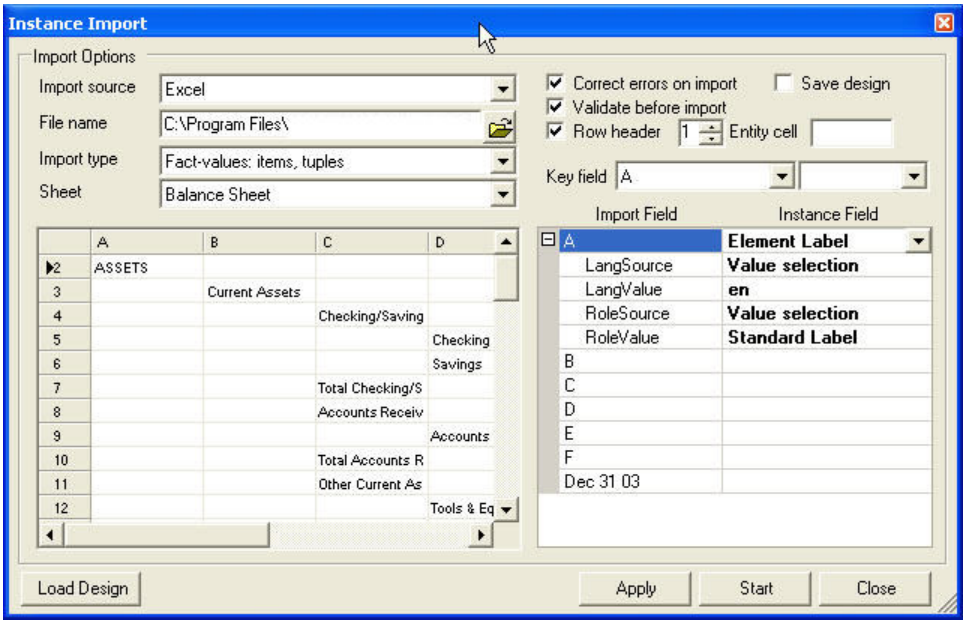
Previously, we imported relationships. Now, we're going to import fact values to populate your instance document.

21 Change the Import Type to **Fact-Values**.



22 In the field mapping section on the left side of the window, click row **A** and select **Element Label**.

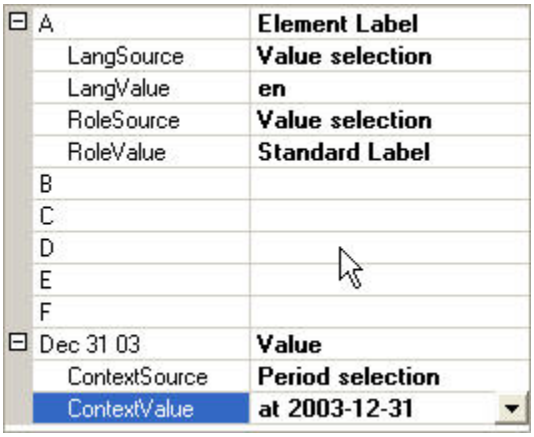
23 Click the + sign to expand.



24 In the **Dec 31 03** row, select **Value**.

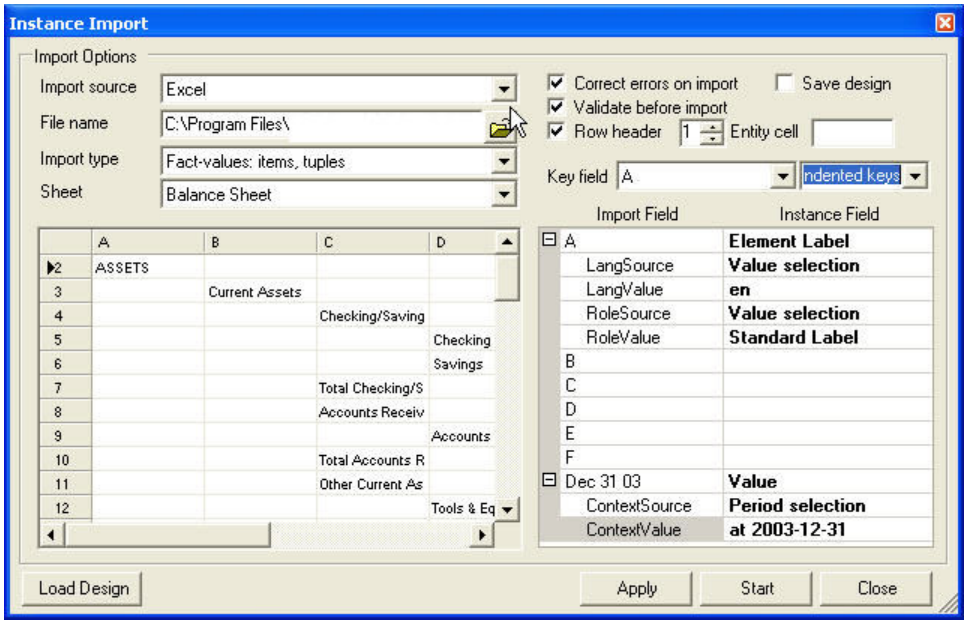
25 Click the + sign to expand the rows.

26 For **ContextValue**, select at **2003-12-31**.



The sample file is indented so we must ensure that XBRL Taxonomy Designer is set for this.

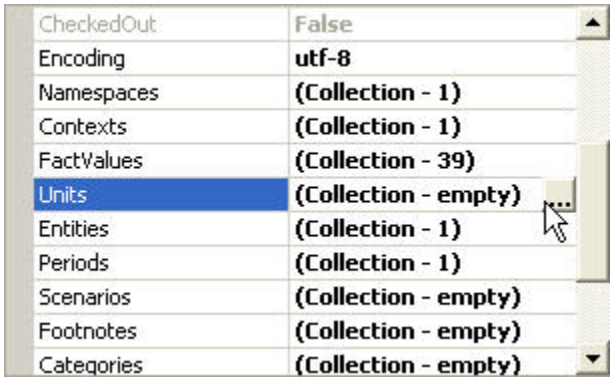
27 With Key Field mapped to **A**, click the drop-down button in the adjacent field and select **Indented Keys**.



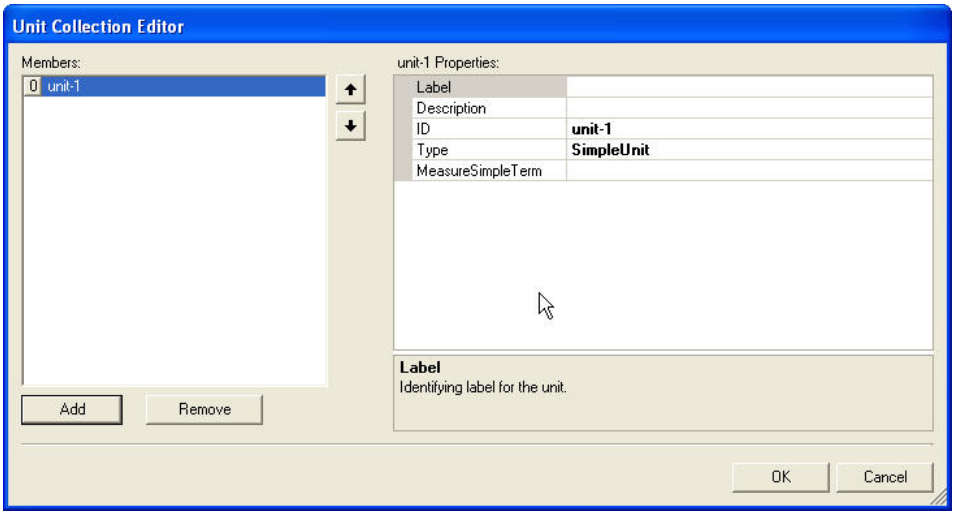
28 **Click Start to import and close the Instance Import window.**

In the Instance Properties Grid in the Instance Document Editor, the Fact Values property should show 39 entries in the collection. The Fact Values List pane should contain 39 entries with contexts, but without units.

29 **Open the Unit Collections editor.**



30 **Click Add.**



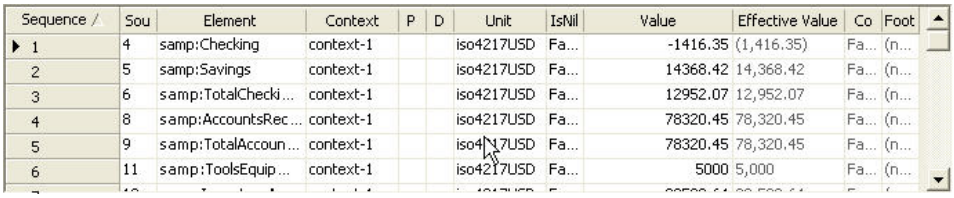
31 For the ID value, enter **units-monetary**.

32 Select the MeasurementSimpleTerm field, and then **select iso4217:USD**.

33 Click **OK**.

34 Click in the top row of the Unit column, and select **iso4217:USD**.

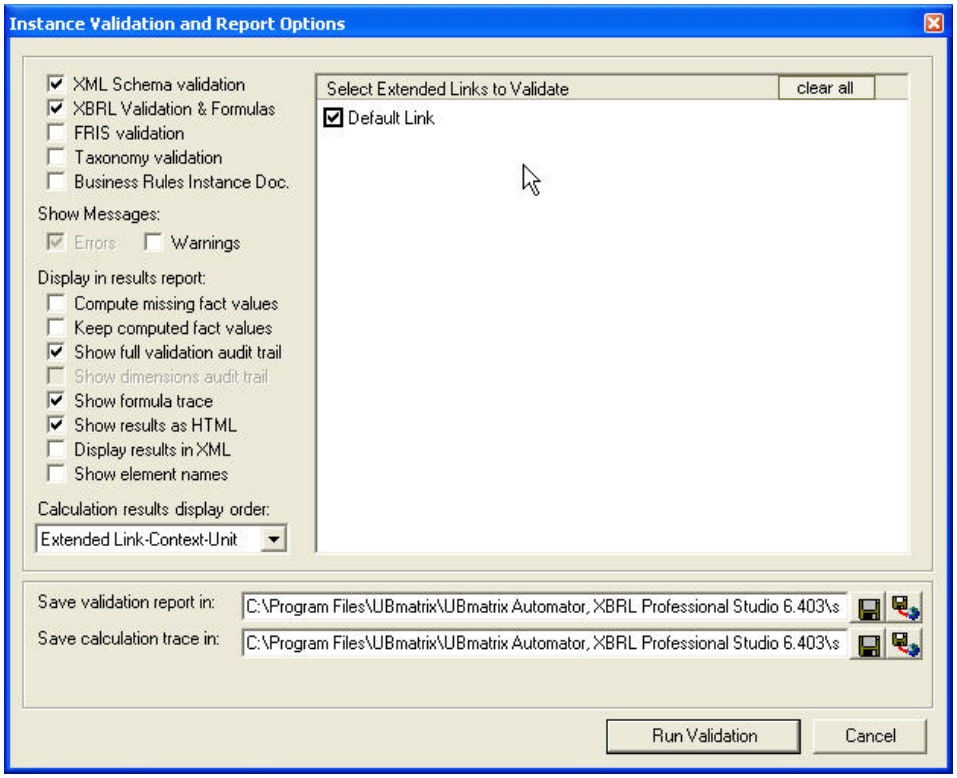
35 Right-click the Unit column header and select **Fill**, and then **Down** to populate the Unit column with the units-monetary value.



Sequence /	Sou	Element	Context	P	D	Unit	IsNil	Value	Effective Value	Co	Foot
1	4	samp:Checking	context-1			iso4217USD	Fa...	-1416.35	(1,416.35)	Fa...	(n...
2	5	samp:Savings	context-1			iso4217USD	Fa...	14368.42	14,368.42	Fa...	(n...
3	6	samp:TotalChecki...	context-1			iso4217USD	Fa...	12952.07	12,952.07	Fa...	(n...
4	8	samp:AccountsRec...	context-1			iso4217USD	Fa...	78320.45	78,320.45	Fa...	(n...
5	9	samp:TotalAccoun...	context-1			iso4217USD	Fa...	78320.45	78,320.45	Fa...	(n...
6	11	samp:ToolsEquip...	context-1			iso4217USD	Fa...	5000	5,000	Fa...	(n...

36 In the D (Decimals) column (if this column is not displayed, right-click anywhere within the factValues window and select Show, then Detail Columns), enter a **2** in the first row.

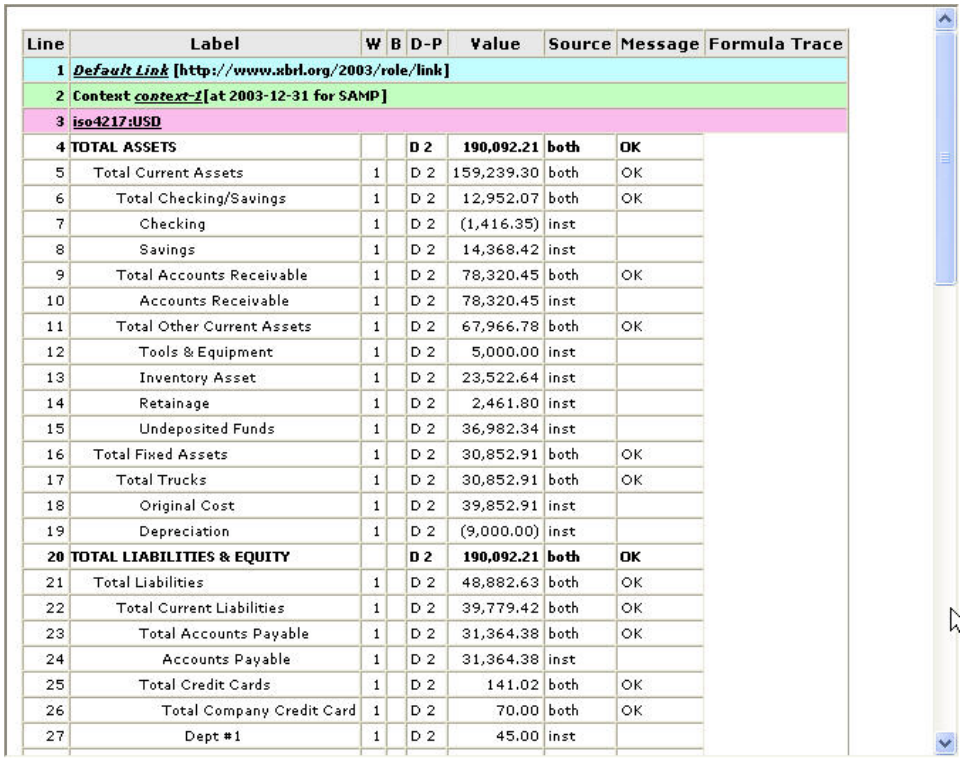
37 Validate the instance document by selecting **Tools**, and then **Validate Instance Document**.



38 Click **XML** and **XBRL** validation as shown above.

39 Click **Run Validation**.

A calculation trace validation is displayed in another window. If you followed the directions, it should show no errors as below.



Line	Label	W	B	D-P	Value	Source	Message	Formula Trace
1	***Default Link*** **[http://www.xbrl.org/2003/role/link]**							
2	**Context *context-1* [at 2003-12-31 for SAMP]**							
3	**iso4217:USD**							
4	**TOTAL ASSETS**			**D 2**	**190,092.21**	**both**	**OK**	
5	Total Current Assets	1		D 2	159,239.30	both	OK	
6	Total Checking/Savings	1		D 2	12,952.07	both	OK	
7	Checking	1		D 2	(1,416.35)	inst		
8	Savings	1		D 2	14,368.42	inst		
9	Total Accounts Receivable	1		D 2	78,320.45	both	OK	
10	Accounts Receivable	1		D 2	78,320.45	inst		
11	Total Other Current Assets	1		D 2	67,966.78	both	OK	
12	Tools & Equipment	1		D 2	5,000.00	inst		
13	Inventory Asset	1		D 2	23,522.64	inst		
14	Retainage	1		D 2	2,461.80	inst		
15	Undeposited Funds	1		D 2	36,982.34	inst		
16	Total Fixed Assets	1		D 2	30,852.91	both	OK	
17	Total Trucks	1		D 2	30,852.91	both	OK	
18	Original Cost	1		D 2	39,852.91	inst		
19	Depreciation	1		D 2	(9,000.00)	inst		
20	**TOTAL LIABILITIES & EQUITY**			**D 2**	**190,092.21**	**both**	**OK**	
21	Total Liabilities	1		D 2	48,882.63	both	OK	
22	Total Current Liabilities	1		D 2	39,779.42	both	OK	
23	Total Accounts Payable	1		D 2	31,364.38	both	OK	
24	Accounts Payable	1		D 2	31,364.38	inst		
25	Total Credit Cards	1		D 2	141.02	both	OK	
26	Total Company Credit Card	1		D 2	70.00	both	OK	
27	Dept #1	1		D 2	45.00	inst		

40 **Click the X in the upper right of the window to close the report.**

41 **Select File, and then Save As File.**

42 **Save the instance document as** `sampleID.xml`.

43 **Select File, and then Close to close the instance document.**

You completed Importing Lesson Two and the Importing Tutorial.

Mapping Tutorial

This tutorial walks you through the process of mapping. Each of these tutorials tackles a mapping problem. Mapping enables you to map from a non-xbrl source to an XBRL document and then convert that information into XBRL.

- "Mapping Lesson One" on page 285—Map a company-specific taxonomy to a Microsoft Excel Balance Sheet and generate an XBRL instance document

  In this lesson, we create a taxonomy based on (column indented) tabular input of an balance sheet that is contained within an Microsoft Excel document. Using the taxonomy import feature, we build our XBRL taxonomy by importing the financial concepts and the labels used for the names of those concepts. The calculations that occur within the balance sheet are imported during that importing process.

  Note: Importing from Microsoft Excel does not import the Excel formulas into XBRL formulas or calculations.

  After the XBRL taxonomy is built, we create the XBRL instance document which contains the reported values. We establish the context of the values being reported, for example the year in which they are being reported and for which entity/company. We establish the unit and precision of the values that we are reporting.

  Finally, we validate the instance document against the calculations that we built in the taxonomy and determine if the values reported are correct.

- "Mapping Lesson Two" on page 303—Map between different taxonomies and generate an migrate an XBRL instance document.

  In this lesson we map the company-specific `SampleCompany.xsd` taxonomy created in Mapping Lesson One to the standardized USGAAP taxonomy. Both taxonomies are included in `/samples/xbrl/usgaap` and `/samples/tutorial/mapping`. Note that GAAP stands for Generally Accepted Accounting Principles.

  Generally this approach is used to map between different taxonomies or taxonomy versions. This enables you to use the mapping to convert instance documents that were created against another taxonomy or version of a taxonomy. Next, we build a mapping file between the Sample Company taxonomy and the corresponding US-GAAP. This lesson utilizes the XBRL Mapper feature and maps the elements by dragging from the US-GAAP taxonomy calculation tree and dropping onto the calculation view of the Sample Company.

After the Sample Taxonomy is mapped to the USGAAP, we create a XBRL instance document by converting the XBRL instance generated in lesson one which was built against SampleCompany.xsd. Then we create an instance document built upon USGAAP, but using the values expressed in the sample company excel source file included in /samples/tutorial/mapping.

- "Mapping Lesson Three" on page 318—Map directly from a balance sheet to a standardized taxonomy such as USGAAP

  `In this lesson, we map the sample`-balancesheet.xls file directly to US-GAAP instead of to the Sample Taxonomy, using the drag-and-drop technique to map from the US GAAP to the Balance Sheet. Subsequently, we convert an instance document from the Sample Taxonomy to US-GAAP Taxonomy and validate it.

The files generated from the Mapping Lessons are included in the `Samples` folder for your review. You may find these files under `/Samples/Tutorial/Mapping/`.

XBRL concepts introduced in this tutorial are explained in greater detail in the XBRL specification. Taxonomy and instance authoring relation sections are detailed further in this online help in the relevant sections. It is recommended that users possess a basic familiarity with XBRL terminology and this product, as it helps understand some of the general mapping tutorial. As such, it is advised that users review the "Basic XBRL Taxonomy Designer Tutorial" on page 238 before attempting the Mapping tutorial.

Basic Mapping Concepts

This section provides an overview of how mapping works from within XBRL Taxonomy Designer. These components are needed for mapping:

- An XBRL taxonomy that contains the concepts of the instance document.
- The source data (currently in a table obtained by query or in a source instance document).
- A mapping specification which explains to the computer how to filter and relate the source data into XBRL taxonomy concepts, contextRefs, unitRefs, decimal and scale factors.
- An audit trail which shows the flow of data from the source to the target XBRL instance document.
- Exception reports that help you debug the mapping.

This section walks you through the basic import and mapping capabilities of XBRL Taxonomy Designer by using three Mapping Examples.

Mapping Lesson One

In this lesson, we create a taxonomy based on (column indented) tabular input of a Balance Sheet that is contained within an Microsoft Excel document. Using the taxonomy import feature we build our XBRL taxonomy, by importing the financial concepts, and the labels used for the names of those concepts. The calculations that occur within the balance sheet are imported during that importing process.

Note: Importing from Microsoft Excel does not import or translate the Excel formulas into XBRL formulas or calculations.

This lesson requires the file sample-balancesheet.xls, which is available in this folder: `samples\tutorial\mapping\sample Company.`

Completed files created from this lesson can be found at `samples\tutorial\mapping\Mapping1\.`

When working with a locked taxonomy such as the USGAAP, XBRL Taxonomy Designer locks the taxonomy from being altered. For information on removing this lock, see "Enabling Editing" on page 25.

➤ To create a taxonomy:

1 **Create a taxonomy.**

2 **In the Taxonomy Property Grid, set the NamespacePrefix to samp and NamespaceIdentifier to http://www.example.com/samp.**

The namespace identifier uniquely identifies the schema document and the elements that belong under the namespace. The namespace prefix is the alias which refers to the identifier.

3 **Select File, then Save As to save the taxonomy as Sample Taxonomy.**

4 **Select File As File, then Close to close the taxonomy. to close the taxonomy.**

Note: For ease of use, it is recommended that you save all files for this lesson in `samples\tutorial\mapping\Sample Company\WorkBook` folder to keep track of the file for use in future lessons. The `Workbook` folder has added to the mapping folder to provide a place to save documents without overwriting the sample files.

In the following procedure, we are going to import labels from a sample Microsoft Excel document into the Sample Taxonomy. We are creating a taxonomy based upon the excel file and the labels that are expressed for financial concepts in the excel file. Concepts and labels are created as a result of this import.

➤ To import indented labels to automatically create linkbases:

1 **Begin the taxonomy import by selecting File, then Import.**

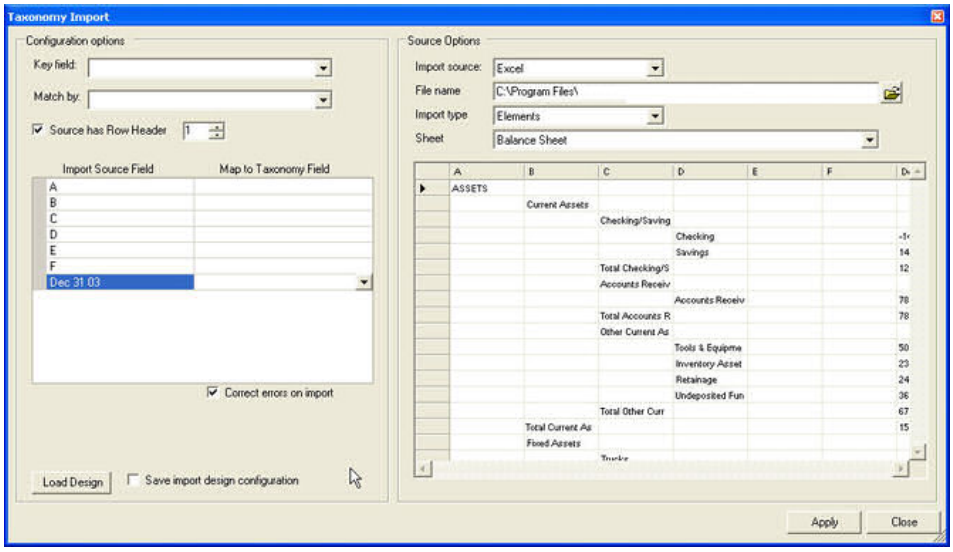
This opens the Import Taxonomy window.

We want to import data from a Microsoft Excel document so we must set XBRL Taxonomy Designer to accept a Microsoft Excel document.

2 **From import source, select Excel.**

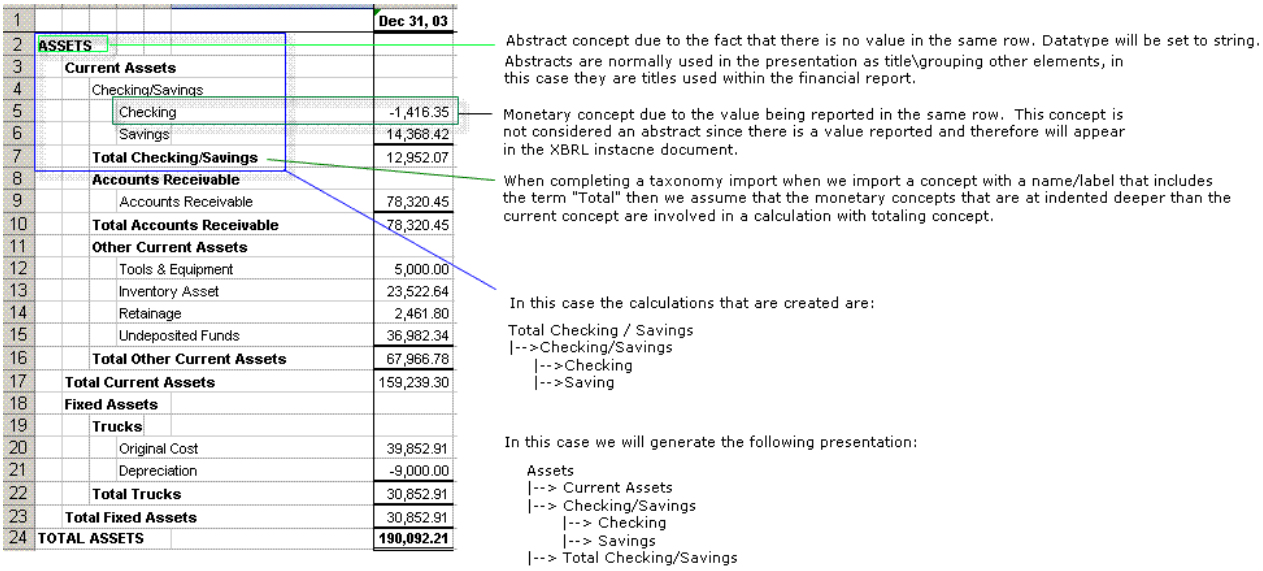
3 **In the File name field, click Browse to navigate and select the `sample-balancesheet.xls` file.**

Note: In a typical installation, the path is `C:\Program Files\Oracle\Disclosure Management\XBRL Taxonomy Designer\samples\Tutorial\Mapping\Sample Company\.`



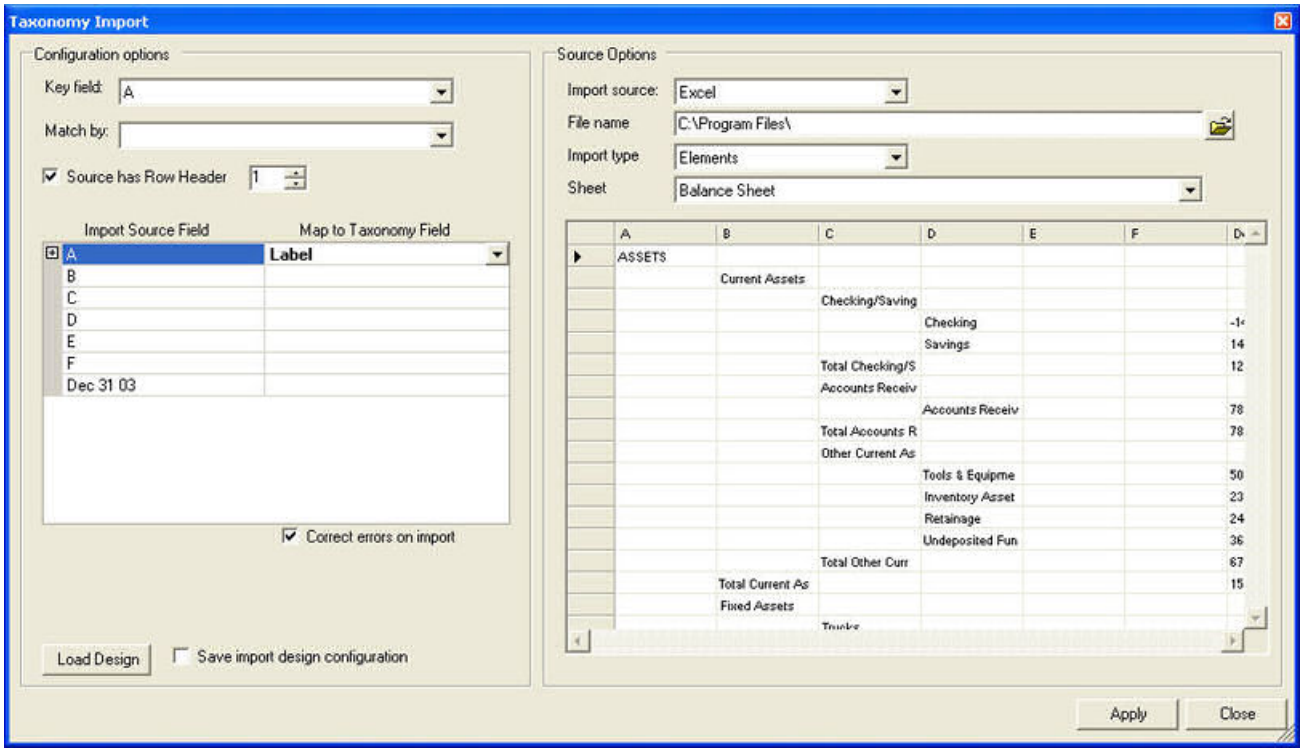
The sample company spreadsheet contains the labels for the concepts that we wish to import into our taxonomy. We generate the XBRL taxonomy data (schema, presentation, and calculations) expressed in this source by importing these labels. To start, we import the XBRL taxonomy elements using the import type Elements.

Note that elements from labels are also generated as labels were used in the source.



4 From Sheet, select **Balance Sheet.**

This calibrates XBRL Taxonomy Designer to import elements from a balance sheet.



The Import Field to Taxonomy Field mapping grid has no entries, so nothing is recognized (based on column headings) for mapping. XBRL Taxonomy Designer must be configured so that it maps information from `sample-balancesheet.xls` to the taxonomy.

5 **Click the drop-down in the Taxonomy Field column opposite of the Import Field A.**

6 **Select Label from the list.**

This calibrates XBRL Taxonomy Designer to define how to identify elements in a taxonomy when imported data is intended to update elements. In this case the key field specifies whether to match on Name, Label, and so on.

There are indented labels in columns A to F (6 columns). XBRL Taxonomy Designer uses the indentation nesting to establish presentation relationships working down the indent level, and calculation relationships based on the word Total appearing in some cell legends.

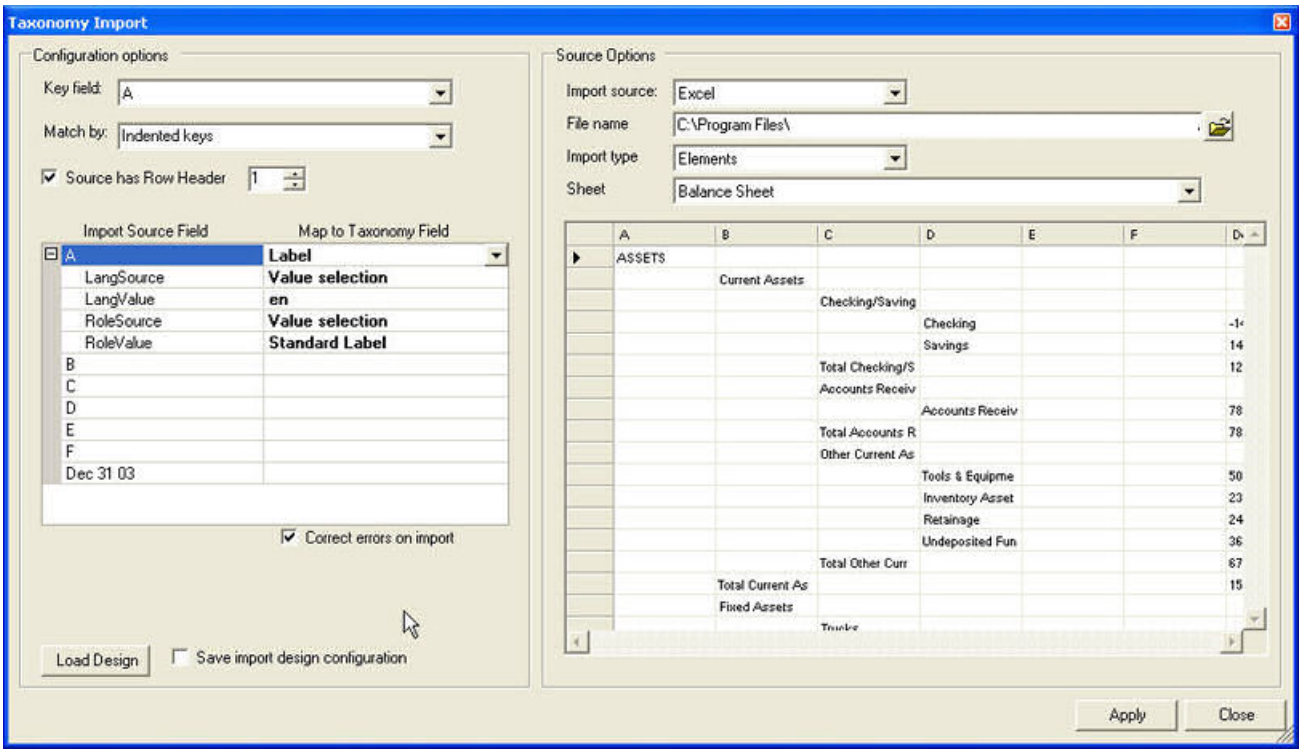
Unlike the example used in lesson one the example balance sheet below does not use column indention. As a result, the XBRL Taxonomy Designer taxonomy import feature can not establish calculation and presentation relationships.

The example below has the same information, but it is presented with indentation. Using this indentation, XBRL Taxonomy Designer can infer the XBRL calculation and presentation relationships from the Excel document.

Using the indented structure in the above example to signify relationships between concepts, XBRL Taxonomy Designer interpret the data and converts it into this tree.

7 **Click the drop-down to the right of Key Field and select Indented key.**

This sets XBRL Taxonomy Designer to look for indented information. The reason this is done is that sample-balancesheet.xls has indented the data and XBRL Taxonomy Designer does not know to look for the indented data until notified.



8 Click **Apply**.

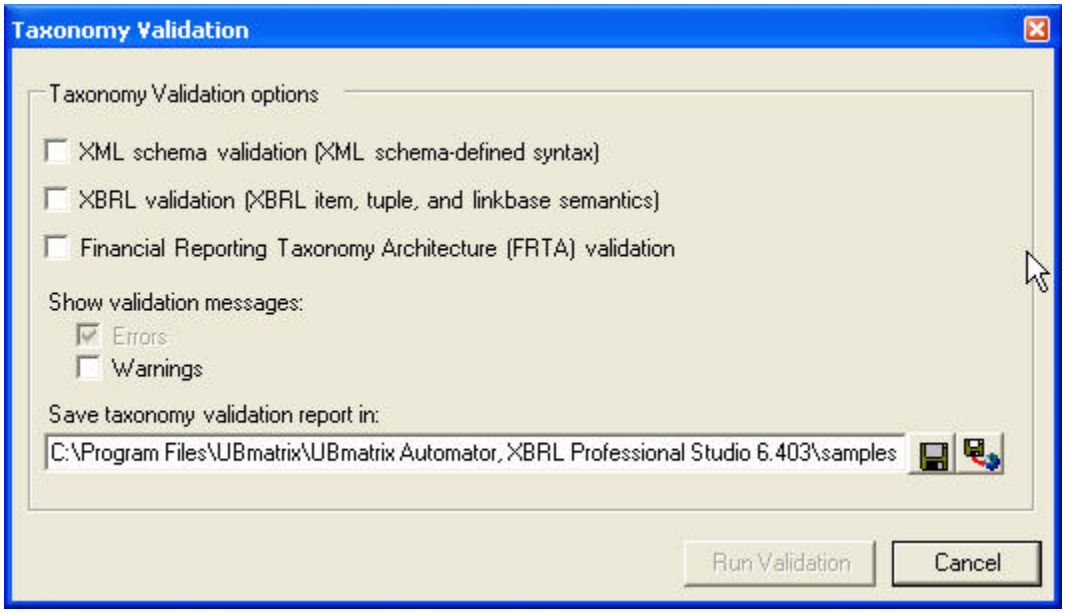
9 Click **Close**.

The message pane reports that 57 rows were read and 56 were imported. As you might recall from Step 6, the Row Header is set to 1. Because of this, XBRL Taxonomy Designer reads each row starting with row 2. XBRL Taxonomy Designer stops on row 57, because that is the last row with information.

10 Display Presentation and Calculation Relationships Views Relationships Views side by side by selecting **View**, then **Show**, then **Add Another Relationship Pane**, and then selecting **Calculation**.

Next, we must validate the taxonomy.

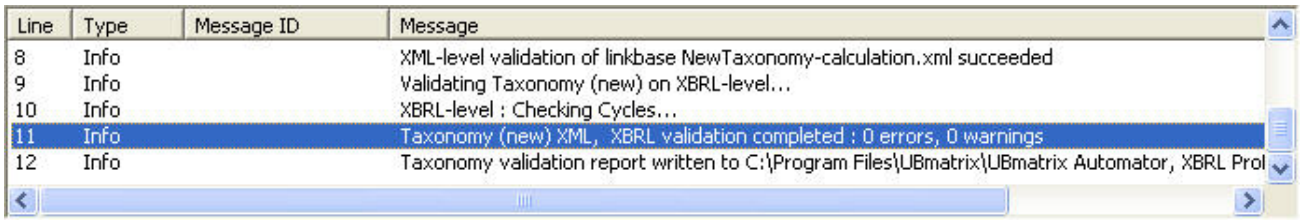
11 Select **Tools**, then **Validate Taxonomy**.



12 **Select XML schema validation and XBRL validation.**

These features set XBRL Taxonomy Designer to validate the taxonomy to make sure that it is XML and XBRL valid.

13 **Click Run Validation.**



Line	Type	Message ID	Message
8	Info		XML-level validation of linkbase NewTaxonomy-calculation.xml succeeded
9	Info		Validating Taxonomy (new) on XBRL-level...
10	Info		XBRL-level : Checking Cycles...
11	Info		Taxonomy (new) XML, XBRL validation completed : 0 errors, 0 warnings
12	Info		Taxonomy validation report written to C:\Program Files\UBmatrix\UBmatrix Automator, XBRL Prof

If you followed the directions correctly, you should receive the result as above.

In the following procedure, we are going create an XBRL instance report by importing the reported values from the Microsoft Excel source document.

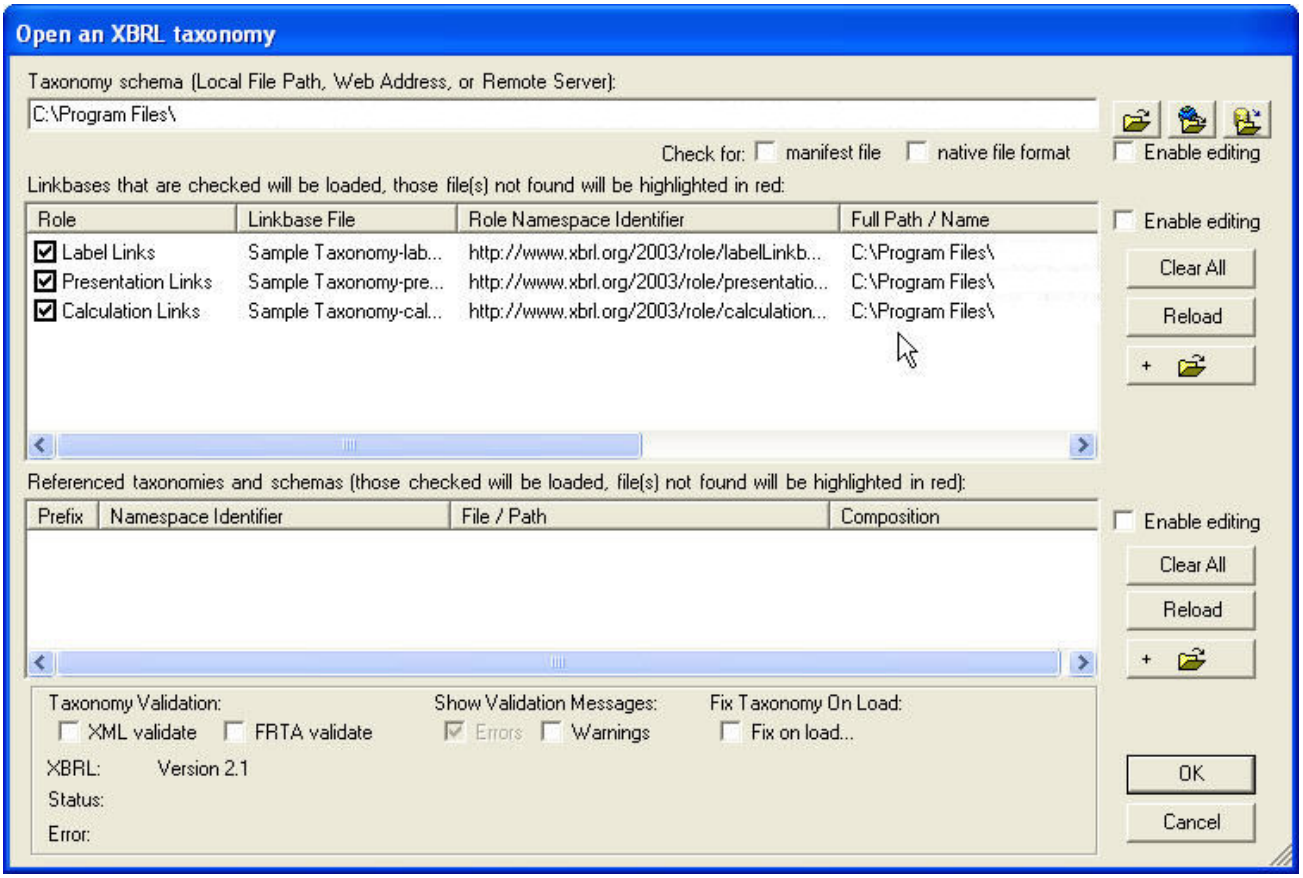
➤ To create an instance document

1 **Select File, then New, and then Instance to create an instance document.**

2 **Select File, then Load, then Taxonomy, and then click the icon in the Open Taxonomy window.**

XBRL instance documents require an accompanying XBRL taxonomy so that the report can be validated against the schema defined for the financial report. In this case, we are going to reference the `Sample Taxonomy.xsd` created in previous steps.

3 **Select the newly-created `Sample Taxonomy.xsd` file and click Open.**



4 **Click OK in the Open Taxonomy window to load the taxonomy and close the window.**

Because the instance data is labeled with Sample Taxonomy labels and we are directly matching, we import without mapping.

There are values reported in the source document under the title Dec 31, 03. This is considered the contextual label. To this context, we add additional contextual information (such as entity being reported, period reported for) to articulate what the contextual meaning is for the label.

	A	B	C	D	E	F	G
1							**Dec 31, 03**
2	**ASSETS**						
3		**Current Assets**					
4			Checking/Savings				
5				Checking			-1,416.35
6				Savings			14,368.42
7			**Total Checking/Savings**				12,952.07
8			**Accounts Receivable**				
9				Accounts Receivable			78,320.45
10			**Total Accounts Receivable**				78,320.45
11			**Other Current Assets**				
12				Tools & Equipment			5,000.00
13				Inventory Asset			23,522.64
14				Retainage			2,461.80
15				Undeposited Funds			36,982.34
16			**Total Other Current Assets**				67,966.78
17		**Total Current Assets**					159,239.30
18		**Fixed Assets**					
19			**Trucks**				
20				Original Cost			39,852.91
21				Depreciation			-9,000.00
22			**Total Trucks**				30,852.91
23		**Total Fixed Assets**					30,852.91
24	**TOTAL ASSETS**						**190,092.21**

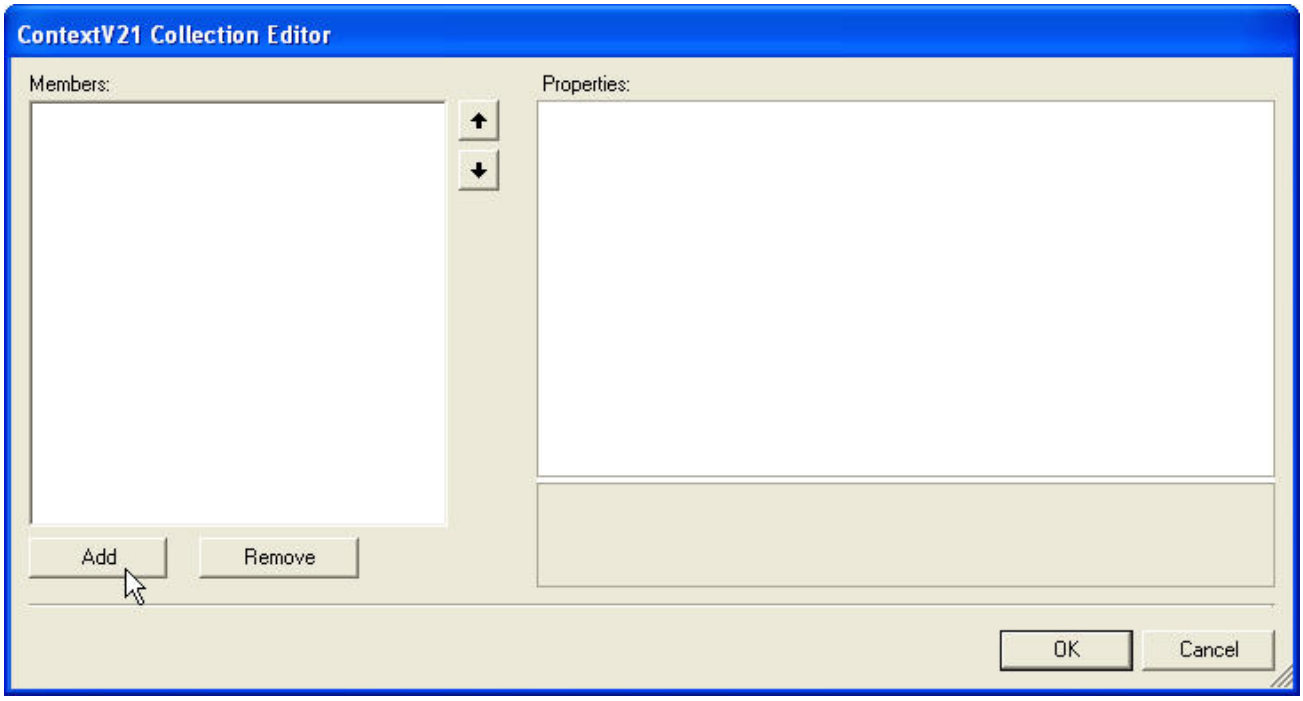
The financial details that are reported for a particular entity/company in a given period of time is considered the context of the information reported in the XBRL instance document.

The context label in this example is "Dec 31, 03" for the entity "SAMP", for the period December 31, 03; all of the concepts with a reported value with be linked to this context.

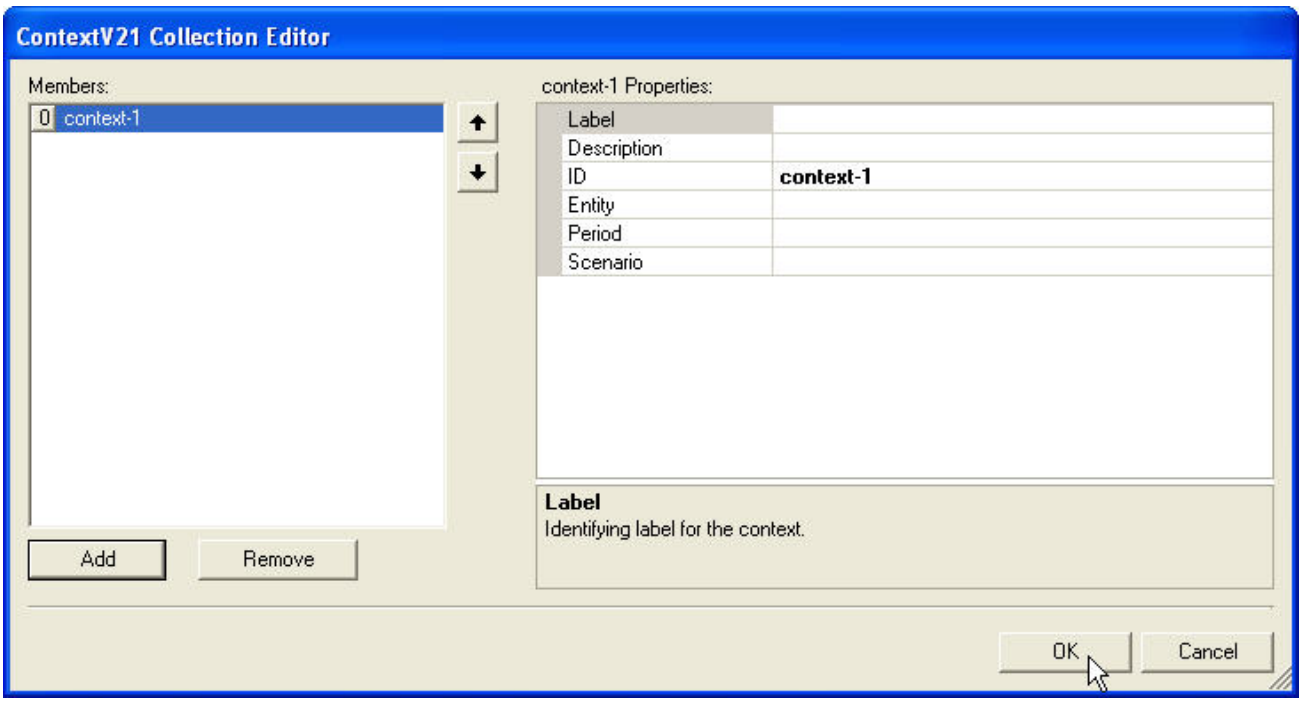
➤ To establish a context:

1 Click the Instance field to ensure that the Instance tab is active in the Instance Property Grid.

2 In the Instance Property Grid, click the **Contexts** property and then click the ellipsis (...).



3 In the lower left of the collection editor, click **Add**.



4 Click **Entity** and then select the drop-down menu.

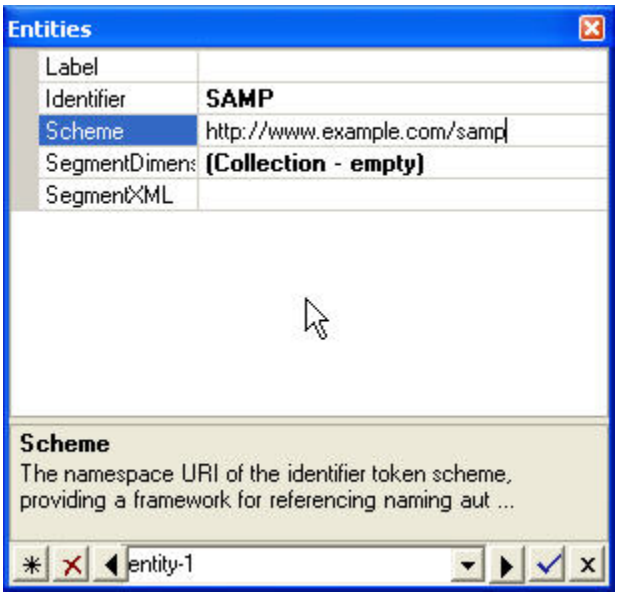
This opens the Entity collection editor.

5 Click * to add an entity.

6 Change the identifier name to SAMP.

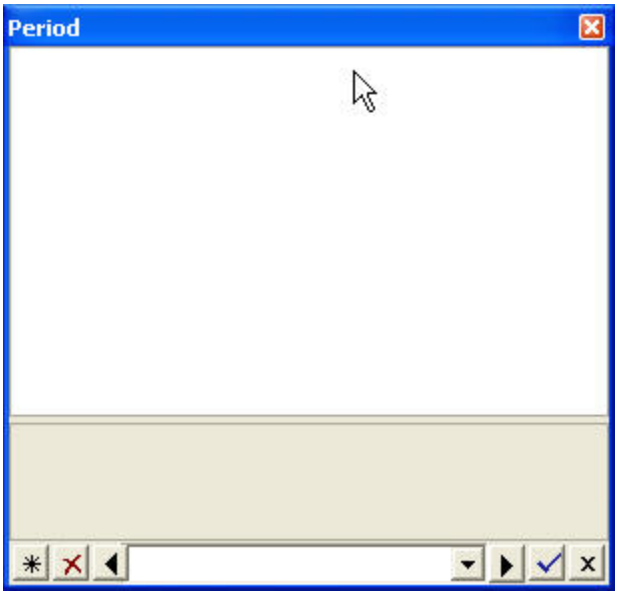
7 Add `http://www.example.com/samp` to the Scheme field.

8 Click **Accept (4)** to accept the changes.

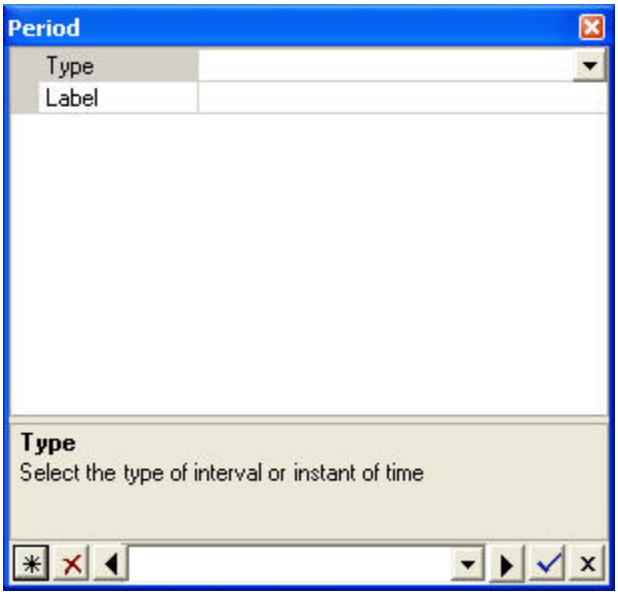


9 Click **Period** and then select the drop-down menu.

This opens the Period collection editor.

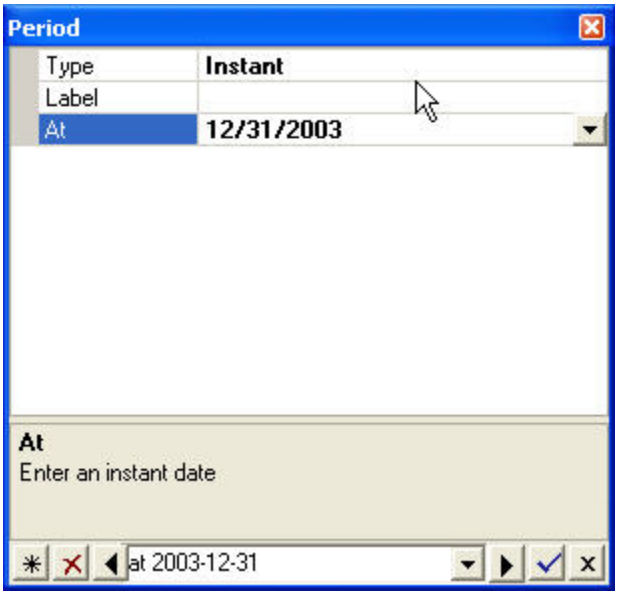


10 Click **New Period** (*) at the bottom of the dialog box.



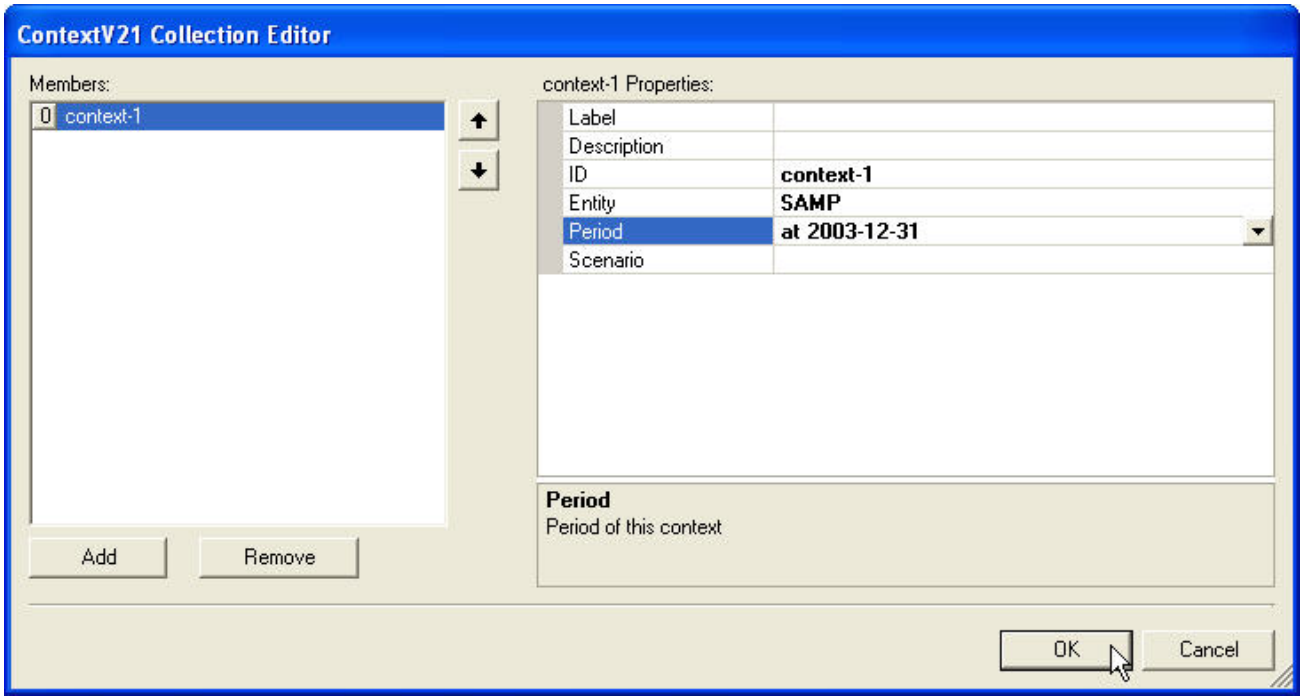
11 From **Type**, select **Instant**.

12 Select a date of **December 31, 2003** from the calendar.



13 Click **Accept (4)** to save and exit the Period pop-up window.

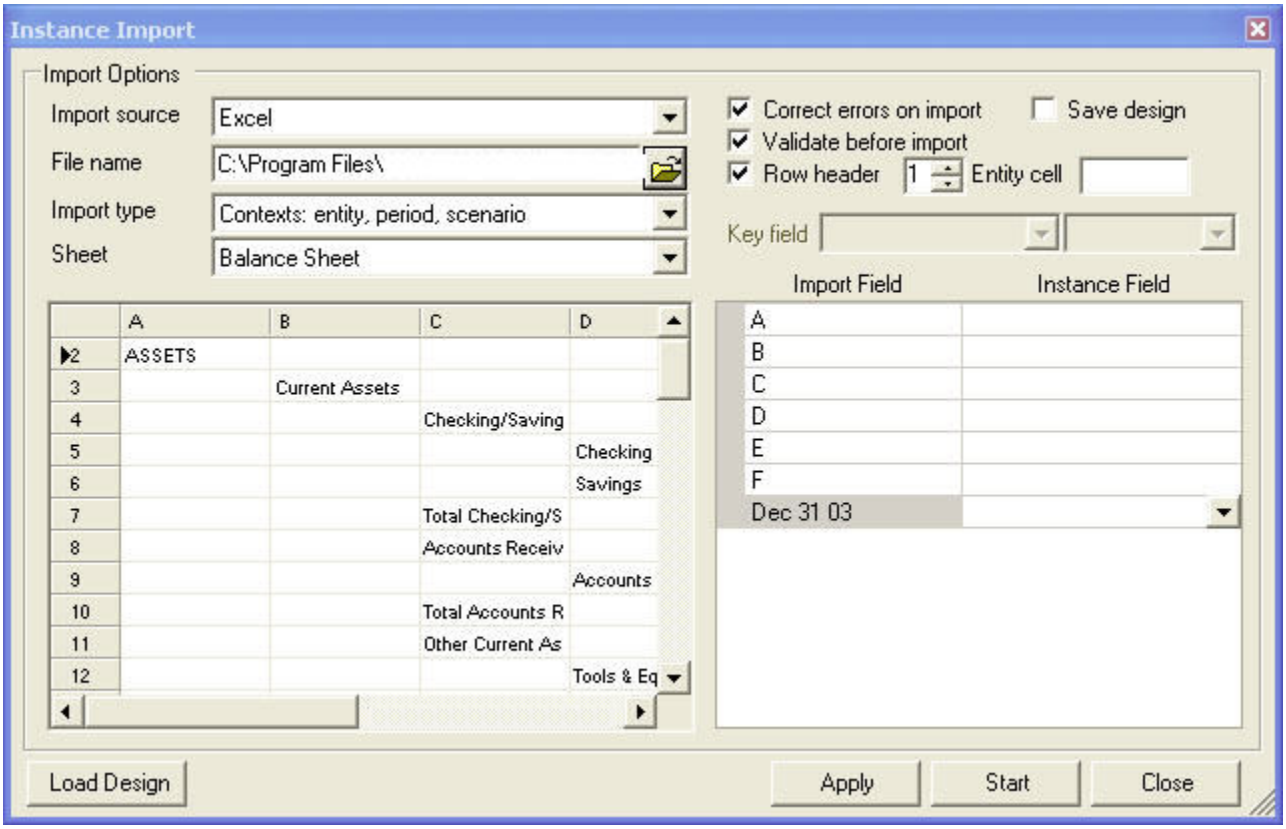
14 Click **OK** to exit the ContextV21 Collection Editor.



15 Click inside the Instance View pane to ensure the correct menus and toolbar buttons are activated.

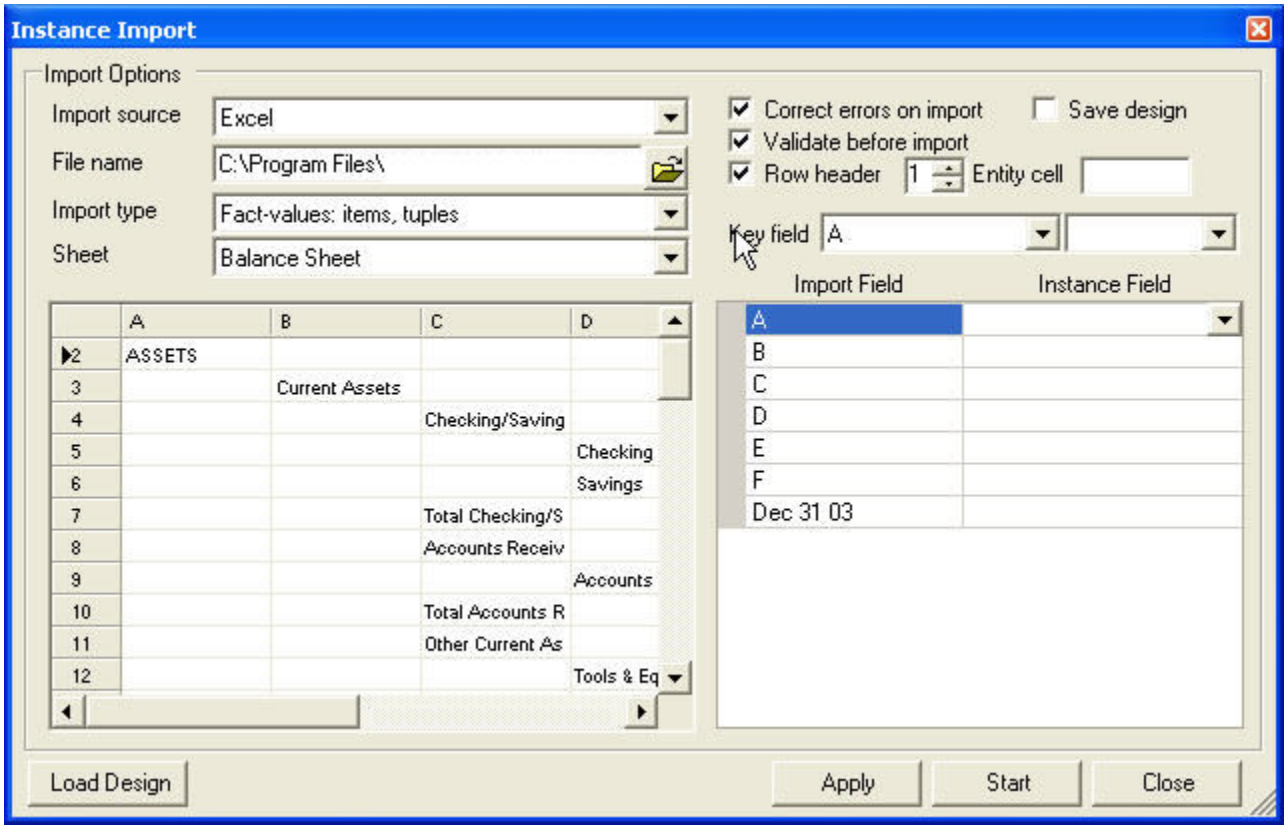
16 Select **File**, and then **Import**.

17 Browse to the `sample-balancesheet.xls` file.

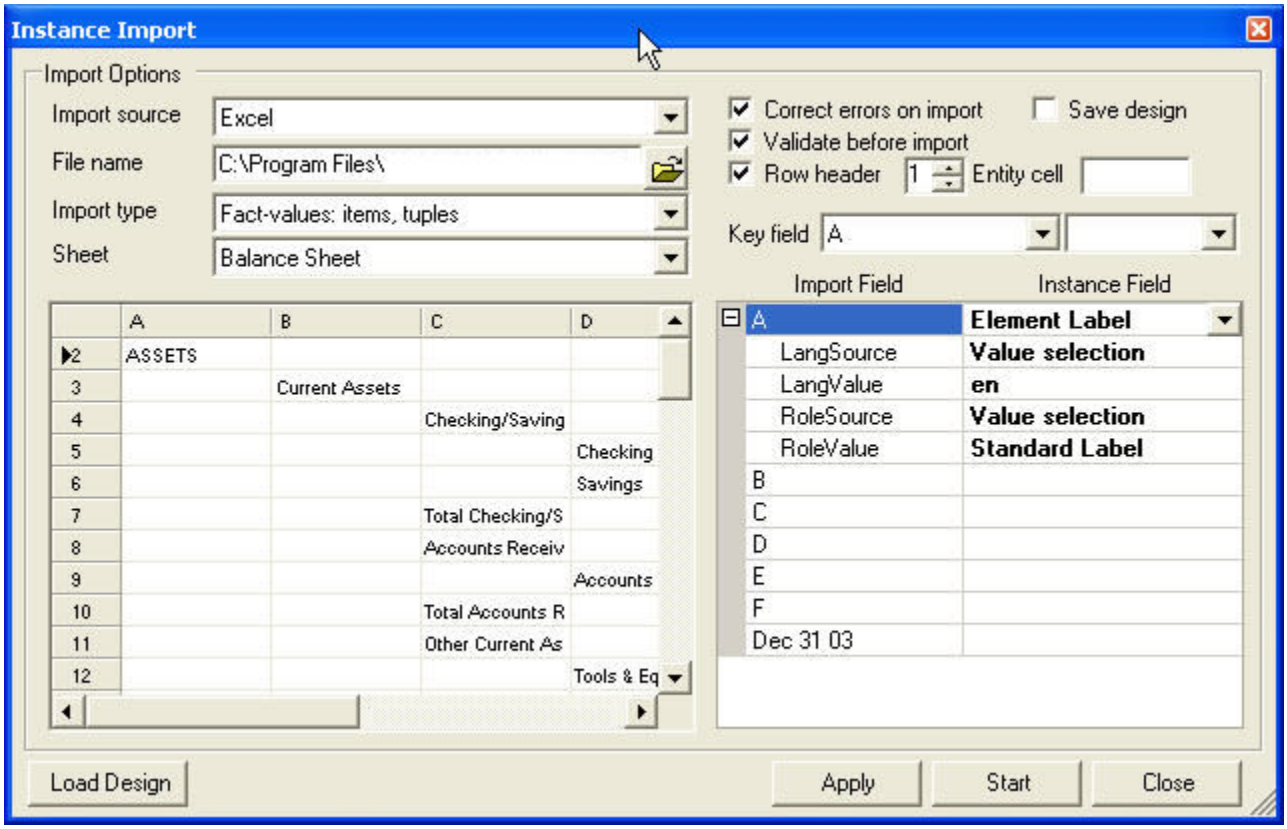


Previously, we imported relationships. Now, we're going to import fact values to populate your instance document.

18 Change the Import Type to **Fact-Values**.



19 In the field mapping section on the left side of the window, click row A and select **Element Label**, and then click the + sign to expand.



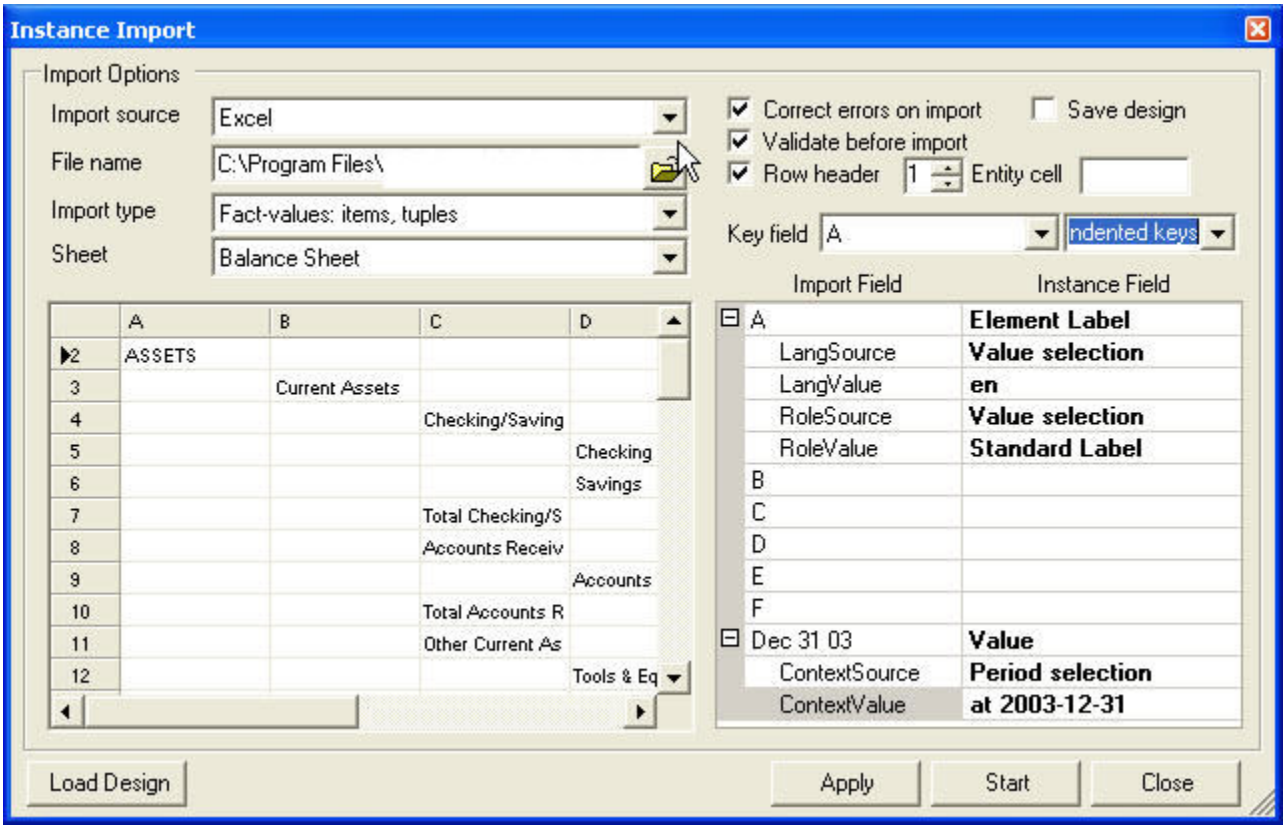
20 In the Dec 31 03 row, select **Value**, and then click the + sign to expand the rows.

21 For ContextValue, select **at 2003-12-31**.

A	**Element Label**
LangSource	**Value selection**
LangValue	**en**
RoleSource	**Value selection**
RoleValue	**Standard Label**
B	
C	
D	
E	
F	
Dec 31 03	**Value**
ContextSource	**Period selection**
ContextValue	**at 2003-12-31**

The sample file is indented so we must ensure that XBRL Taxonomy Designer is calibrated for this.

22 With Key Field mapped to A, click the drop-down list in the adjacent field and select **Indented Keys**.



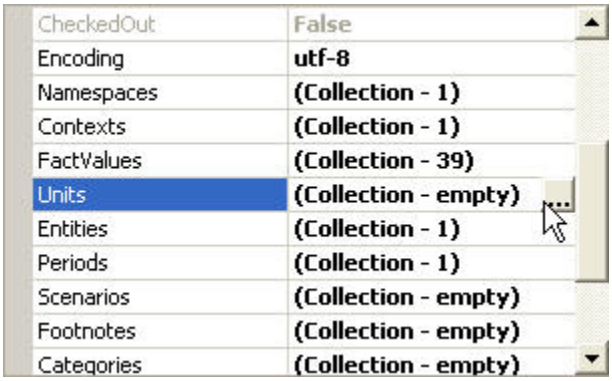
23 Click **Start** to import and close the Instance Import window.

In Instance Properties Grid, the Fact Values property should show 39 entries in the collection. The Fact Values List pane should contain 39 entries with contexts.

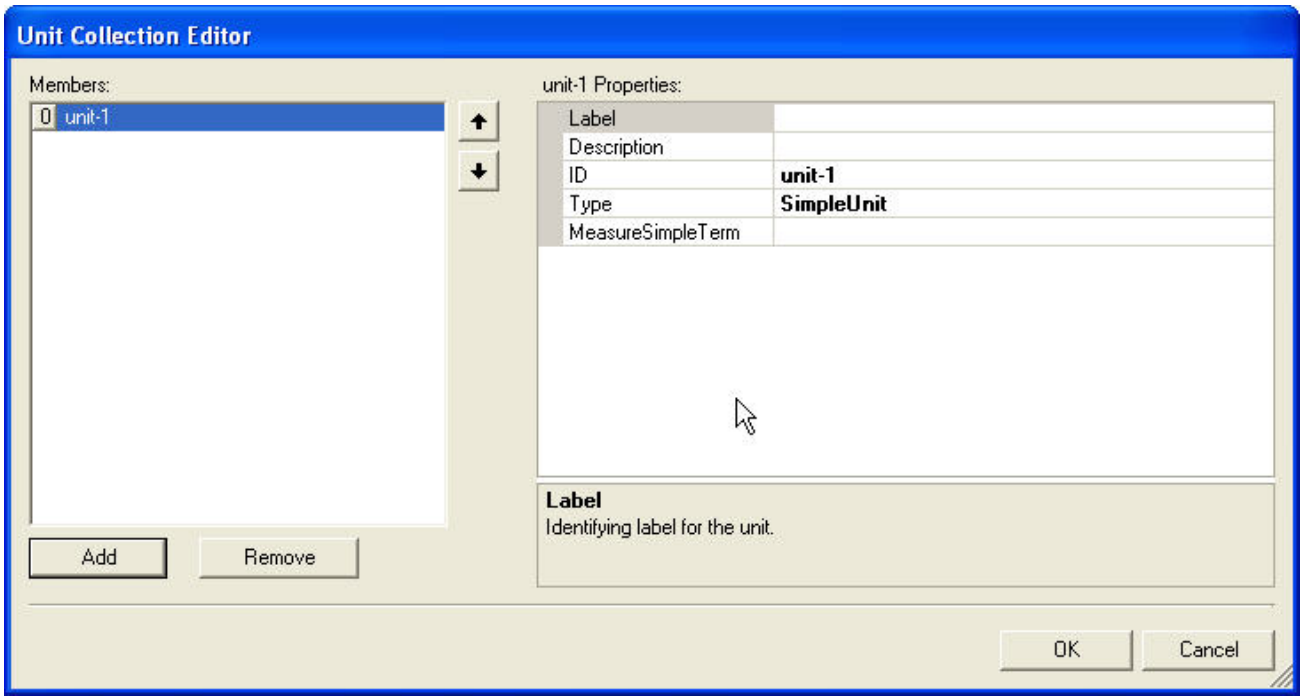
Now that we populated the instance document with fact values, we must create units and decimals for those fact values.

➤ To add units and decimals to the instance:

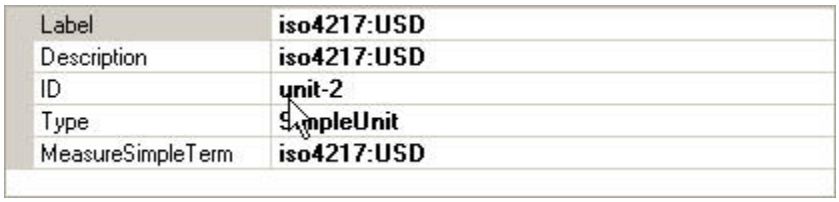
1 Open the Unit Collections editor.



2 Click **Add**.



3 In the ID value field, enter **units-monetary**.



4 Click MeasurementSimpleTerm field to select it and select **iso4217:USD**.

5 Click **OK**.

6 Click the top row of the Unit column to activate a drop-down list, and then select **iso4217:USD**.

7 Right-click Unit column header and select **Fill**, then **Down** to populate the Unit column with the units-monetary value.

Sequence /	Sou	Element	Context	P	D	Unit	IsNil	Value	Effective Value	Co	Foot
1	4	samp:Checking	context-1			iso4217USD	Fa...	-1416.35	(1,416.35)	Fa...	(n...
2	5	samp:Savings	context-1			iso4217USD	Fa...	14368.42	14,368.42	Fa...	(n...
3	6	samp:TotalChecki...	context-1			iso4217USD	Fa...	12952.07	12,952.07	Fa...	(n...
4	8	samp:AccountsRec...	context-1			iso4217USD	Fa...	78320.45	78,320.45	Fa...	(n...
5	9	samp:TotalAccoun...	context-1			iso4217USD	Fa...	78320.45	78,320.45	Fa...	(n...
6	11	samp:ToolsEquip...	context-1			iso4217USD	Fa...	5000	5,000	Fa...	(n...

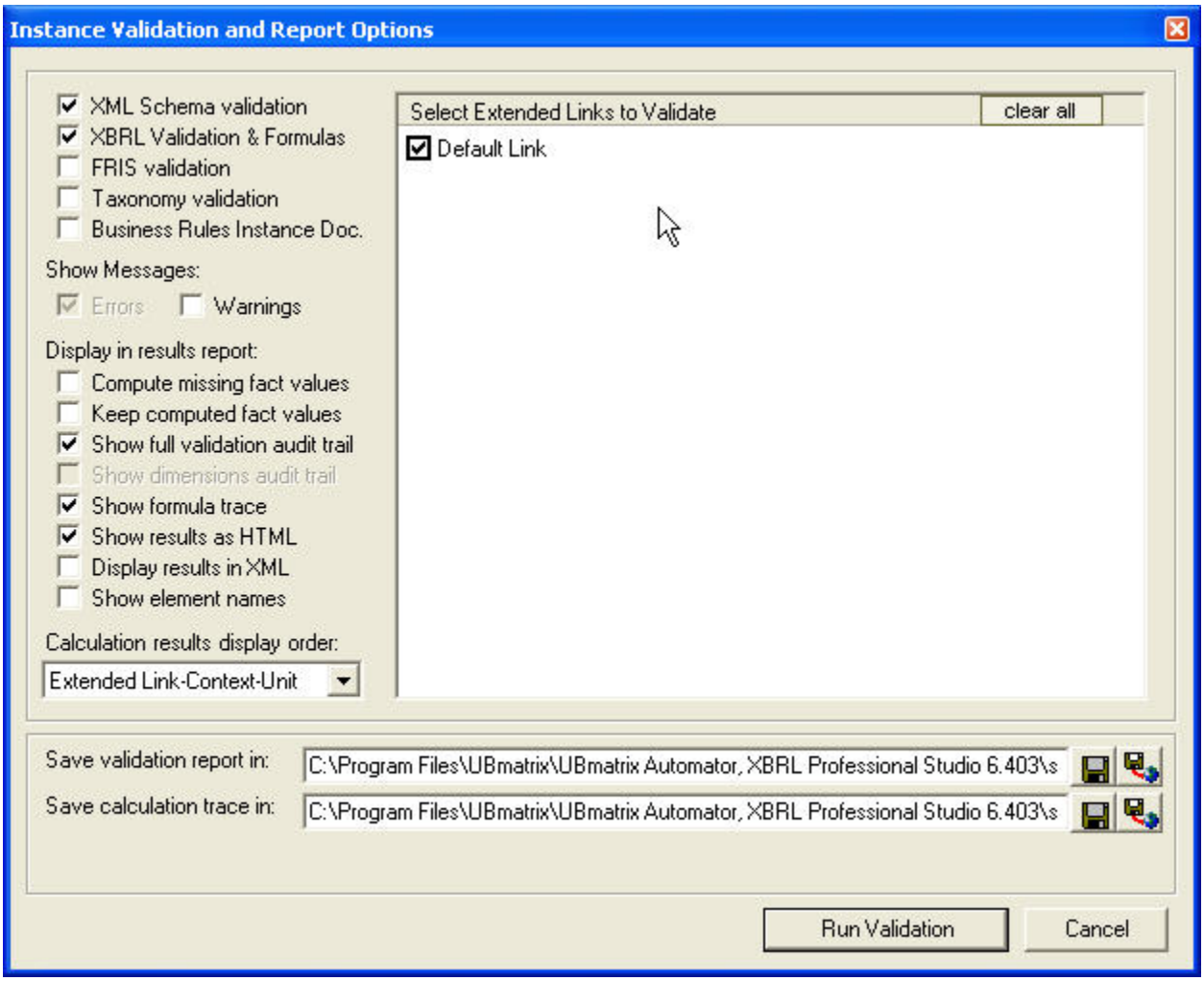
8 **In the D (Decimals) column (if this column is not displayed, right-click anywhere within the factValues window and select Show, then Detail Columns), enter a 2 in the first row.**

This configures XBRL Taxonomy Designer to allow decimals up to two integers, which is standard when dealing with money.

9 **Select Fill, then Down to populate the D column with the value of 2.**

➤ To validate the instance document:

1 **Select Tools, and then Validate Instance Document.**



2 **Select XML and XBRL validation.**

3 **Click Run Validation.**

A calculation trace validation is displayed in another window. If you followed the directions, it should show no errors as below.

Line	Label	W	B	D-P	Value	Source	Message	Formula Trace
1	***Default Link*** **[http://www.xbrl.org/2003/role/link]**							
2	**Context *context-1*[at 2003-12-31 for SAMP]**							
3	**iso4217:USD**							
4	**TOTAL ASSETS**			**D 2**	**190,092.21**	**both**	**OK**	
5	Total Current Assets	1		D 2	159,239.30	both	OK	
6	Total Checking/Savings	1		D 2	12,952.07	both	OK	
7	Checking	1		D 2	(1,416.35)	inst		
8	Savings	1		D 2	14,368.42	inst		
9	Total Accounts Receivable	1		D 2	78,320.45	both	OK	
10	Accounts Receivable	1		D 2	78,320.45	inst		
11	Total Other Current Assets	1		D 2	67,966.78	both	OK	
12	Tools & Equipment	1		D 2	5,000.00	inst		
13	Inventory Asset	1		D 2	23,522.64	inst		
14	Retainage	1		D 2	2,461.80	inst		
15	Undeposited Funds	1		D 2	36,982.34	inst		
16	Total Fixed Assets	1		D 2	30,852.91	both	OK	
17	Total Trucks	1		D 2	30,852.91	both	OK	
18	Original Cost	1		D 2	39,852.91	inst		
19	Depreciation	1		D 2	(9,000.00)	inst		
20	**TOTAL LIABILITIES & EQUITY**			**D 2**	**190,092.21**	**both**	**OK**	
21	Total Liabilities	1		D 2	48,882.63	both	OK	
22	Total Current Liabilities	1		D 2	39,779.42	both	OK	
23	Total Accounts Payable	1		D 2	31,364.38	both	OK	
24	Accounts Payable	1		D 2	31,364.38	inst		
25	Total Credit Cards	1		D 2	141.02	both	OK	
26	Total Company Credit Card	1		D 2	70.00	both	OK	
27	Dept #1	1		D 2	45.00	inst		

4 **Click the X in the upper right of the window to close the report.**

➤ To save an instance document:

1 **Select File, and then Save As File.**

2 **Save the instance document as** `sampleID.xml`.

Note: For ease of use, it is recommended that you save all files for this lesson in `samples\tutorial\mapping\Sample Company\WorkBook` folder to keep track of the file for use in future lessons. The `Workbook` folder has added to the mapping folder to provide a place to save documents without overwriting the sample files.

3 **Select File, and then Close to close.**

Mapping Lesson Two

In this lesson we map the company-specific `SampleCompany.xsd` taxonomy created in Mapping Lesson One to the standardized USGAAP taxonomy. Both taxonomies are included in `/samples/xbrl/usgaap` and `/samples/tutorial/mapping`. Note that GAAP stands for "Generally Accepted Accounting Principles".

Generally this approach is used to map between taxonomies or taxonomy versions. This enables you to use the mapping to convert instance documents that were created against another taxonomy or version of a taxonomy. Next, we build a mapping file between the Sample Company taxonomy and the corresponding US-GAAP. This lesson utilizes the XBRL Mapper feature and maps the elements by dragging from the US-GAAP taxonomy calculation tree and dropping onto the calculation view of the Sample Company.

After the Sample Taxonomy is mapped to the USGAAP, we create a XBRL instance document by converting the XBRL instance generated in lesson one which was built against SampleCompany.xsd. Then we create an instance document built upon USGAAP, but using the values expressed in the sample company excel source file.

Note: Completed files created from this lesson can be found at `samples\tutorial\mapping\Mapping2\`.

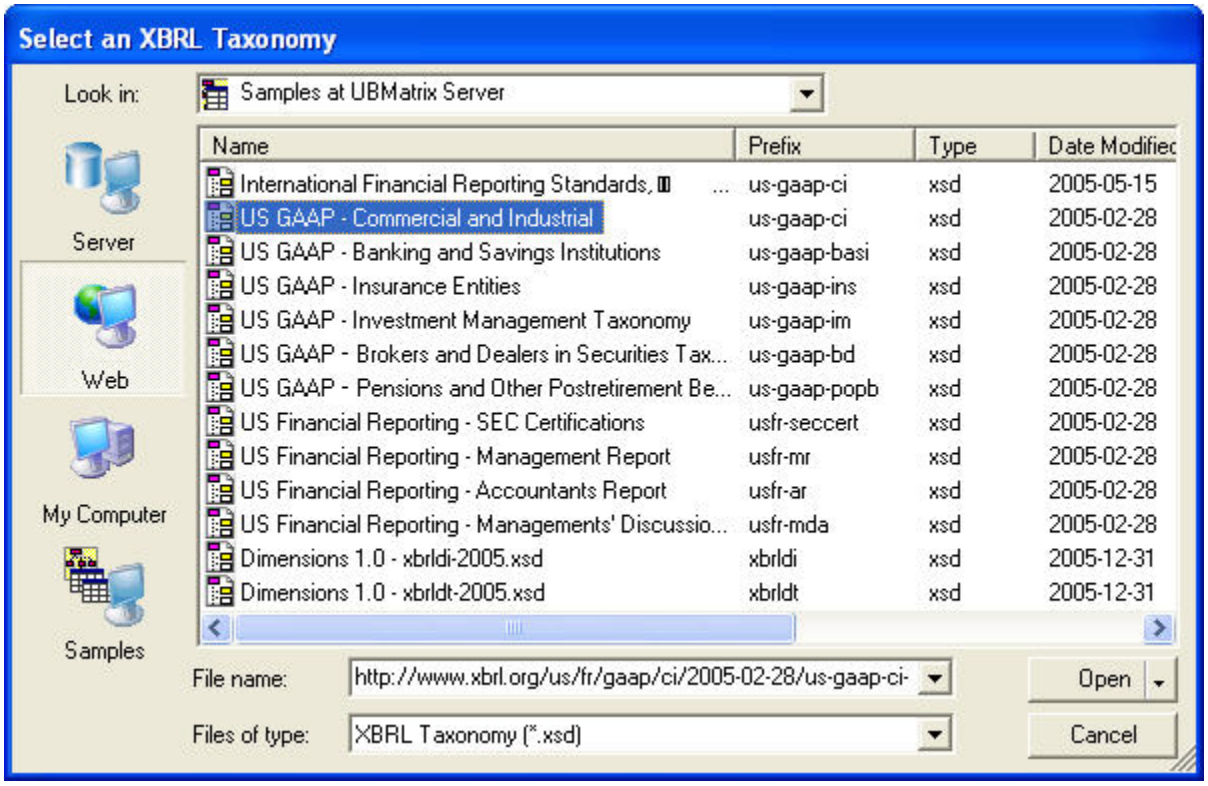
When working with a locked taxonomy such as the USGAAP, XBRL Taxonomy Designer locks the taxonomy from being altered. For information on removing this lock, see "Enabling Editing" on page 25.

Before you start, make sure all taxonomies and instance documents from the previous lesson are closed.

➤ To map the sample taxonomy to US-GAAP taxonomy:

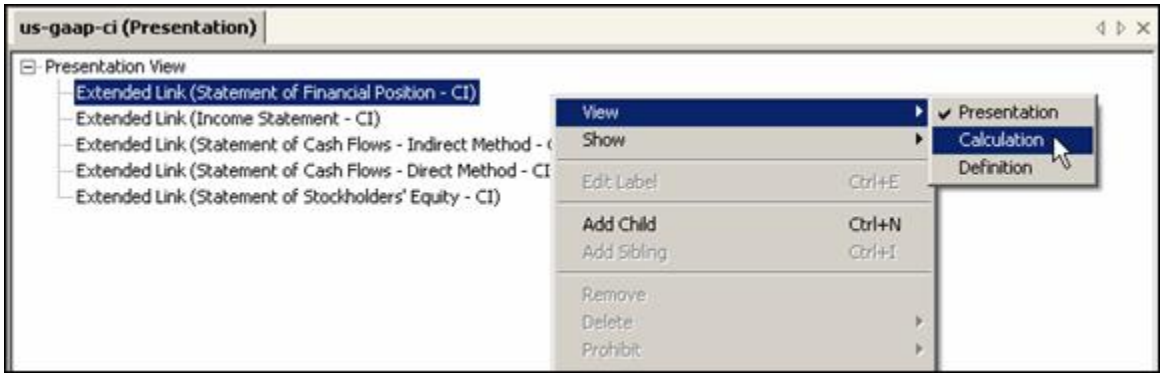
1 **Open the latest US GAAP-Commerical and Industrial taxonomy.**

The fastest method is to use the open a taxonomy using the Web option. Alternatively, you couple browse to the location where the XBRL Taxonomy Designer files were unzipped and select the `us-gaap-ci-2005-02-28.xsd` file.

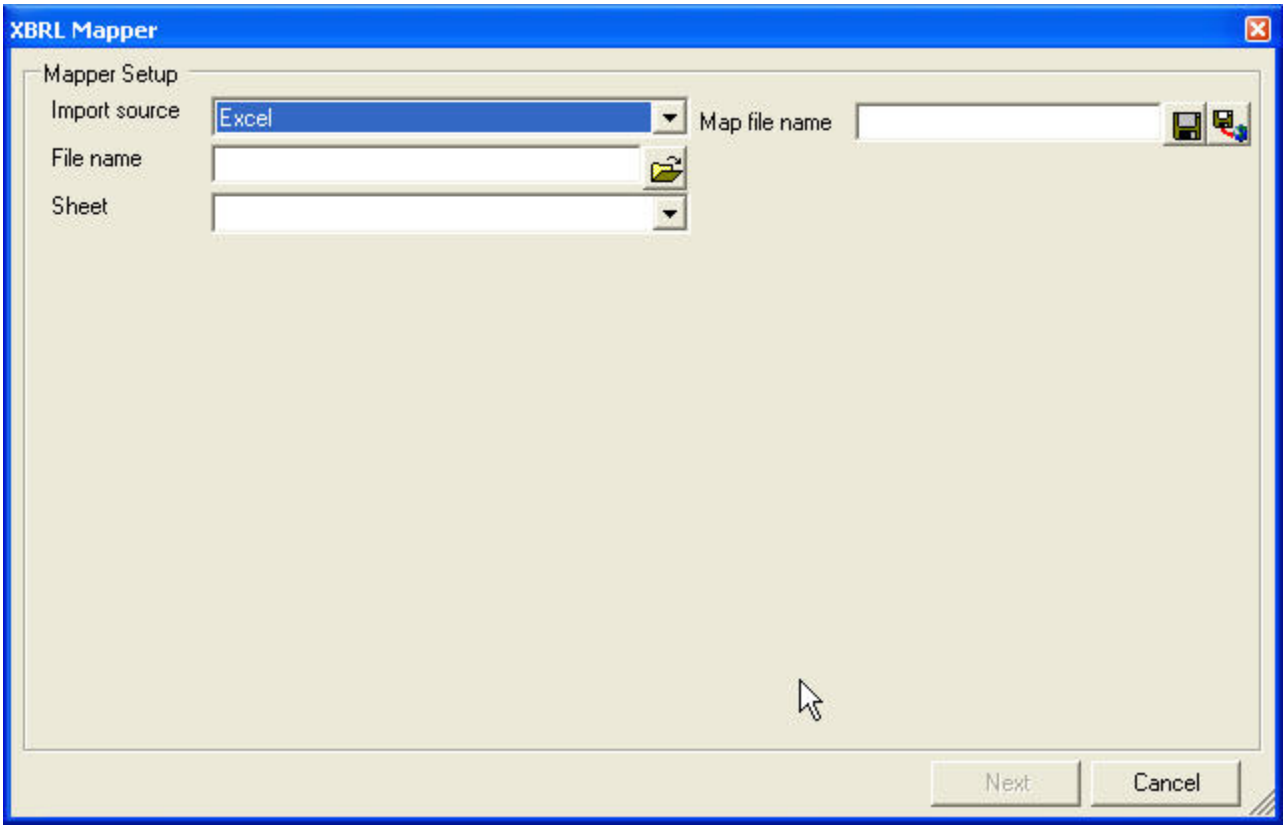


We now map the Sample Company taxonomy to the US GAAP taxonomy. Because we are mapping monetary concepts, it is easier to change the view of the taxonomy to the calculation view. These are occasions where you must map a taxonomy using Presentation, Calculation, Definition, and Element list views. In this case, it is simpler to deal with only the calculation relationship view which only displays the concepts which are involved in a calculation in the Balance sheet.

2 Change to the calculation view by right-clicking in the Relationships View pane and selecting **View**, and then **Calculation**.



3 Open the XBRL Mapper by selecting **Tools**, and then **XBRL Mapper**.



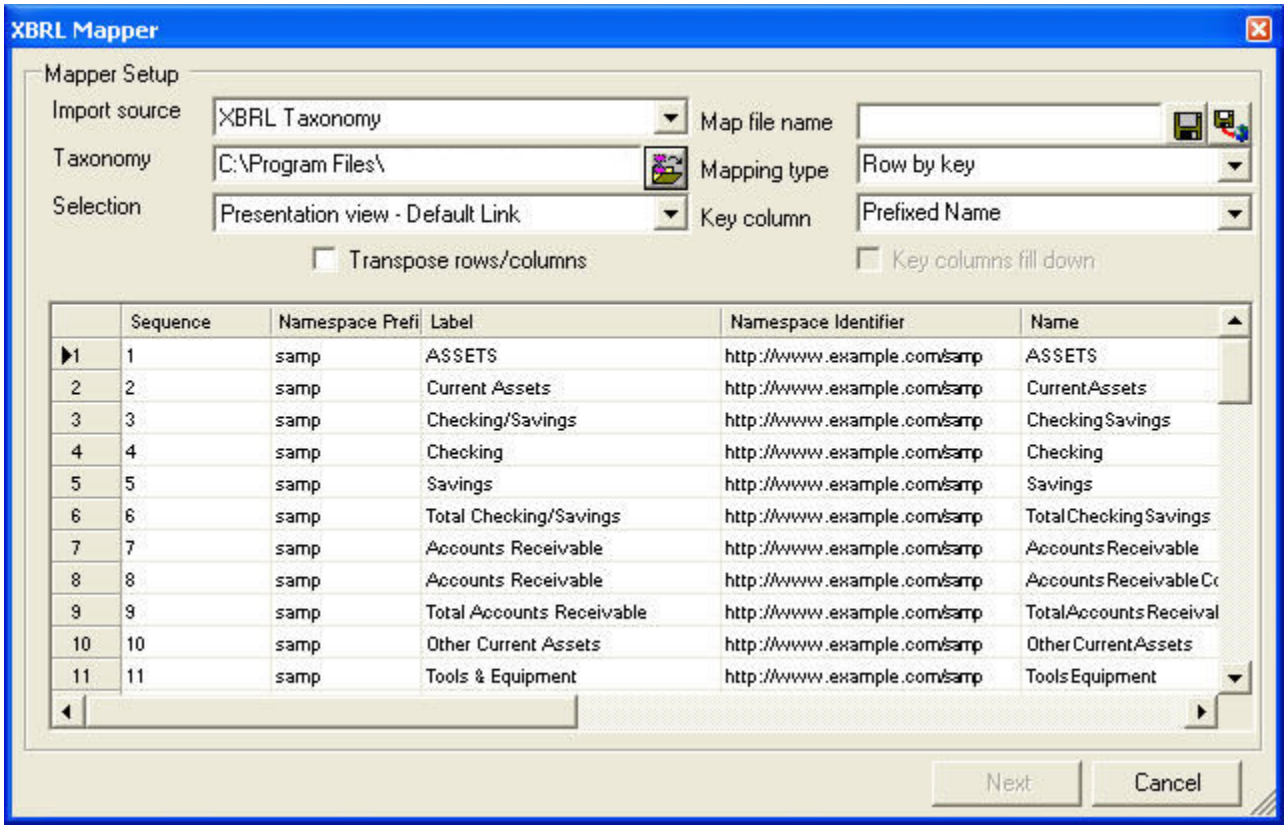
4 **From Import Source, select XBRL Taxonomy.**

The Import Source indicates the source of the import data or original taxonomy to be mapped. See "XBRL Mapper" on page 170.

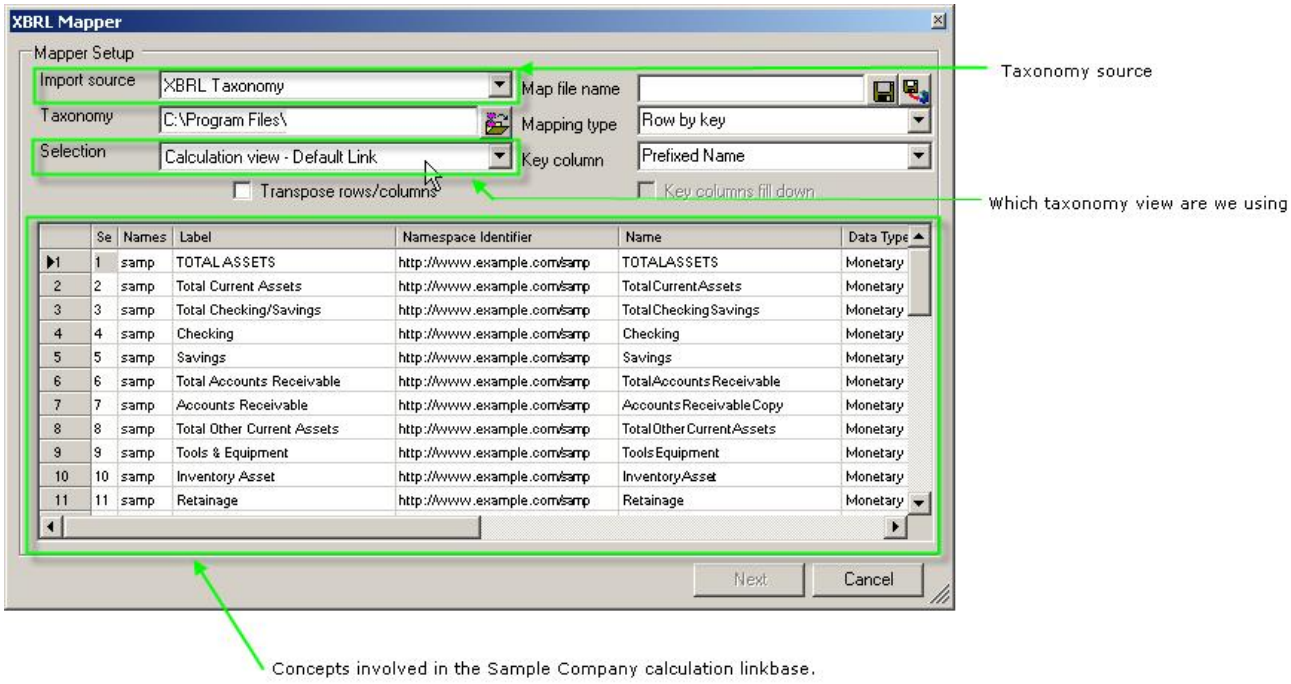
5 **In the Taxonomy field, click the icon to open the Open Taxonomy window.**

6 **Select the `Sample Taxonomy.xsd` file that was created in Lesson One.**

This enables us to map the `Sample Taxonomy.xsd` to the US GAAP-Commerical and Industrial taxonomy.



The values that are expressed in the source document, sample `taxonomy.xsd`, are monetary values. The Calculation View contains all of the monetary concepts. We must change to the Calculation View to map from the US GAAP to the Sample Taxonomy.



Mapping options include:

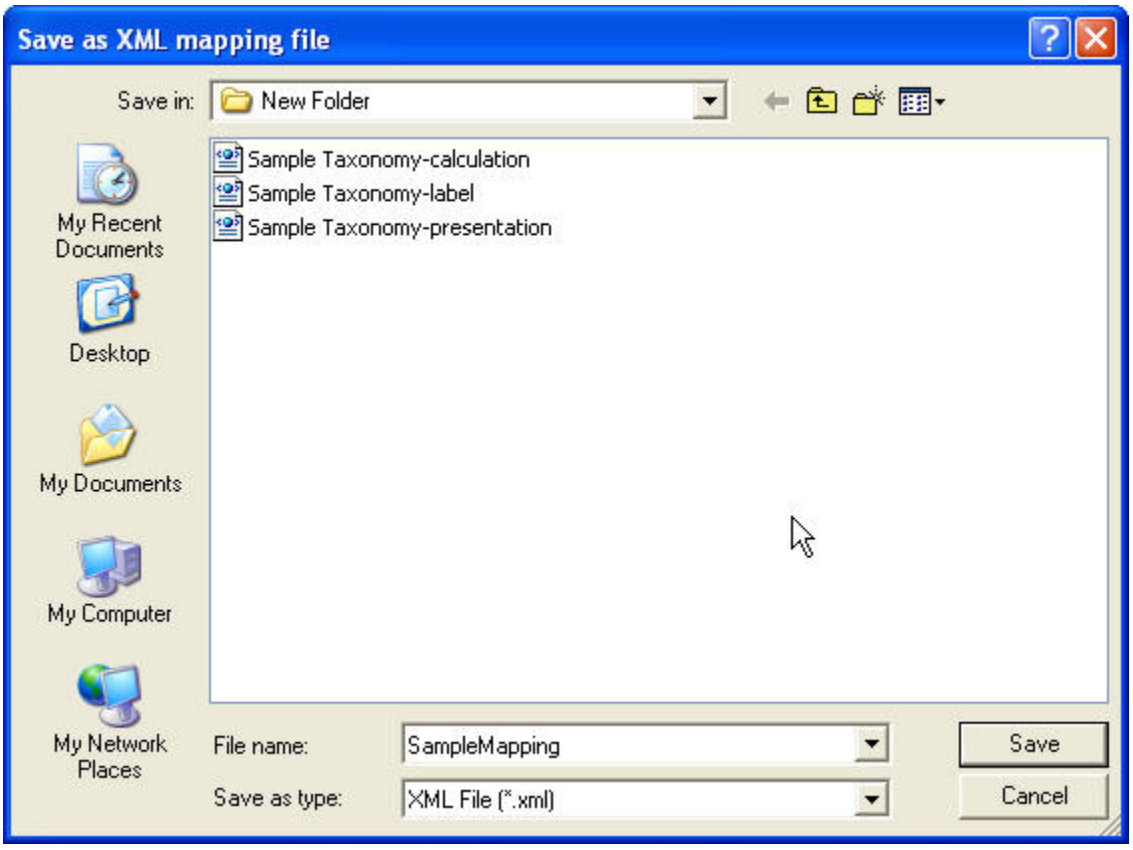
- **Row by key**—Each row or record in the source data pertains to one taxonomy concept. Mapping is by key value (such as account number or label). For taxonomy mapping this is the prefixed name.
- **Row by row number**—Same as above, but row or record is mapped by row ordinal position.
- **Cell by row key and column name**—Each cell may be mapped to a different concept, recognized by row key and column header. Groups data in repeating row groups. Allows future data to contain more rows or columns without regenerating mapping. Pertains to data where adjacent columns are different elements, such as cash flow schedules.
- **Cell by cell reference**—Same as above, but each cell identified by reference, such as A3 or C9.

  In this case, we mapped row by key because we wanted to map the concept and then its row. The reason for this is that we want to stated on which row the concept exists. Now, when we generate the instance, the fact value also resides on the same row.

7 **In the Selection field, select Calculation view – Default Link.**

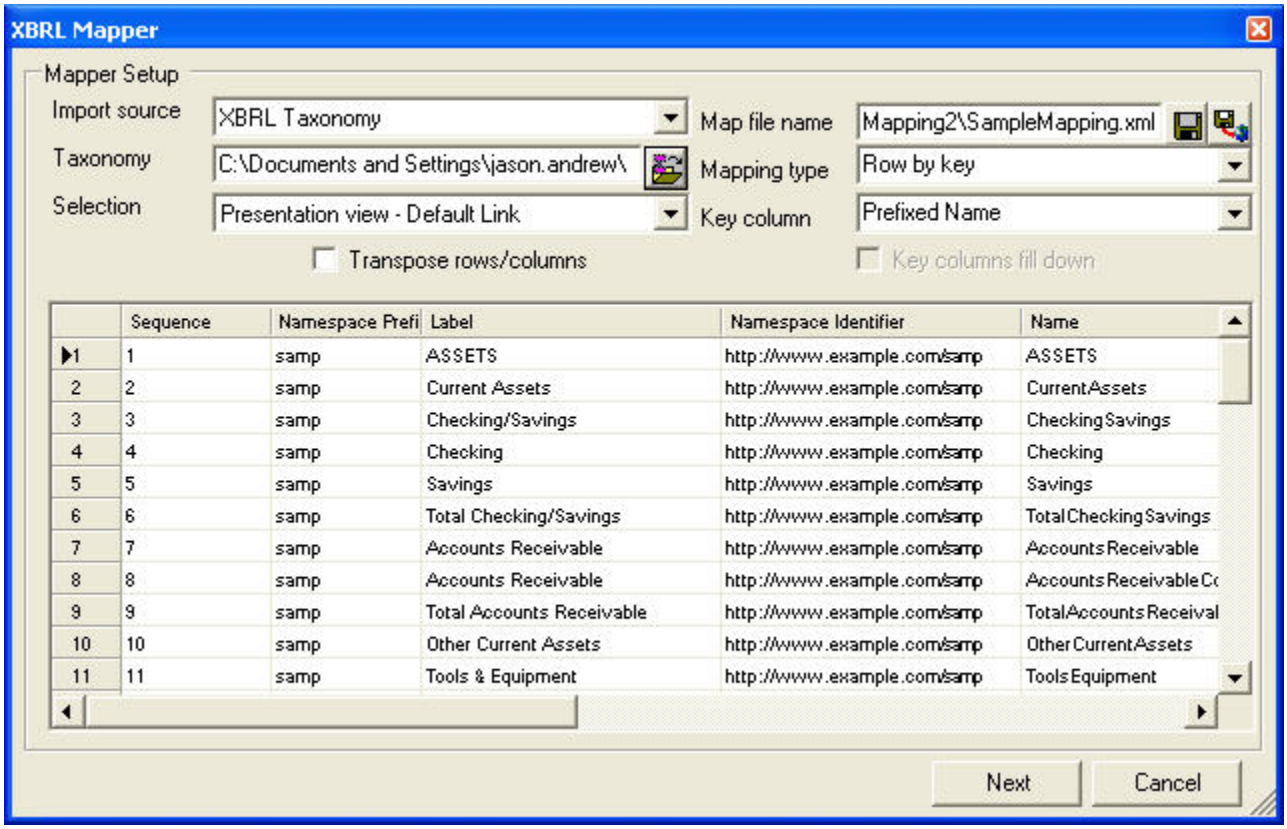
See "XBRL Mapper" on page 170.

8 **In the Map file name field, click .**

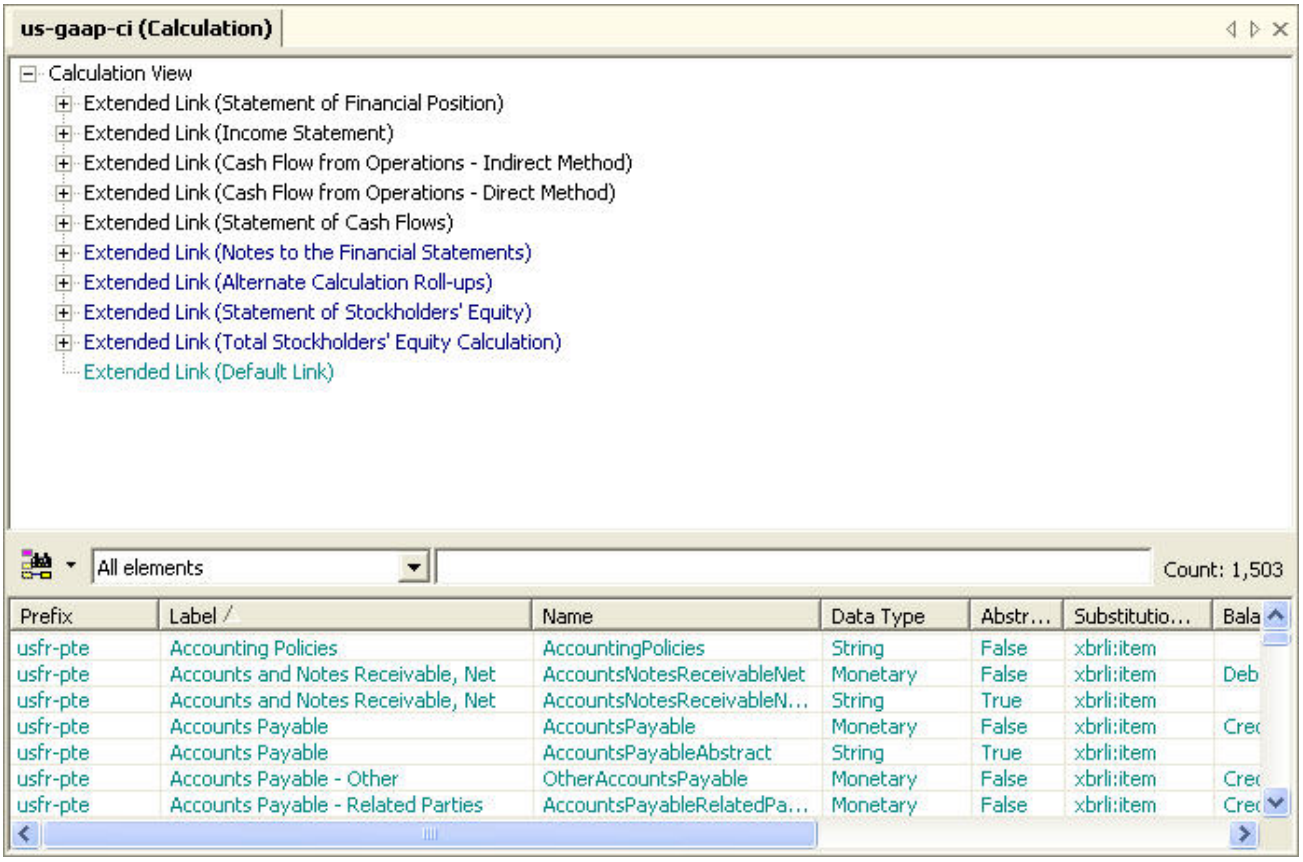


Now we must create an XML mapping output file. This file contains the mapping information used by XBRL Taxonomy Designer.

9 **In the File name field, enter SampleMapping and click Save.**



10 Click **Next** at the bottom of the XBRL Mapper dialog to load the data into the Taxonomy Editor and close the window.

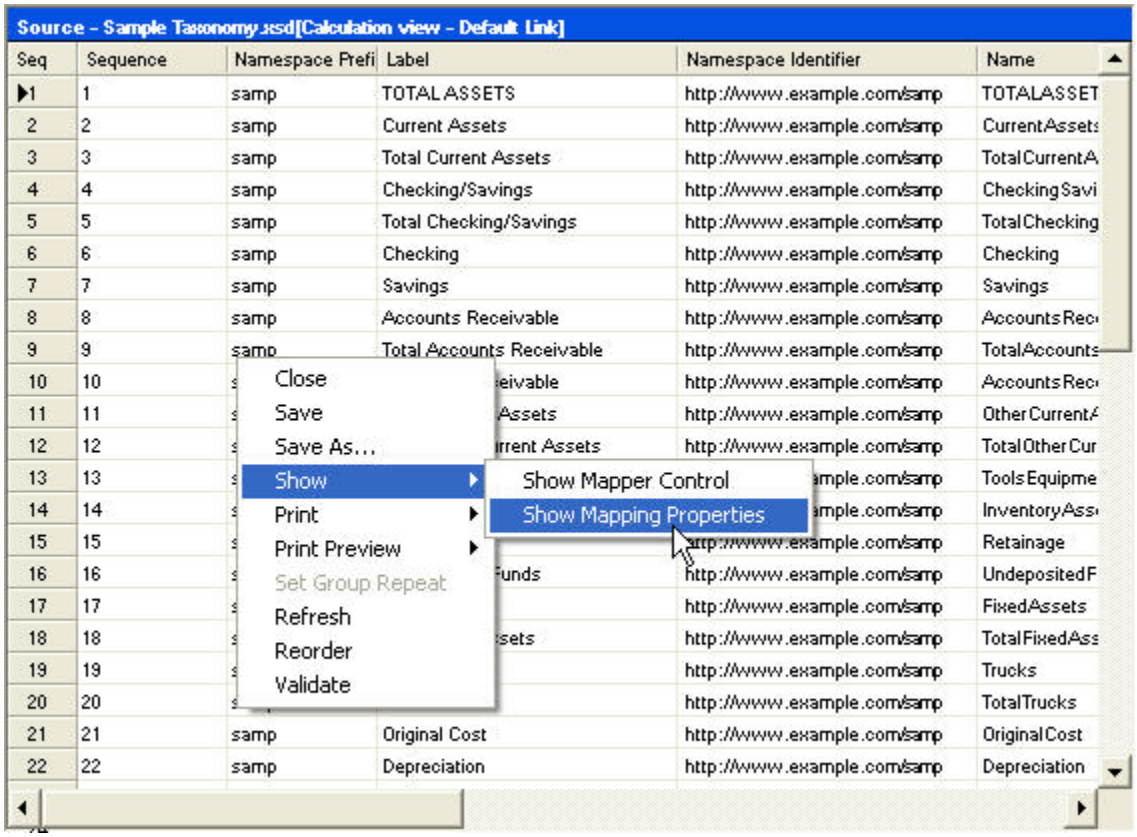


The US-GAAP taxonomy is displayed in the Relationships View pane and the Sample Taxonomy is displayed in the Mapper Grid. Now we can map elements by dragging from the US-GAAP taxonomy calculation tree and dropping onto the calculation view of the Sample Taxonomy. Each time we do this, we are mapping a link from one taxonomy to another. When a link is created, the field of that context turns pink.

We want to adjust the sizes and positions of the Relationships View pane and the Mapper Grid for ease of the drag-and-drop process.

Note: If you have trouble making changes, it might be because XBRL Taxonomy Designer has set the US-GAAP as a read only taxonomy. To fix this, see "Enabling Editing" on page 25.

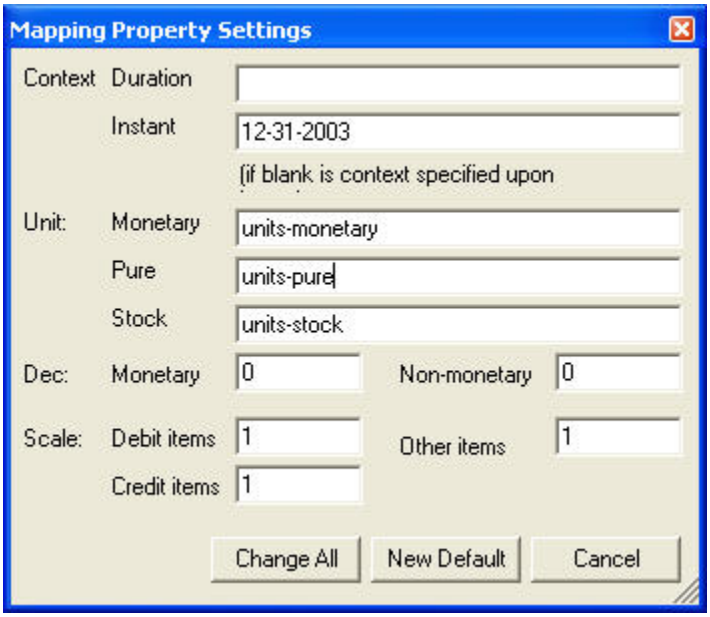
11 **Open the Mapping Properties Window by right-clicking the Mapping Grid and selecting Show, and then Show Mapping Properties.**



12 **Define the context ref to use when mapping: enter samp in the Instant field.**

This may need to be changed several times if there are multiple kinds of context involved, but in this example or lesson there is only one.

Note: The current window is incorrect and lists Context Instant and will be correct in the next build.



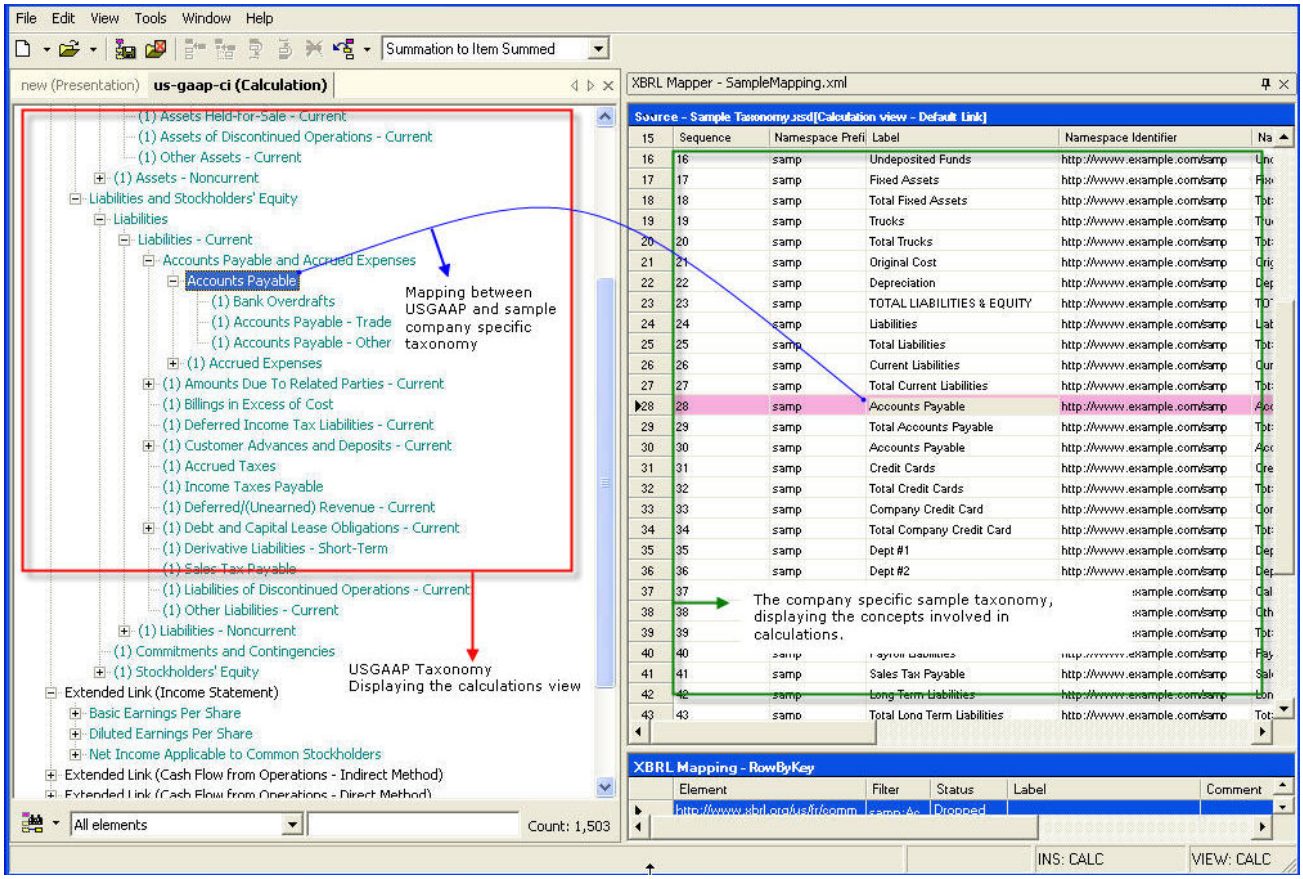
13 **Define the units of measurement to use for each mapping: select units-monetary.**

Each time you drag a drop to create a mapping then this property is used.

14 **Click Change All to change the settings of all of the mappings.**

15 **Click the extended link, Statement of Financial Position, then the + sign to expand Liabilities and Stockholders' Equity in the US-GAAP Calculation View pane.**

Continue to expand until you find the Accounts Payable element as in the figure below.



16 **Left-click and hold on the Accounts Payable element in the US-GAAP taxonomy and drag it into the Mapper Grid until the + sign on the mouse curser is in the AccountsPayable row in the Sample Taxonomy.**

17 **Release the mouse button to drop it and establish the mapping relationship between the two elements.**

After the mapping relationship is established the row is highlighted in pink.

Use this drag and drop technique to establish all of the mapping relationships indicated in the table below. We recommend that you complete the mapping manually initially; however the mapping files for this mapping lesson are included in the `Samples` folder for your review. If you wish to skip this step, you can load the mapping information using the XBRL Mapper.

Note that the elements are given in PRE-FIX: NAME format. You can use the column sorting feature to sort by Name column in the Elements List pane and in the Mapper Grid to easily locate the elements; remember to check that the elements you select contain the correct prefix. You can even drag from the Elements List rather than the Relationships View pane, if you find that easier.

In the table below, note that you may need to map the same US GAAP element to multiple Sample Taxonomy elements.

Table 111 Mapping US GAAP Elements to Sample Taxonomy Elements

US-GAAP-CI Taxonomy	SAMPLE TAXONOMY
PRE-FIX: NAME format	COLUMN: NAME format

US-GAAP-CI Taxonomy	SAMPLE TAXONOMY
usfr-pte: Accounts Payable	samp:AccountsPayableCopy
samp:CalOilCard	
samp:Depte1	
samp:Depte2	
usfr-pte: AccountsReceivableTradeGross	samp:AccountsReceivableCopy
usfr-pte:AccumulatedComprehensiveIncome	samp:NetIncome
usfr-pte:EmployeeRelatedLiabilities	samp:PayrollLiablities
usfr-pte:InventoriesNet	samp:InventoryAsset
usfr-pte:LoansLongTermPortion	samp:VehicleLoan
usfr-pte:MachineryEquipment	samp:ToolsEquipment
usfr-pte:NotesPayableLongTerm	samp:NotePayable
usfr-pte:OtherCurrentAssets	samp:TotalOtherCurrentAssets
samp:Retainage	
usfr-pte:OtherCurrentLiabilities	samp:TotalOtherCurrentLiabilities
usfr-pte:OtherEquity	samp:OwnersContribution
samp:OwnersDraw	
usfr-pte:RetainedEarnings	samp:RetainedEarnings
usfr-pte:SalesTaxPayable	samp:SalesTaxPayable
usfr-pte:UnrestrictedCash	samp:Checking
samp:Savings	
samp:UndepositedFunds	
usfr-pte:Vehicles	samp:Depreciation
samp:OriginalCost	

18 **When finished, save the mapping file by right-clicking in the mapper grid and selecting Save.**

19 **Right-click again and select Close to close the Mapper Grid.**

This saves the mapping links to the mapping file.

Note: For ease of use, it is recommended that you save all files for this lesson in `samples\tutorial\mapping\Sample Company\WorkBook` folder to keep track of the file for use in future lessons. The `Workbook` folder has added to the mapping folder to provide a place to save documents without overwriting the sample files.

20 **Select File, and then Close to close the US GAAP taxonomy.**

Now that we mapped the US GAAP to the Sample Taxonomy file, we can convert an instance document to the US GAAP using the previous data. The sample company taxonomy reposts values for concepts that the USGAAP has but uses different concepts and labels. This lesson shows how you can convert the data for USGAAP and create a mapping between the USGAAP and the sample company taxonomy so someone else can examine how you got to the final instance.

➤ To convert the instance document to US-GAAP taxonomy:

1 **Select File, then New, and then Instance.**

2 **Select File, then Load, and then Taxonomy.**

3 **Open the latest US GAAP-Commercial and Industrial taxonomy.**

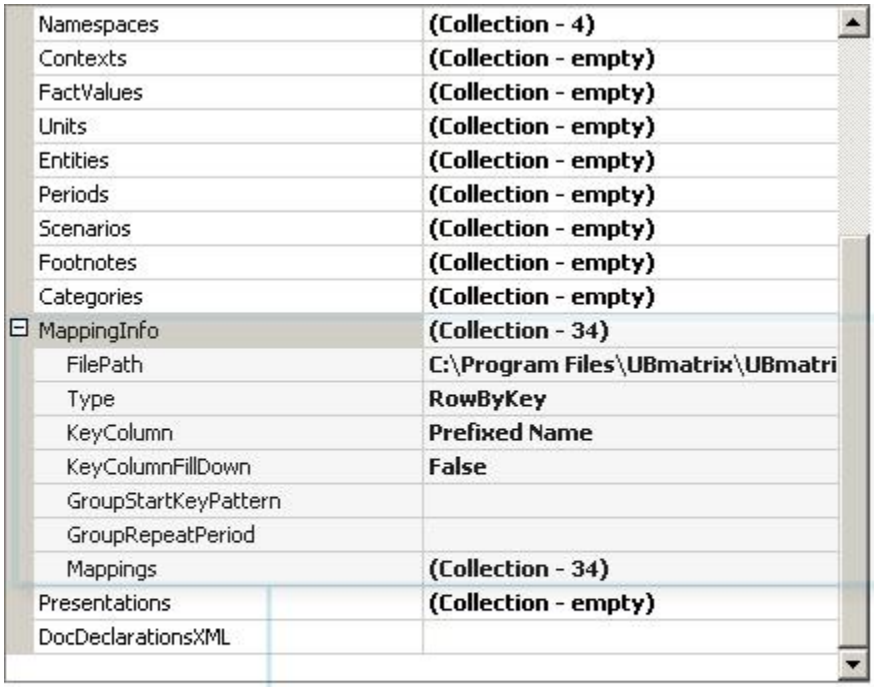
The fastestmethod is to use the open a taxonomy using the Web option. Alternatively, you couple browse to the location that the XBRL Taxonomy Designer files were unzipped and select the `us-gaap-ci-2005-02-28.xsd` file. For a typical default installation of the product, the file path would be: `C:\Oracle\Tutorials\US GAAP 2.1 Taxonomy\`

4 **Select File, then Load, then Mapping, and then From File.**

This loads the mapping XML file produced in Step 3.

5 **Select `SampleMapping.xml`.**

6 **Verify that mappings are loaded by checking that there are entries in the Mapping Info collection in the Instance Properties grid.**



Screenshot of the mapping file properties loaded into an instance document prior to completing an import.

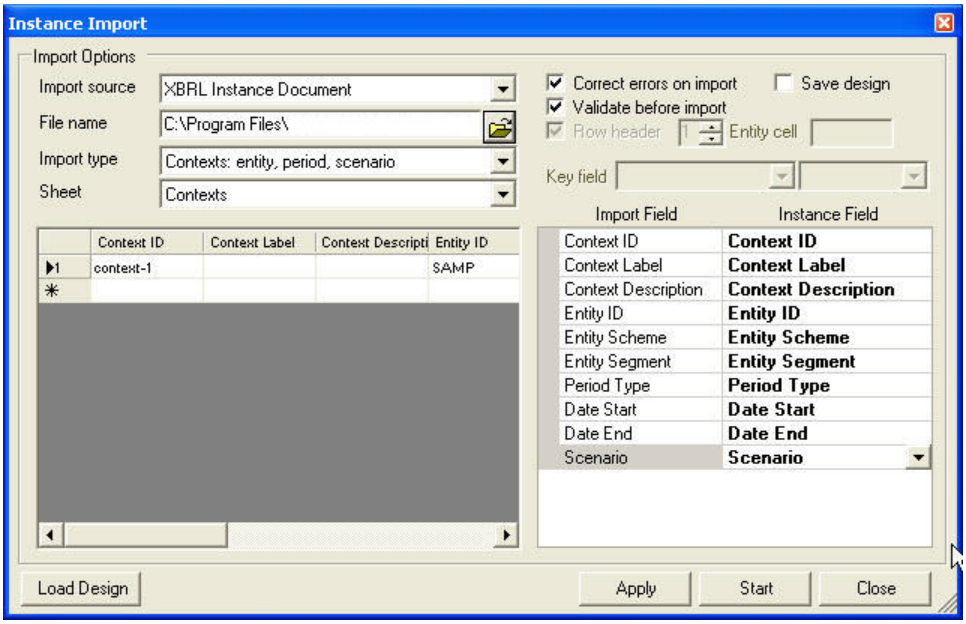
Now we import the instance document created in Example 1 that was based on the `Sample Taxonomy.xsd` file. Before beginning the import, make sure that the Instance View pane is selected (active) to ensure the correct menus and toolbar buttons are activated.

7 Select **File**, and then **Import**.

8 Select **Import Source as XBRL Instance Document**.

9 Click **Browse** and select the `sampleID.xbrl` file that you created in Lesson One.

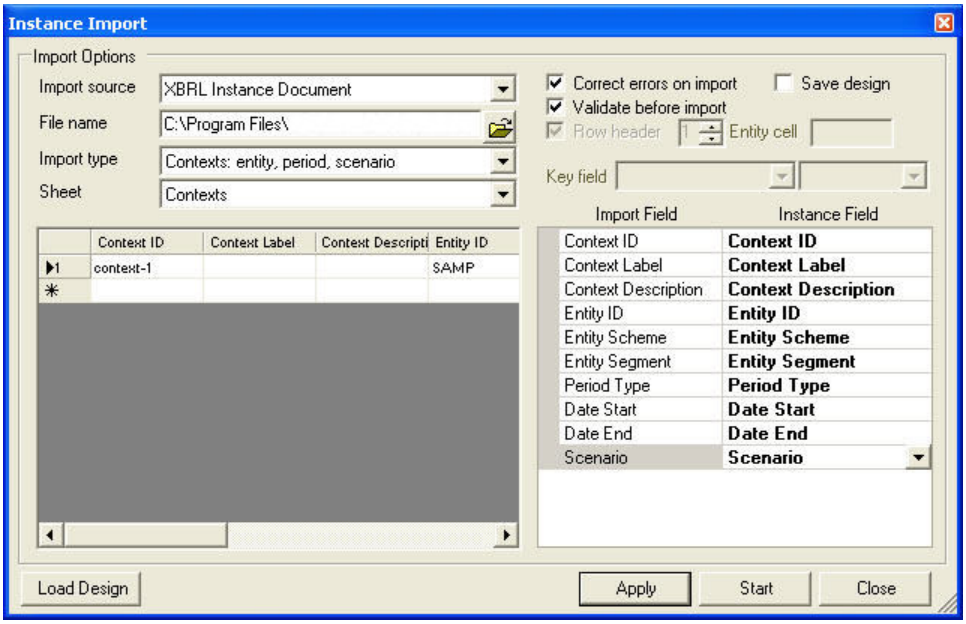
10 Click **OK** to select and close the window.



We first import contexts and units, and then map the fact values.

11 **Select Import Type Contexts, which auto-selects the contexts of the original instance document, and then click Apply.**

The Context collection should indicate members.



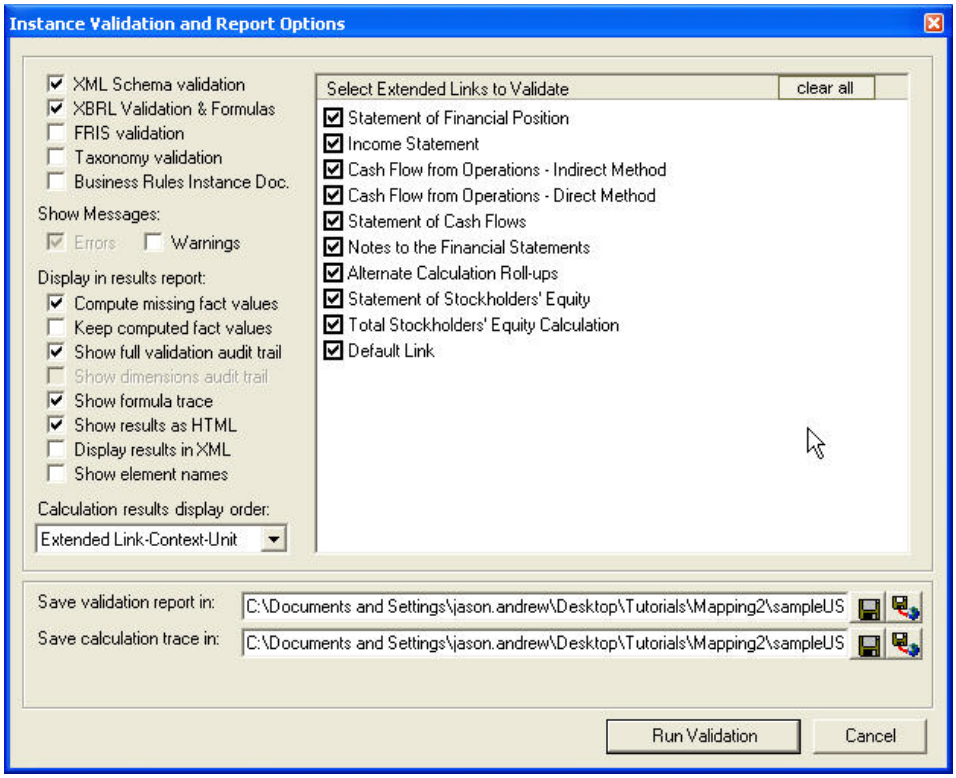
12 **Select Import type Units, which auto-selects the units of the original instance document, and then click Apply.**

13 **Select Import Type Fact Values, and then click Apply.**

There should be fact values according to those original instance taxonomy facts which had mapping specifications.

14 **Click Start to start the import and close the Instance Import window.**

15 Select **Tools**, and then **Validate Instance Document.**

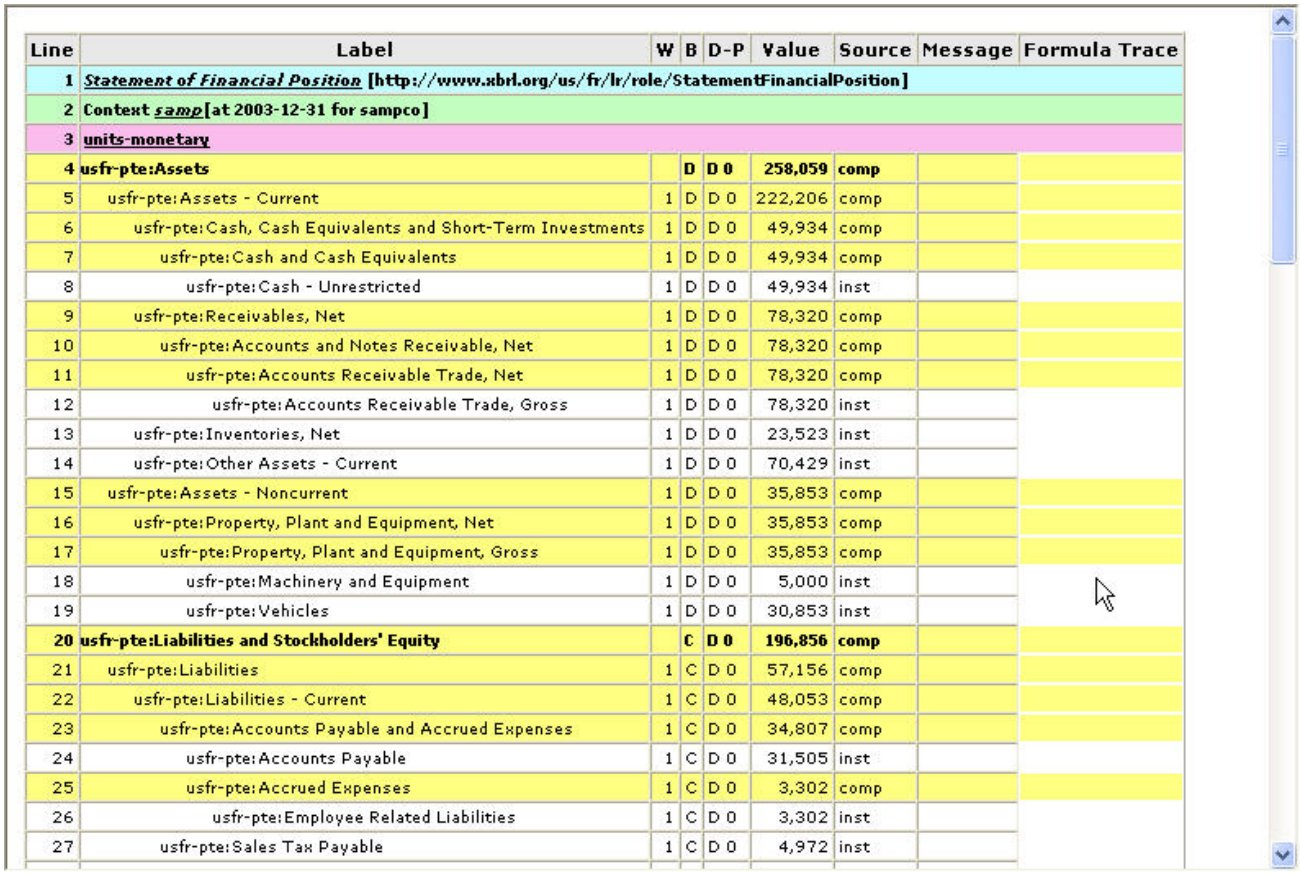


16 Select **XML**, **XBRL validation**, and the **Compute Missing Fact Values** option.

17 Click **Run Validation.**

The simple taxonomy does not contain every item listed in the US GAAP taxonomy, and this option calibrates XBRL Taxonomy Designer to calculate the missing values.

The example validation below has no errors. To see a mapping validation error, see "Troubleshooting Mapping Errors" on page 177.



Line	Label	W	B	D-P	Value	Source	Message	Formula Trace
1	*Statement of Financial Position* [http://www.xbrl.org/us/fr/lr/role/StatementFinancialPosition]							
2	Context *samp* [at 2003-12-31 for sampco]							
3	units-monetary							
4	**usfr-pte:Assets**		**D**	**D 0**	**258,059**	**comp**		
5	usfr-pte:Assets - Current	1	D	D 0	222,206	comp		
6	usfr-pte:Cash, Cash Equivalents and Short-Term Investments	1	D	D 0	49,934	comp		
7	usfr-pte:Cash and Cash Equivalents	1	D	D 0	49,934	comp		
8	usfr-pte:Cash - Unrestricted	1	D	D 0	49,934	inst		
9	usfr-pte:Receivables, Net	1	D	D 0	78,320	comp		
10	usfr-pte:Accounts and Notes Receivable, Net	1	D	D 0	78,320	comp		
11	usfr-pte:Accounts Receivable Trade, Net	1	D	D 0	78,320	comp		
12	usfr-pte:Accounts Receivable Trade, Gross	1	D	D 0	78,320	inst		
13	usfr-pte:Inventories, Net	1	D	D 0	23,523	inst		
14	usfr-pte:Other Assets - Current	1	D	D 0	70,429	inst		
15	usfr-pte:Assets - Noncurrent	1	D	D 0	35,853	comp		
16	usfr-pte:Property, Plant and Equipment, Net	1	D	D 0	35,853	comp		
17	usfr-pte:Property, Plant and Equipment, Gross	1	D	D 0	35,853	comp		
18	usfr-pte:Machinery and Equipment	1	D	D 0	5,000	inst		
19	usfr-pte:Vehicles	1	D	D 0	30,853	inst		
20	**usfr-pte:Liabilities and Stockholders' Equity**		**C**	**D 0**	**196,856**	**comp**		
21	usfr-pte:Liabilities	1	C	D 0	57,156	comp		
22	usfr-pte:Liabilities - Current	1	C	D 0	48,053	comp		
23	usfr-pte:Accounts Payable and Accrued Expenses	1	C	D 0	34,807	comp		
24	usfr-pte:Accounts Payable	1	C	D 0	31,505	inst		
25	usfr-pte:Accrued Expenses	1	C	D 0	3,302	comp		
26	usfr-pte:Employee Related Liabilities	1	C	D 0	3,302	inst		
27	usfr-pte:Sales Tax Payable	1	C	D 0	4,972	inst		

18 Select **File**, and then **Save As** and save this instance document as `sampleUSGAAP-ID.xbrl`.

Note: For ease of use, it is recommended that you save all files for this lesson in `samples\tutorial\mapping\Sample Company\WorkBook` folder to keep track of the file for use in future lessons. The `Workbook` folder has added to the mapping folder to provide a place to save documents without overwriting the sample files.

19 Select **File**, and then **Close** to close the instance document.

You completed Mapping Lesson Two. Proceed to "Mapping Lesson Three" on page 318.

Mapping Lesson Three

We use the `sample-balancesheet.xls` file, but map it directly to US-GAAP instead of to the Sample Taxonomy. Subsequently, we convert an instance document from the Sample Taxonomy to US-GAAP Taxonomy.

Note: Completed files created from this lesson can be found at `samples\tutorial\mapping\Mapping3\.`

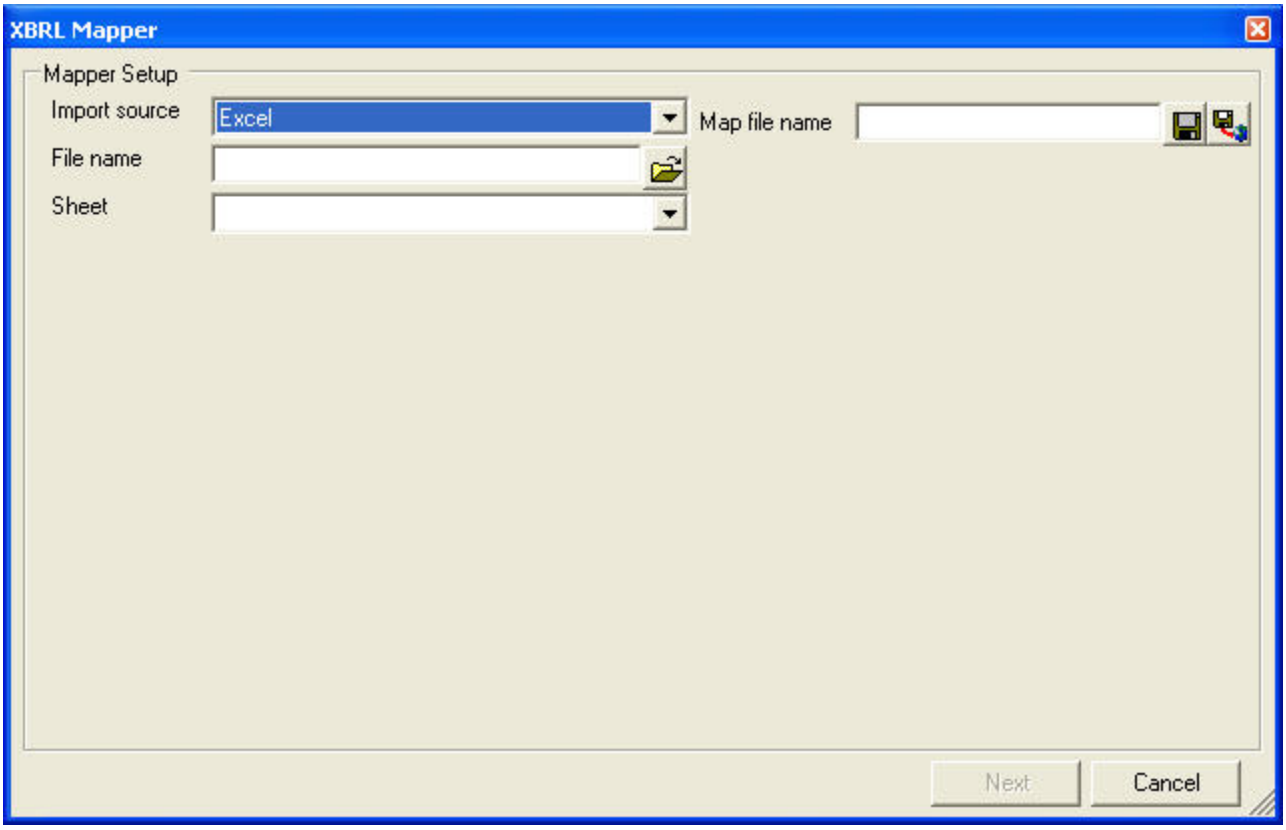
When working with a locked taxonomy such as the USGAAP, XBRL Taxonomy Designer locks the taxonomy from being altered. For information on removing this lock, see "Enabling Editing" on page 25.

➤ To map the balance sheet to the US GAAP taxonomy:

1 **Open the latest US GAAP-Commerical and Industrial taxonomy.**

The fastest method is to use the open a taxonomy using the Web option. Alternatively, you couple browse to the location where the XBRL Taxonomy Designer files were unzipped and select the `us-gaap-ci-2005-02-28.xsd` file.

2 **Select Tools, and then XBRL Mapper.**



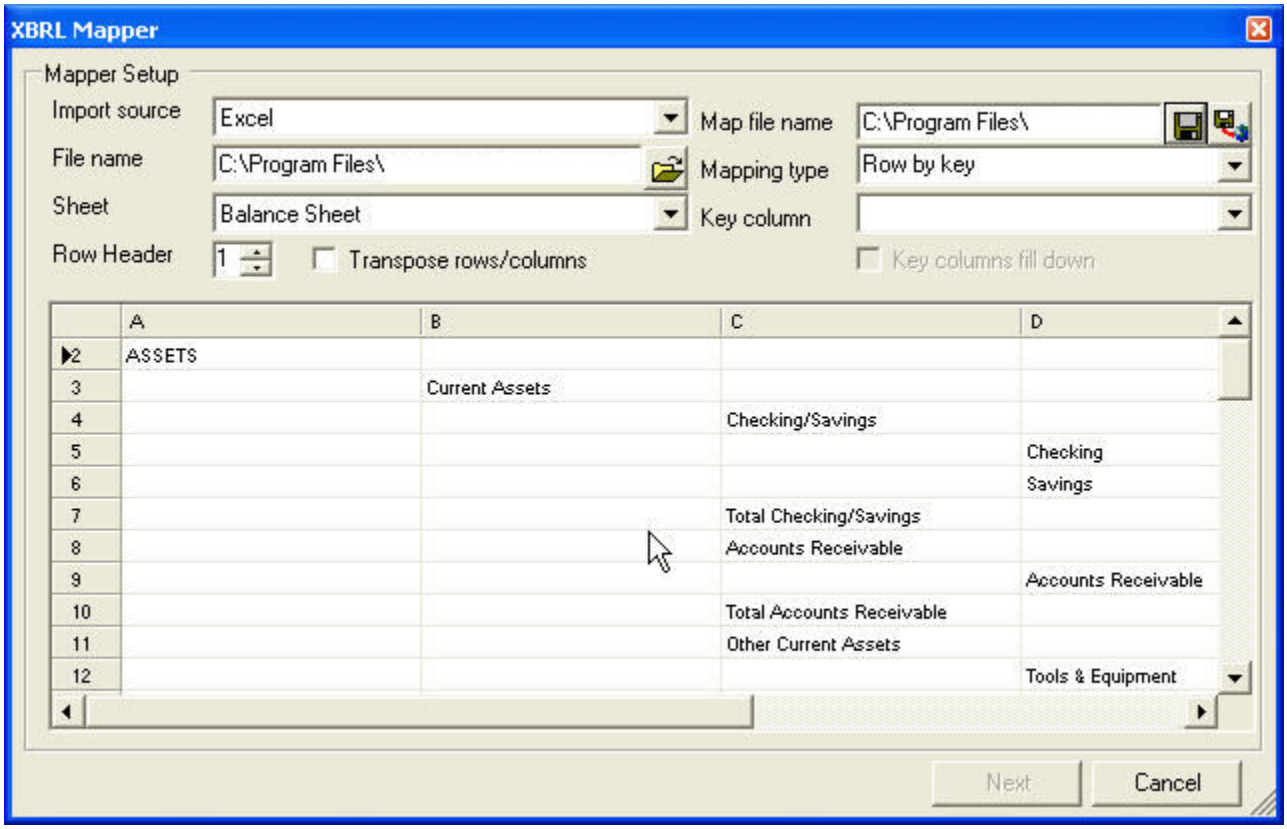
3 **In the XBRL Mapper window, select Excel as the Import source.**

4 **In the File name field, click Browse to navigate to the `Sample` folder, and then select the `sample-balancesheet.xls` file.**

After the sample-balancesheet.xls file selected, the Sheet field defaults to Balance Sheet.

5 **In the Mapping file name field, click Save.**

6 **In the File Name field, enter `SampleMapping2.xml`, and then click Save to save and close the window.**

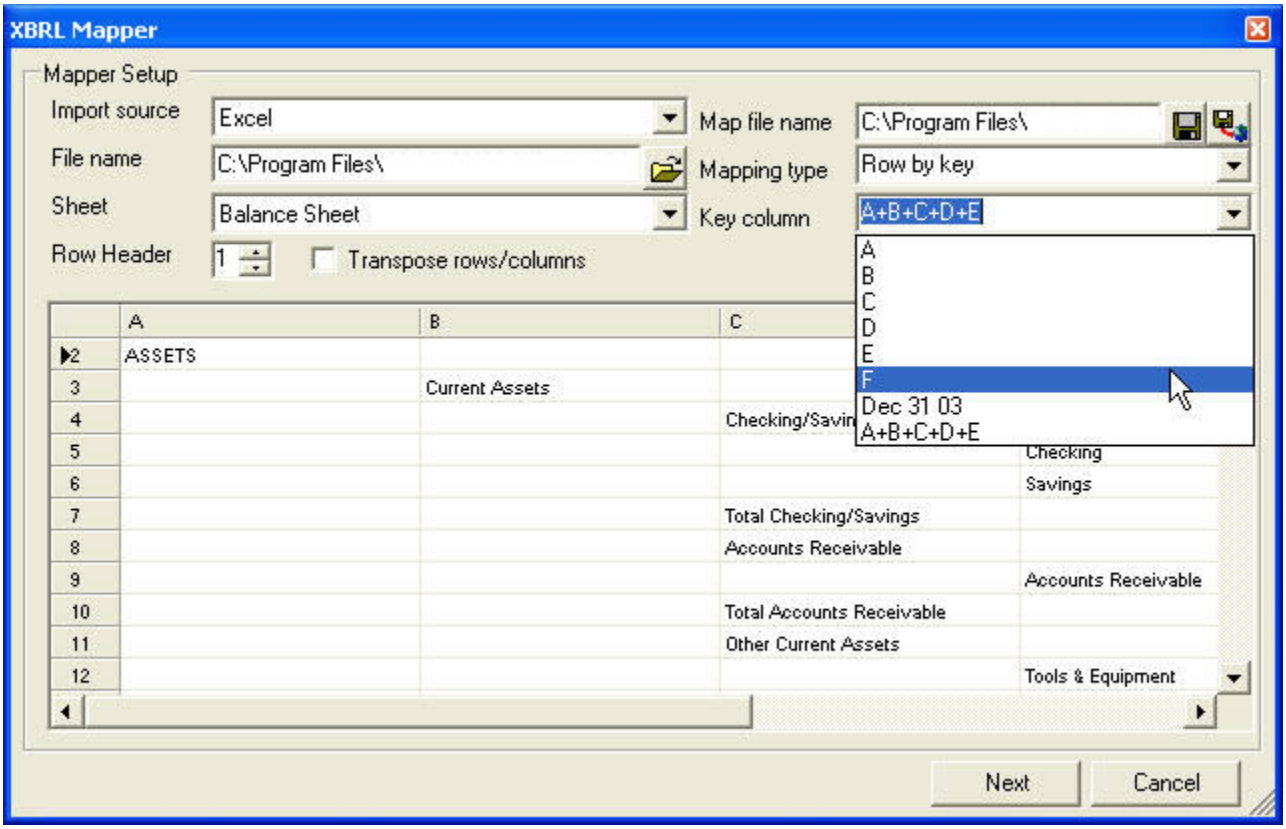


7 **In the Mapping type field, select Row by key.**

Note how the financial concepts are reported within the sample company balance sheet in the example above. They use column indentation, and therefore to ensure that our mapping contains the path from the top level concept to the concept being mapped we enter the columns to be mapped. In other words, the columns A through F are used to search for the concepts that we are mapping to. Specifically for these sample company charts of accounts, a sub-account may only be unique within its nesting structure in a parent account (for example, frequent use of accounts called "Cash" or "Other"). For this reason, we specify all of A through F as the pattern matching keys.

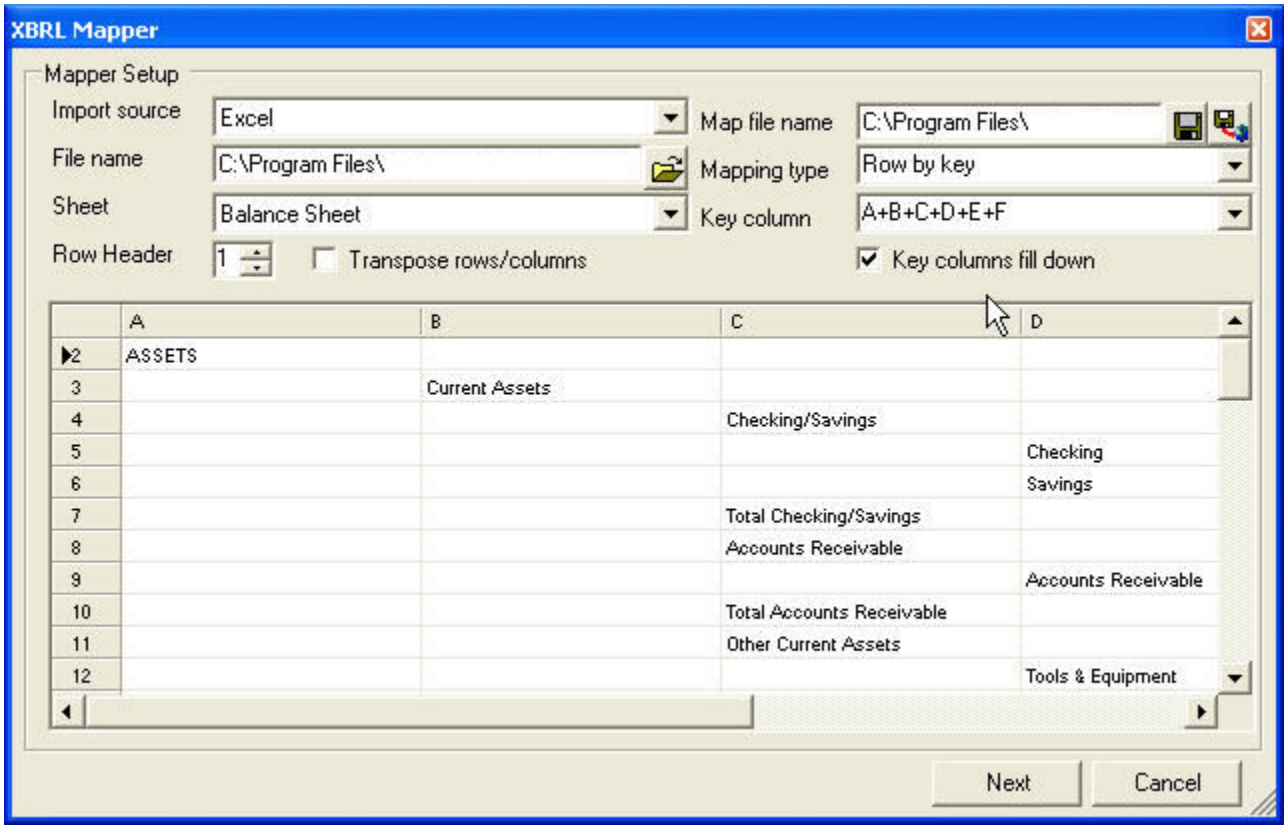
8 **In the Key column field, select A then hold the Ctrl key down while clicking on the drop-down arrow and selecting B, then C, then D, E, and F.**

They use plus signs to separate them generating a composite key matching specification.

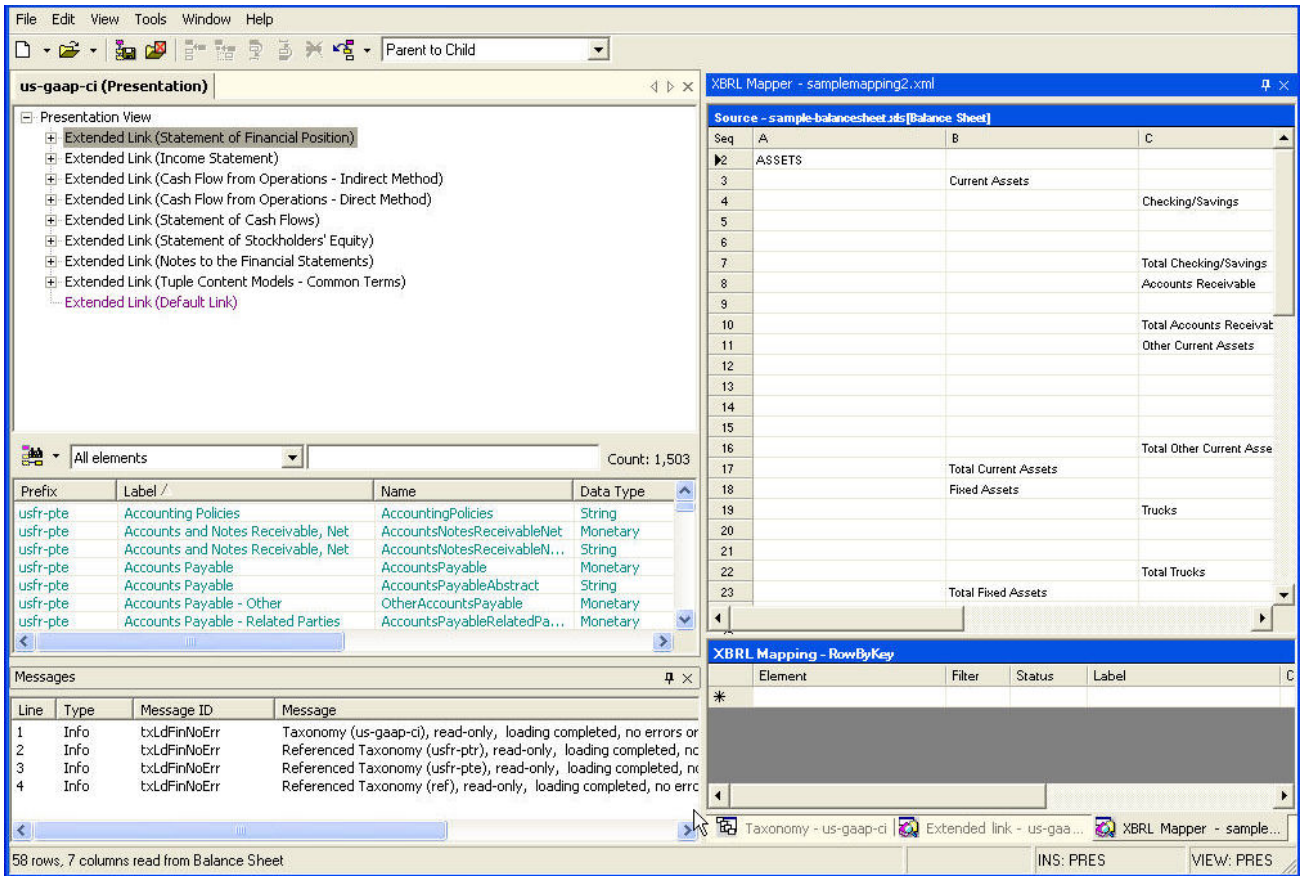


9 **Select the Key columns fill down box.**

This causes each indented column to "fill" from above to eliminate potential ambiguities if sub-accounts are named the same.



10 Click **Next**.



The Mapping Grid is displayed on the right. Again, it might be convenient to resize the mapping components for ease of use.

Next we use the drag-and-drop technique to map from the US GAAP to the Balance Sheet. Again, simply drag a taxonomy element from the Relationships View pane or the Elements List onto the Mapping Grid; a mapping entry is created in the lower-right portion of the screen.

➤ To drag-and-Drop XBRL mapping:

1 **As in Lesson Two, use the drag and drop technique to establish all of the mapping relationships indicated in Table 112.**

Table 112 Mapping Relationships

US-GAAP-CI Taxonomy	Balance Sheet Excel File
PRE-FIX: NAME format	COLUMN: NAME
usfr-pte: Accounts Payable	D Accounts Payable
E CalOil Card	
F Dept1	
F Dept2	
usfr-pte: AccountsReceivableTradeGross	D Accounts Receivable
usfr-pte:AccumulatedComprehensiveIncome	C Net Income
usfr-pte:EmployeeRelatedLiabilities	E Payroll Liabilities
usfr-pte:InventoriesNet	D Inventory Asset
usfr-pte:LoansLongTermPortion	D Vehicle Loan
usfr-pte:MachineryEquipment	D Tools & Equipment
usfr-pet:NotesPayableLongTerm	D Note Payable
usfr-pte:OtherCurrentAssets	C Other Current Assets
D Retainage	
usfr-pte:OtherCurrentLiabilities	D Other Current Liabilities
usfr-pte:OtherEquity	D Owners' Contribution
D Owners' Draw	
usfr-pte:RetainedEarnings	C Retained Earnings
usfr-pte:SalesTaxPayable	E Sales Tax Payable

US-GAAP-CI Taxonomy	Balance Sheet Excel File
usfr-pte:UnrestrictedCash	D Checking
D Savings	
D Undeposited Funds	
usfr-pte:Vehicles	D Depreciation
D Original Cost	

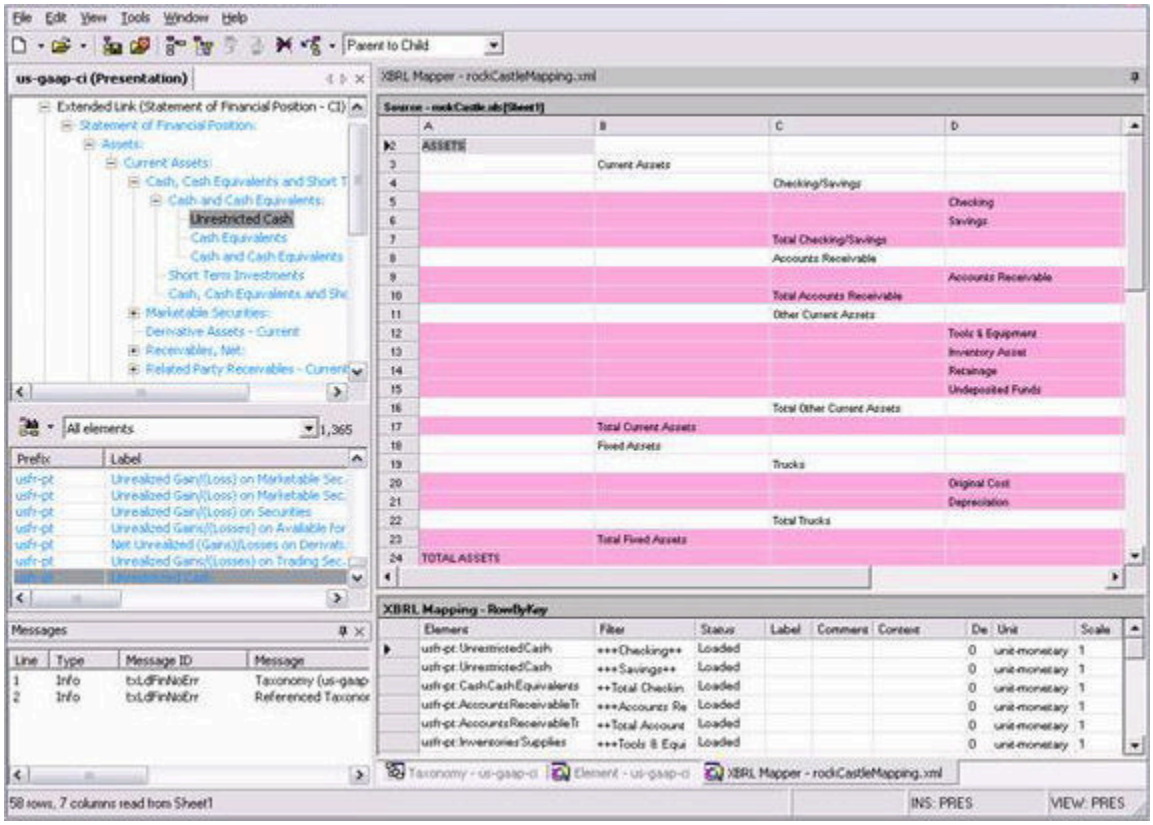
The Filter column represents the Key column pattern for the taxonomy element, context, unit, scale, and so on, of this mapping.

2 **Save the mapping file by right-clicking anywhere in the Mapping Grid and clicking Save.**

3 **To close the mapping grid without closing the taxonomy, right-click in the Mapping Grid and click Close.**

4 **Select File, and then Close to close the taxonomy.**

Note: For ease of use, we recommend that you save all files for this lesson in `Samples/WorkBook` folder so that you can keep track of it and use it for later lessons.



The next step is to create an instance document.

➤ To create an instance document:

1 **Select File, then New, and then Instance.**

2 **Use what you learned so far to load the USGAAP taxonomy.**

The fastest method is to use the open a taxonomy using the Web option. Alternatively, browse to the location that the XBRL Taxonomy Designer files were unzipped and select the `us-gaap-ci-2005-02-28.xsd` file. For a typical default installation of the product, the file path would be: `C:\UBmatrix\Tutorials\US GAAP 2.1 Taxonomy\.`

3 **Load the above-produced mapping file, which should be named `SampleMapping2.xml`.**

Notice that for the MappingInfo property in the Instance Property grid, the number of entries reflects the mappings done in Step 2 above.

As in Lesson One: step five, we need a unit before we can import the data.

4 **Enter iso4217:USD for units-monetary to correspond with the unit selections in the mapping grid before.**

Next we need a Context, so we can import the data.

5 **Follow the directions from Lesson One: Step four.**

6 **Select File, and then Import.**

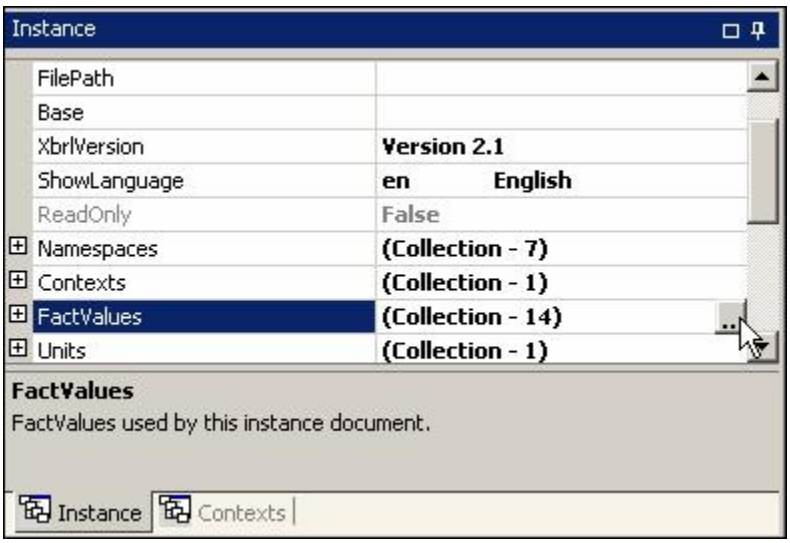
7 **Select Excel as the Import Source and for File name select the `sample-balancesheet.xls` file.**

8 **Using what you learned in Lesson One: Step four, make these changes:**

- Import type is Fact-values
- Import Field A is the mapping key (select Mapping Key from the Instance Field drop-down list.)
- In the Dec 31 03 row, select Value for Instance Field column. Click the + sign to expand the rows and select the period entered when establishing the context

9 **Click Start.**

You should have 14 fact values in an instance document from an Excel spreadsheet.

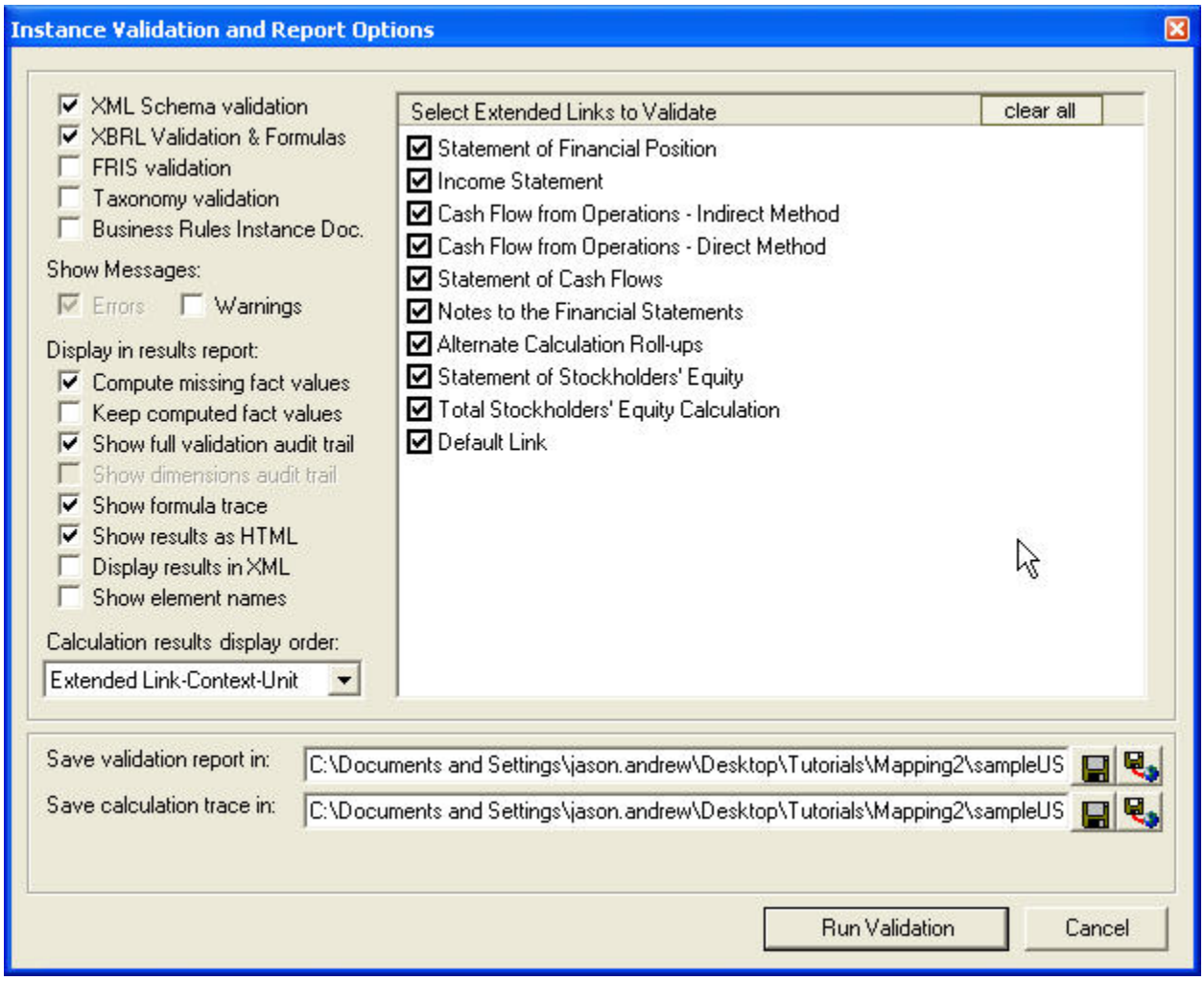


10 **Click the ellipsis (shown in the figure above) to open the editor and view the Fact Values.**

factValues - (New)

Seq	Element	Context	Unit	Value
1	usfr-pt:UnrestrictedCash	context-1	units-monetary	49934
2	usfr-pt:AccountsReceivableTrade	context-1	units-monetary	78320
3	usfr-pt:MachineryEquipment	context-1	units-monetary	5000
4	usfr-pt:InventoriesNet	context-1	units-monetary	23523
5	usfr-pt:OtherCurrentAssets	context-1	units-monetary	2462
6	usfr-pt:Vehicles	context-1	units-monetary	30853
7	usfr-pt:AccountsPayable	context-1	units-monetary	141
8	usfr-pt:EmployeeRelatedLiabilities	context-1	units-monetary	3302
9	usfr-pt:SalesTaxPayable	context-1	units-monetary	4972
10	usfr-pt:NotesPayableLongTerm	context-1	units-monetary	3441
11	usfr-pt:LoansLongTermPortion	context-1	units-monetary	5662
12	usfr-pt:OtherEquity	context-1	units-monetary	29000
13	usfr-pt:RetainedEarnings	context-1	units-monetary	78200
14	usfr-pt:AccumulatedComprehensi	context-1	units-monetary	32500
*				

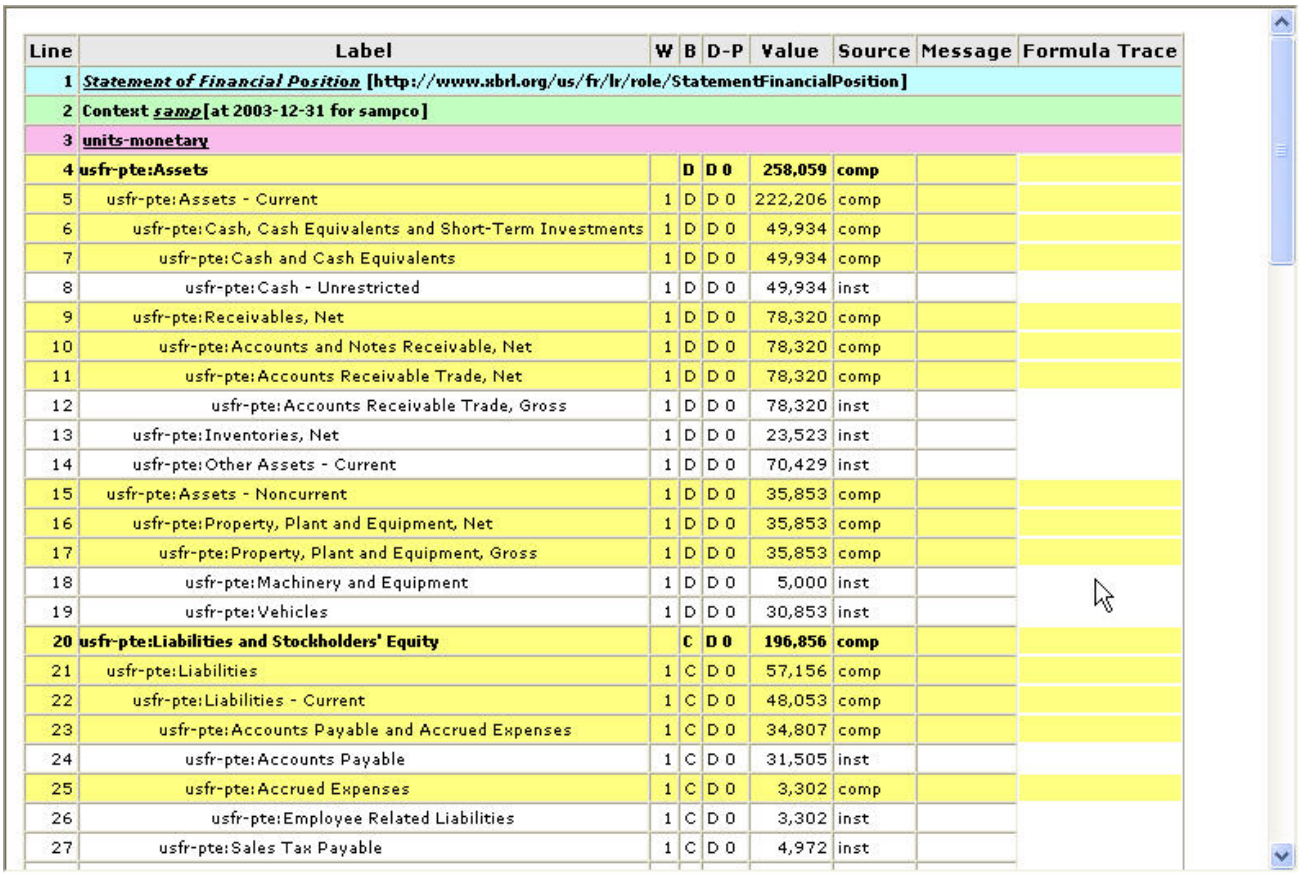
11 Select **Tools**, and then **Validate Instance Document**.



12 Select **XML**, **XBRL validation** and the **Compute Missing Fact Values** option.

The simple taxonomy does not contain every item listed in the US GAAP taxonomy, and this option calibrates XBRL Taxonomy Designer to calculate the missing values.

13 Click **Run Validation**.



Line	Label	W	B	D-P	Value	Source	Message	Formula Trace
1	***Statement of Financial Position*** **[http://www.xbrl.org/us/fr/lr/role/StatementFinancialPosition]**							
2	**Context *samp* [at 2003-12-31 for sampco]**							
3	**units-monetary**							
4	**usfr-pte:Assets**		**D**	**D 0**	**258,059**	**comp**		
5	usfr-pte:Assets - Current	1	D	D 0	222,206	comp		
6	usfr-pte:Cash, Cash Equivalents and Short-Term Investments	1	D	D 0	49,934	comp		
7	usfr-pte:Cash and Cash Equivalents	1	D	D 0	49,934	comp		
8	usfr-pte:Cash - Unrestricted	1	D	D 0	49,934	inst		
9	usfr-pte:Receivables, Net	1	D	D 0	78,320	comp		
10	usfr-pte:Accounts and Notes Receivable, Net	1	D	D 0	78,320	comp		
11	usfr-pte:Accounts Receivable Trade, Net	1	D	D 0	78,320	comp		
12	usfr-pte:Accounts Receivable Trade, Gross	1	D	D 0	78,320	inst		
13	usfr-pte:Inventories, Net	1	D	D 0	23,523	inst		
14	usfr-pte:Other Assets - Current	1	D	D 0	70,429	inst		
15	usfr-pte:Assets - Noncurrent	1	D	D 0	35,853	comp		
16	usfr-pte:Property, Plant and Equipment, Net	1	D	D 0	35,853	comp		
17	usfr-pte:Property, Plant and Equipment, Gross	1	D	D 0	35,853	comp		
18	usfr-pte:Machinery and Equipment	1	D	D 0	5,000	inst		
19	usfr-pte:Vehicles	1	D	D 0	30,853	inst		
20	**usfr-pte:Liabilities and Stockholders' Equity**		**C**	**D 0**	**196,856**	**comp**		
21	usfr-pte:Liabilities	1	C	D 0	57,156	comp		
22	usfr-pte:Liabilities - Current	1	C	D 0	48,053	comp		
23	usfr-pte:Accounts Payable and Accrued Expenses	1	C	D 0	34,807	comp		
24	usfr-pte:Accounts Payable	1	C	D 0	31,505	inst		
25	usfr-pte:Accrued Expenses	1	C	D 0	3,302	comp		
26	usfr-pte:Employee Related Liabilities	1	C	D 0	3,302	inst		
27	usfr-pte:Sales Tax Payable	1	C	D 0	4,972	inst		

14 Save this instance as `sampleid2.xbrl`.

Note: For ease of use, it is recommended that you save all files for this lesson in `samples\tutorial\mapping\Sample Company\WorkBook` folder to keep track of the file for use in future lessons. The `Workbook` folder has added to the mapping folder to provide a place to save documents without overwriting the sample files.

You completed Mapping Lesson Three and the Mapping Tutorial.

Dimensions Tutorial

This tutorial and series of lessons walk you through the fundamental creation steps for a dimensional taxonomy using XBRL Taxonomy Designer.

For the purposes of this tutorial, we divided the dimensional taxonomy into six taxonomies, one of which combines all schemas into a dimensional taxonomy. This division is intended to help you learn the concepts into simple building steps. It is feasible to create the same dimensional taxonomy as one taxonomy.

Disclaimer: These dimensional taxonomies tutorials for dimension and dimensions aggregation is based upon the www.xbrl.org candidate recommendation (CR3) dated 3/30/2006. Because this is a recommendation there will be future changes in the specification and therefore it is important to verify which specification your taxonomies are built against.

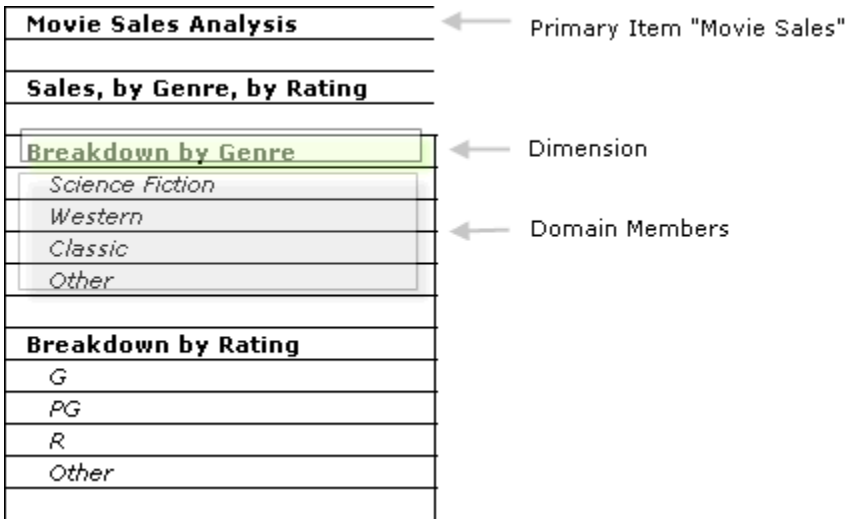
The sample files reference in this tutorial can be found in the `samples` folder under Dimensions.

Lessons in this tutorial include:

- "Lesson One: Creating Domain-Member Taxonomies" on page 328 teaches you how to use a taxonomy to define a dimension using outlining the domain-members of that dimension.
- "Lesson Two: Creating a Dimension Taxonomy" on page 331 teaches you how to create a dimension taxonomy.
- "Lesson Three: Creating a Primary Item Taxonomy" on page 332 teaches you how to use a taxonomy to define a primary item.
- "Lesson Four: Creating a Hypercube Taxonomy" on page 333 teaches you how to use a taxonomy to define a hypercube.
- "Lesson Five: Generating an XBRL Instance Document with Dimensions" on page 335 teaches you how to use the context editor to insert values into an instance document. This lesson is designed to teach users how the instance editor works. Advanced users use the Instance Document Context Import Mapping function.
- "Lesson Six: Creating XBRL Instance Documents with Dimensional Contexts" on page 342 shows you how to use the Import option to populate instance documents with data.
- "Lesson Seven: Creating a Multi-Hypercube Taxonomy with a Restricted Hypercube" on page 345 shows how to generate a multiple hypercube taxonomy and how to restrict one of them.

Lesson One: Creating Domain-Member Taxonomies

The domain-member taxonomy defines a set of explicit members that can be used to describe the dimension.



The first step of modelling dimensional data begins with identifying the building blocks of the dimensional taxonomy. Begin with the primary item (the item to be measured), and then identify the dimensions, and then the domain members (elements explicitly used to describe the dimension in a standardized way). The primary item for the above example is Movie Sales and the dimensions are Genre and Rating. The domain members for Movie Genres are Science Fiction, Westerns, Classics, Other.

Using the steps below create the movie genre dimension Domain-Members taxonomy; repeat the same steps to create the Region dimensions domain member taxonomy. Completed files created from this lesson can be found at `\samples\tutorial\dimensions`.

➤ To create a domain-member taxonomy:

1 **Create a Taxonomy.**

2 **Enter the unique namespace Identifier as http://www.example.com/genre_domainmembers.**

3 **Enter the unique Prefix Genre-domainmembers.**

4 **Add a reference to the `xbrldt.xsd` schema by selecting File, then Add, then Reference Taxonomy.**

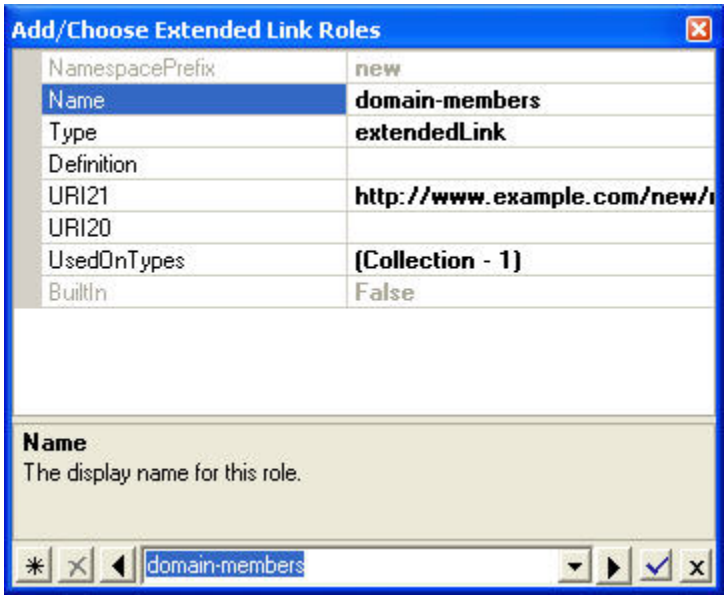
5 **Use the Web location button to select `xbrldt-2005.xsd`.**

This schema contains the dimensional taxonomy syntax.

6 **Change the Relationship View from Presentation to Definition View and set the arc role filter to (domain-members).**

7 **Create a domain-members extended link by right-clicking on the Definition View and selecting Add Extended Link.**

8 **Click the * button to add an extended link role.**



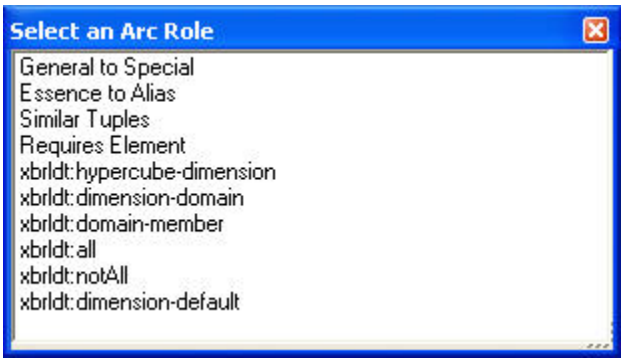
9 **Enter the name `domain-members` and a URI21 for the extended link and click the check mark button.**

10 **Select the taxonomy properties tabbed window and set the default data type to: Monetary for each domain-member element.**

Create the domain members with the data type Monetary so that it matches the data type of the primary item being measured; however this is just a convention and is not mandatory.

11 Add these child elements of Genre: **Science Fiction**, **Western**, **Classic**, and **Other**.

Note: If your arc role filter is set to show all arc roles each time you drag an element to be a child of the domain members, you are prompted to select an arc role for the relationship being created. As these elements are domain members of the Movie Genre dimension, select the xbrldt:domain-member arc role.



12 Change the **Period** type of the Movie Genre and the domain member elements to **Duration** so it matches the period by which these elements are expected to be reported as.

13 To construct the dimension calculations using the dimension-aggregation arc role, load the dimension-aggregation schema by selecting **File**, then **Load**, then **Referenced Taxonomy**, and then browse to `samples\tutorial\dimensions\xbrldta-2005.xsd`.

14 Change the Relationship View from **Definition View** (show all arc-roles) to **Presentation View**.

15 Right-click the **Product** element and select **Copy**, then **subtree**, and then **relationships only**.

16 Change the Relationship View from **Presentation** relationship view to the **Calculation** relationship view and set the **arc role filter** to **dimension-aggregation**.

17 Right-click the **Calculation View** and paste the **Movie Genre** relationship tree into the **Calculation View**.

18 Select **Aggregation** for the Arc Role.

19 Save this taxonomy as `Genre_DomainMember.xsd`.

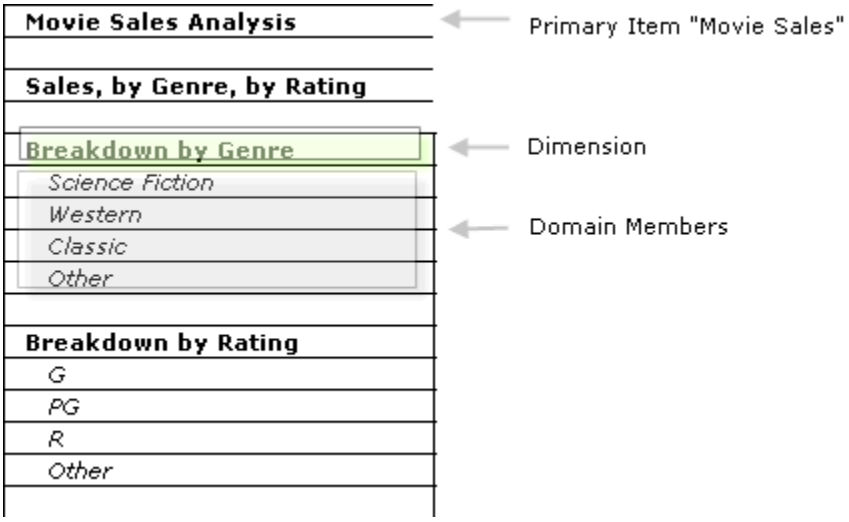
This completes the creation of the Movie Genre Domain-Member taxonomy `Genre_DomainMember.xsd`.

20 For the purposes of the next lesson, repeat the steps above to create a Rating Domain-Member taxonomy `Rating_DomainMember.xsd` with the domain members **G**, **PG**, **R**, and **other**.

After you created these domain-member taxonomies, proceed to "Lesson Two: Creating a Dimension Taxonomy" on page 331.

Lesson Two: Creating a Dimension Taxonomy

During the last lesson, we identified two dimensions and created the domain-members taxonomies for those dimensions. The next step is to create the dimension taxonomies. Files created from this lesson can be found at `\samples\tutorial\dimensions`.



➤ To create a dimension taxonomy:

1 **Create a Taxonomy.**

2 **Enter the unique namespace Identifier http://www.example.com/genre_dimension and Prefix genre_dimension.**

3 **Add a reference to the `xbrldt.xsd` schema by selecting File, then Add, and then Reference Taxonomy.**

4 **Click Web location and select `xbrldt-2005.xsd`.**

This schema contains the dimensional taxonomy syntax.

5 **Add a reference to the Genre domain members taxonomy to the Genre dimension taxonomy by selecting File, then Add, then Reference Taxonomy, and then select `genre_domainmembers.xsd` that was created during Lesson One.**

6 **Change the arc role relationships view from Presentation to the Definition View.**

7 **Set the arc role filter to Show all arc-roles.**

8 **Create an extended link named genre_dimension.**

9 **Create the dimension element named dimension-item.**

10 **Change the element properties of dimension-item to these:**

- Abstract: **True**
- Substitution group: **dimension-item**
- Period: **Duration**

  This defines the dimension as valid and defines it is measuring the primary item over a set period of time.

11 Drag the **dimension-item** element from the element list to the **genre-dimension extended link**.

12 Drag the **Movie Genre** element from the element list to be a child of **dimension-item element** in the presentation view.

13 Select the **Movie Genre** element and select **dimensions-domain** as the **arc-role**.

This arc role defines the connection between the dimension and its domain which contains the domain members elements that describe the dimension.

14 Select the element **Movie Genre**, then select the element property window, and then select **parentlink**.

15 Change **useable** to **True**.

16 Change the **TargetRole** to **domain-dimensions**.

17 Save this taxonomy as `Genre_dimension.xsd`.

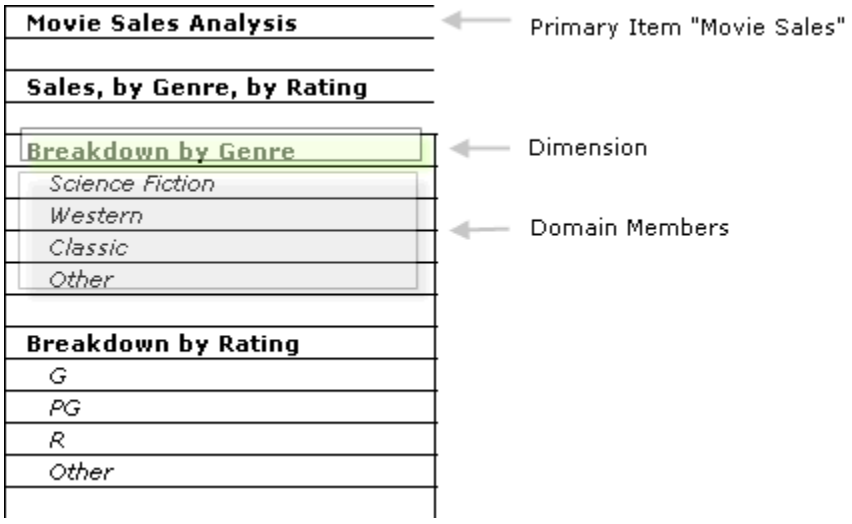
This completes the creation of the Movie Genre `Genre_dimension.xsd` dimension taxonomy.

18 For the purposes of the next lesson, repeat these same steps to complete the Rating `Rating_dimension.xsd` dimension taxonomy.

After you completed the Dimension taxonomies in Lesson Two, proceed to "Lesson Three: Creating a Primary Item Taxonomy" on page 332.

Lesson Three: Creating a Primary Item Taxonomy

The next step is to create the primary item taxonomy (which contains the element used to report the values). Completed files created from this lesson can be found at `samples\tutorial\dimensions`.



➤ To create a Primary Item taxonomy:

1 Create a Taxonomy.

2 Enter the unique namespace Identifier **http://www.example.com/Primaryitem** and Prefix **Primaryitem**.

3 Select the default data type to: **monetary**.

4 Add an element named **Movie Sales** and set these attributes:

- Period type: **Duration**
- Datatype: **Monetary**

5 Save this taxonomy as `MovieSales_primary_item.xsd`.

You completed the creation of the Primary Items taxonomy for Lesson Three. Proceed to "Lesson Four: Creating a Hypercube Taxonomy" on page 333.

Lesson Four: Creating a Hypercube Taxonomy

The final step in creating the dimensional taxonomy is to create the hypercube taxonomy which references all taxonomies created from lesson one through to lesson three. The hypercube taxonomy defines the hypercubes and the dimensions that are effective.

Note: Completed files created from this lesson can be found at `samples\tutorial\dimensions`.

➤ To create a hypercube taxonomy:

1 Create a Taxonomy.

2 Enter a unique namespace Identifier **http://www.example.com/Hypercube** and Prefix **Hypercube**.

3 Add a reference to the `xbrldt.xsd` schema by selecting **File**, then **Add**, and then **Reference Taxonomy**.

4 Click the **Web** location button and select `xbrldt-2005.xsd`.

This schema contains the dimensional taxonomy syntax.

The purpose of this taxonomy is to construct the hypercube connecting it to the effective dimensions. The primary item is connected to its dimensions using the hypercubes.

5 Select **File**, then **Add**, then **Reference Taxonomy**, and then **Genre_Dimension**.

6 Select **File**, then **Add**, then **Reference Taxonomy**, and then **Rating_Dimension**.

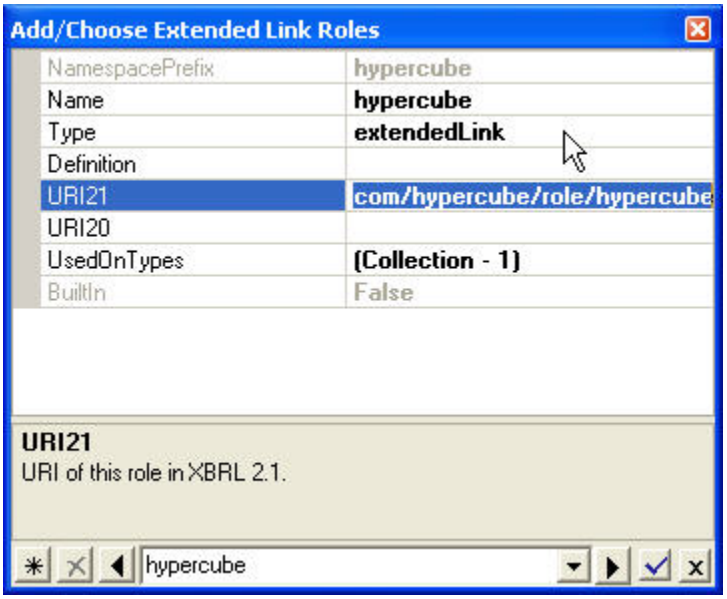
7 Select **File**, then **Add**, then **Reference Taxonomy**, and then `MovieSales_primary_item.xsd`.

8 Change the arc role relationship view from **Presentation** to **Definition View**.

9 Ensure that you arc role filter is set to **show all arc-roles**, as you may have noticed that several arc roles are in use within the definition relationship view to construct a dimensional taxonomy.

10 To create the hypercube, select the **Definition View**, right-click, and then select **Add an extended link**.

11 Click the * button to create an extended link role.



12 Change the **Name** to **hypercube**.

13 Change the **URI21** to **http://www.example.com/role/Hypercube**.

14 When you are finished, click the **check mark** button.

The next step is to create the relationship between the Primary Item and its associated hypercubes.

15 Drag the element **Movie Sales** to the extended link **hypercube**.

Next we must define the dimensions that are related to the hypercubes.

16 Create the element named **hypercube-item**.

17 Drag the element **hypercube-item** to **Movie Sales**.

18 Right-click, then select **Change**, and then set **Arc Role** to **hypercube-dimensions** for the relationship between the hypercube and the primary item.

19 Change the element properties of **hypercube-item** to these:

- Abstract: **True**
- Substitution group: **hypercube-item**
- Period Type: **Duration**

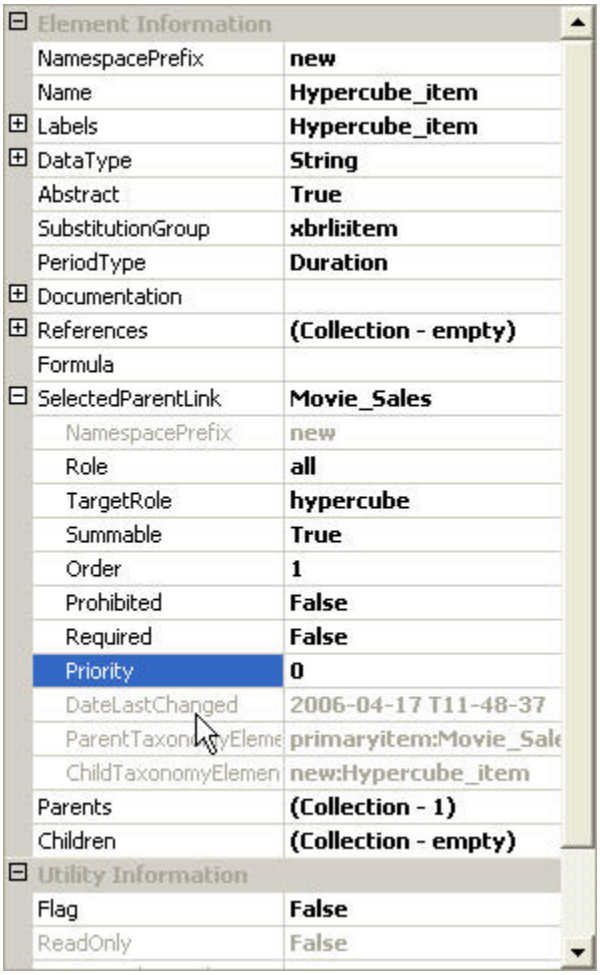
20 Select the **hypercube-item** element, and then the element **property** window.

21 Select the selected **parentlink**, and then change **summable** to **true**.

Setting the hypercube and dimensions to summable means that the values reported total the primary item.

22 Right-click and select **change**, then **arc role**, and then **all**.

23 Change the **TargetRole** to **Hypercube**.



24 Drag **genre-dimension** and **rating-dimension** from the element list to be a child of hypercube-item.

25 Change the **targetrole** for these relationships to match their select dimension.

For example, the targetrole of genre-dimension should be the dimension-item from genre-dimension.

26 Save this taxonomy as `hypercube.xsd`.

You completed Dimensions Lesson Four. Proceed to "Lesson Five: Generating an XBRL Instance Document with Dimensions" on page 335.

Lesson Five: Generating an XBRL Instance Document with Dimensions

After you create a dimensional taxonomy, the next step is to generate an XBRL instance document that uses the dimensional taxonomy and its concepts.

Note: Completed files created from this lesson can be found at `\samples\tutorial\dimensions`.

➤ To create an Dimensional XBRL instance document using dimensions:

1 Select **File**, then **New**, and then **XBRL instance document**.

2 Select **File**, then **Add**, and then reference taxonomy, and then select the schema for instance documents `xbrldi-2006.xsd`.

The syntax within this schema defines how we should construct our instance with dimensions.

3 Select **File**, then **Add**, then **reference taxonomy**, and then select `hypercube.xsd` **(created in Lesson Four)**.

Now for each primary item called movie sales reported in the XBRL instance, a context must be constructed that contains the dimensions, and explicit domain members to identify which row from the table is being reported.

Movie Sales Analysis		2003	2002	2002
Sales, by Genre, by Rating		32,038	35,805	32,465
Breakdown by Genre				
Science Fiction	Fact Value	20,181	18,150	15,275
Western		2,433	1,973	1,823
Classic		6,675	6,514	5,752
Other		2,749	9,168	9,615
	Grand Total	32,038	35,805	32,465

Genre dimension

Science Fiction domain member

Reported primary item value for the science fiction genre for all rated movies.

The hypercube was built with the 'all' constructor therefore we must report each primary item with both dimensions within the 'scenario' for each context defined.

4 Select **Context**, and then **Add**.

5 Enter the label for the context using the ID field: D-**2003-genres-sciencefiction**.

6 Select **Entity**, then the ellipses, and then click the * button to create an entity.

7 Set the Identifier as **Movie Sales Example**.

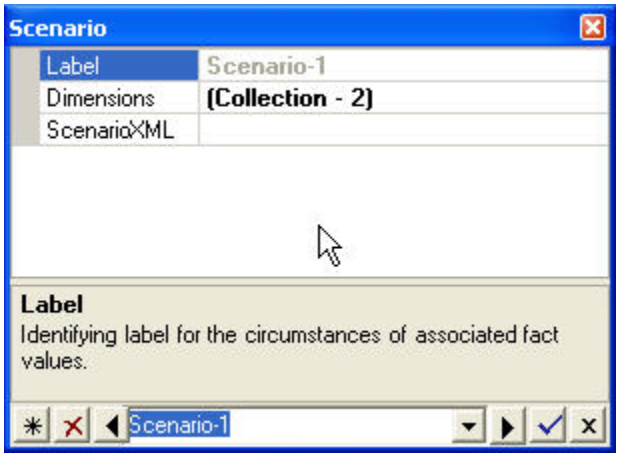
8 Set the entities scheme as **http://www.example.com/movie_store_example**, and then click the **check mark** button.

9 Select **Periods**, then the ellipses, and then click the * button to create a period.

10 Set the period type for this context to: **Duration**.

11 Set **start date** for the period to **1-1-2003** and **end date** to **12-31-2003**.

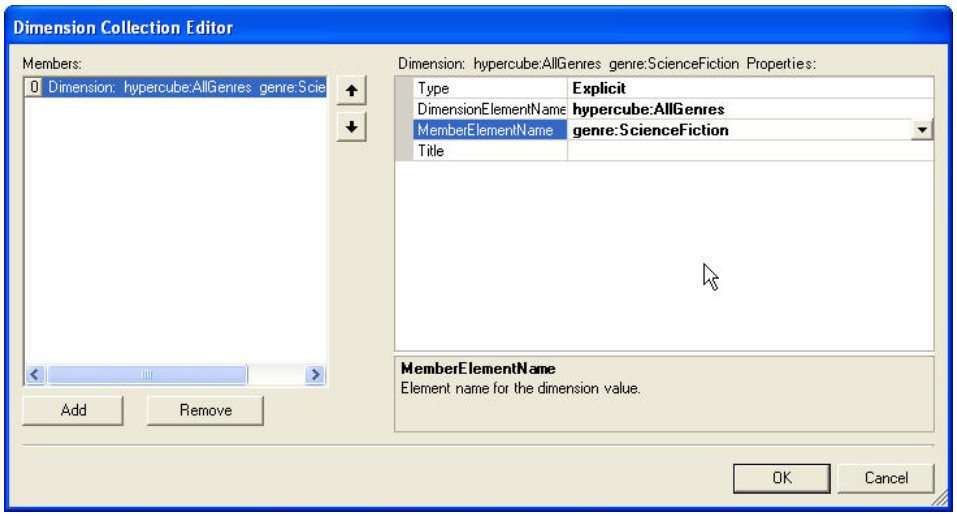
12 Select **Scenarios**, then the ellipses, and then select **Dimensions**.



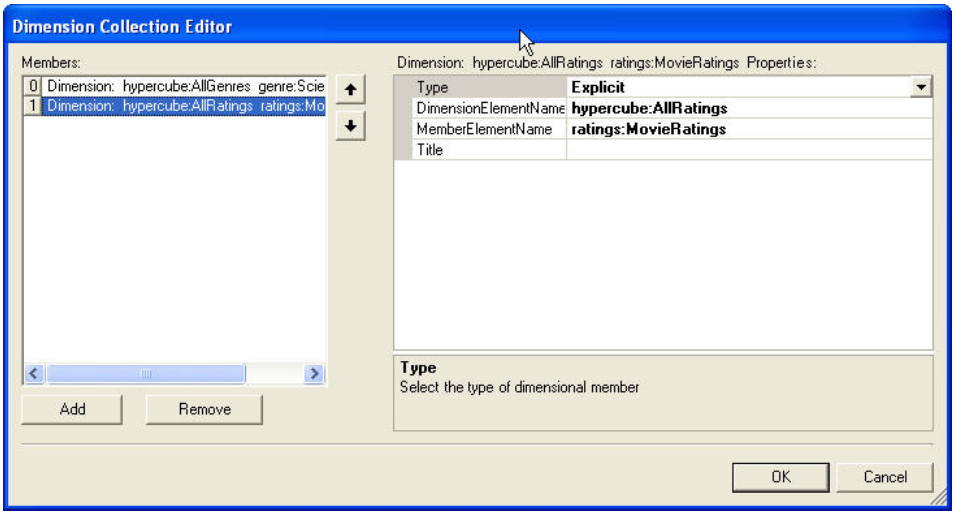
13 Set the type to **Explicit** as we explicitly define the dimension and its containing domain members for each scenario.

14 Select the **hypercube:AllGenres** for the DimensionElementName.

15 Select the **Genre:ScienceFiction** for the MemberElementName.



16 To add a second dimensions, select **hypercube:AllRatings** for the DimensionElementName and **ratings:MovieMatings** for MemberElementName.

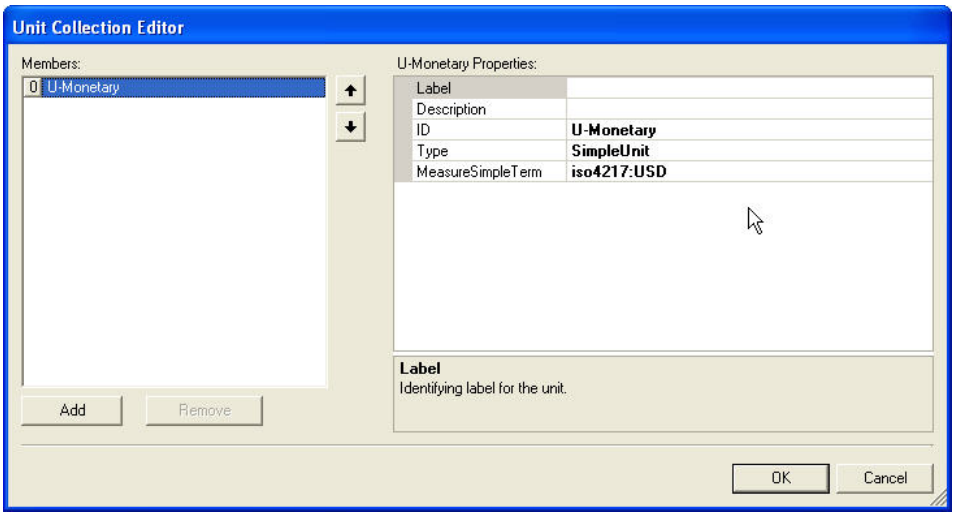


17 Click **OK**, and then click the **check mark** button to accept the changes.

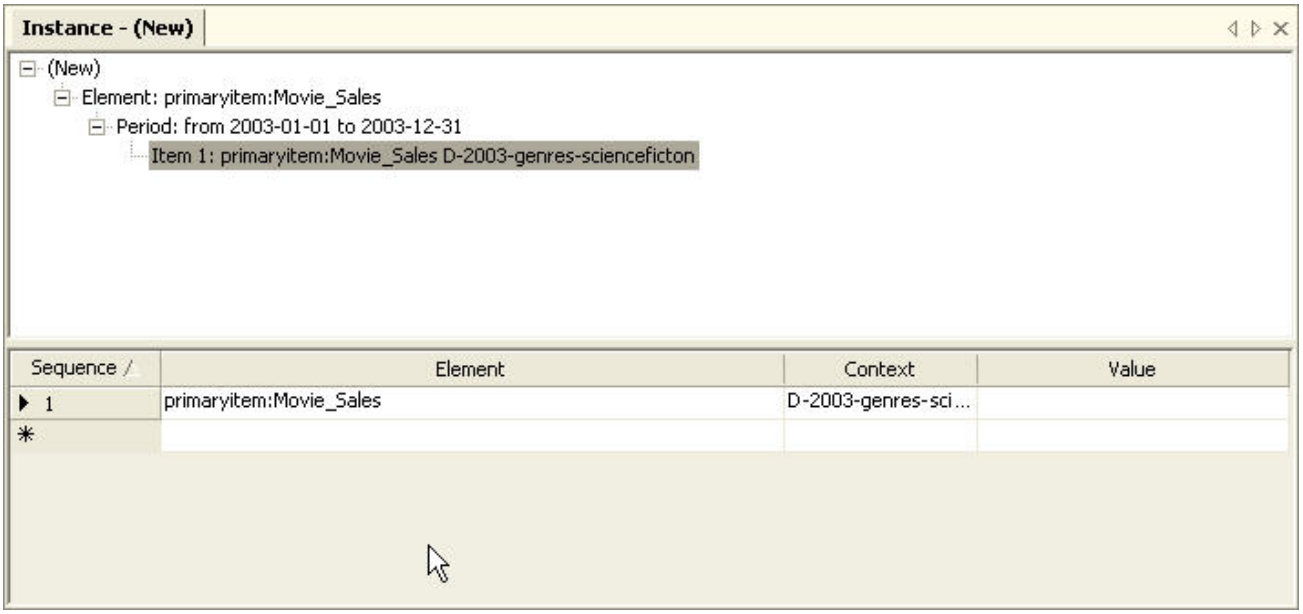
18 Click **OK**.

19 From **Element**, select **primaryitem:Movie_Sales**.

20 From **Context**, select **d-2003-genres-sciencefiction**, and then enter the value `20181`.



21 Click the **Units** ellipsis to use the Unit Collection Editor to create a unit titled **U-Monetary** with the **MeasureSimpleTerm** value of **iso4217:USD**



You entered in your first fact value for the instance document. This fact value is the value highlighted in yellow below.

The fact value entered in Step 15 represents the figure highlighted below.

22 **Repeat Steps 6 to 13 for each additional fact value.**

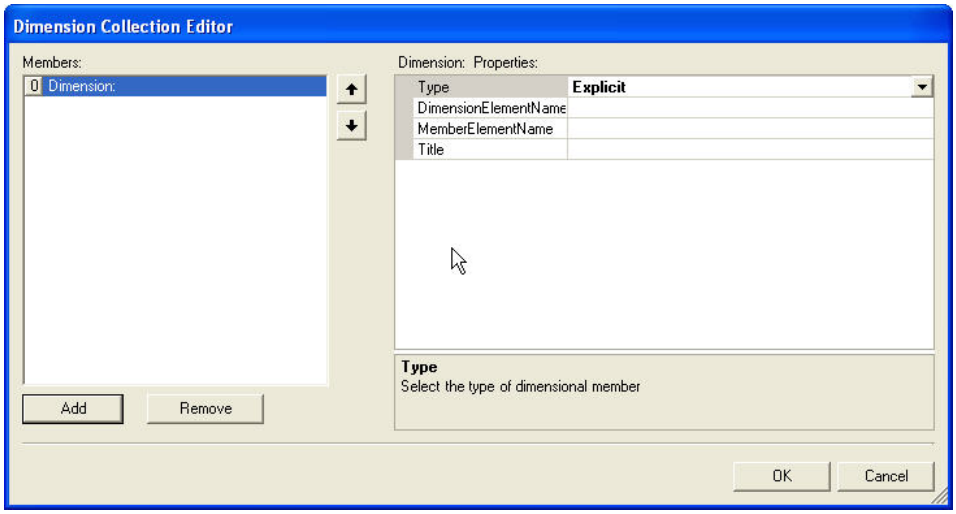
After you entered the fact values, you must create an element for each of the yearly totals.

23 **Create a context named D-2003-allgenre-allratings.**

24 **Select the Entity ellipsis to use the Entity Collection Editor to accept the entity movie store.**

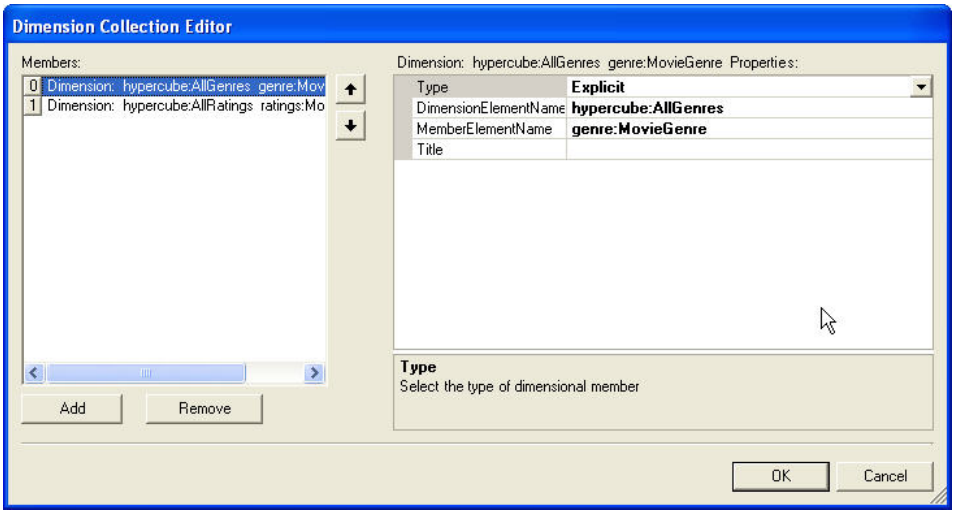
25 **Select the Periods ellipsis to use the Periods Collection Editor to select duration of 1-1-2003 to 12-31-2003.**

26 **Select the Scenarios ellipsis to use Scenarios Collection Editor to add a dimensions scenario.**



27 **Add a dimension: Select the hypercube:AllGenres for the Dimension Element Name, and then select the Genre:MovieGenre for the Member Element Name.**

28 Add a second dimension; Select **hypercube:AllRatings** for the Dimension Element Name and **ratings:MovieMatings** for Member Element Name.



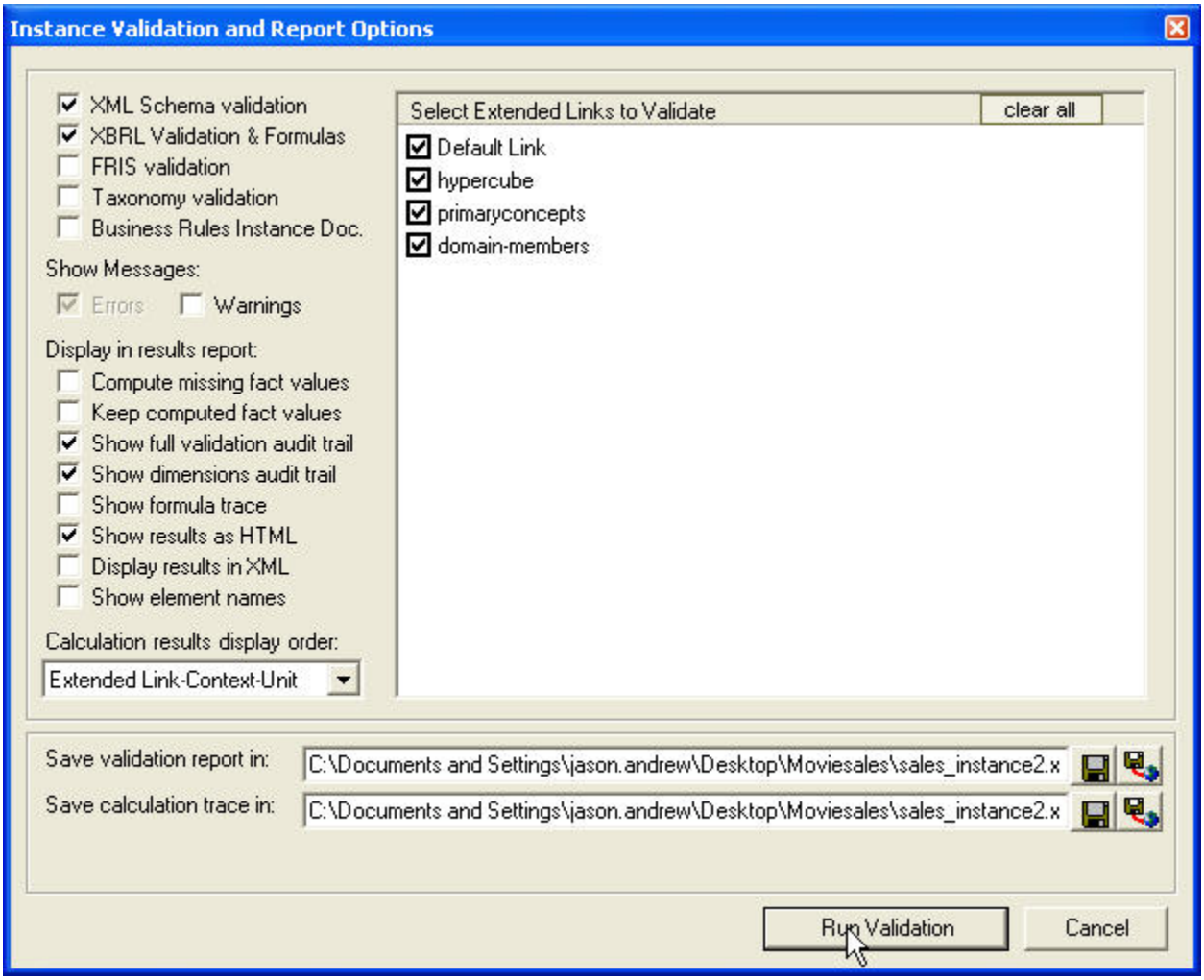
29 Click **OK**, and then click the **check mark** button to accept the changes.

30 Click **OK**.

31 Repeat steps 17-24 for the years 2002 and 2001.

32 Ensure that you selected the instance menu and toolbar.

33 To validate the instance document, select **Tools**, and then **Validate Instance Document**.



34 Check the **XBRL Validation** box and the **Show dimensions audit trail** box.

35 Click **Run Validation**.

See "Reading a Calculation Trace" on page 151.

Line	Label	W	B	D-P	Value	Source	Message
1	**Dimensions period from 2001-01-01 to 2001-12-31 for moviestore Scenario-9**						
2	**U-Monetary**						
3	**hypercube:all genres//genre:Movie Genre**						
4	**primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Movie Ratings]**			**D 0**	**32,465**	**both**	**OK**
5	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:G]	1		D 0	10,137	inst	
6	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:PG]	1		D 0	10,396	inst	
7	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:R]	1		D 0	3,210	inst	
8	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Others]	1		D 0	8,722	inst	
9	**hypercube:all ratings//ratings:Movie Ratings**						
10	**primaryitem:Movie_Sales[..., hypercube:all genres//genre:Movie Genre]**			**D 0**	**32,465**	**both**	**OK**
11	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Science Fiction]	1		D 0	15,275	inst	
12	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Westerns]	1		D 0	1,823	inst	
13	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Classics]	1		D 0	5,752	inst	
14	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Others]	1		D 0	9,615	inst	
15	**Dimensions period from 2002-01-01 to 2002-12-31 for moviestore Scenario-9**						
16	**U-Monetary**						
17	**hypercube:all genres//genre:Movie Genre**						
18	**primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Movie Ratings]**			**D 0**	**35,805**	**both**	**OK**
19	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:G]	1		D 0	12,649	inst	
20	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:PG]	1		D 0	10,374	inst	
21	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:R]	1		D 0	4,371	inst	
22	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Others]	1		D 0	8,411	inst	
23	**hypercube:all ratings//ratings:Movie Ratings**						
24	**primaryitem:Movie_Sales[..., hypercube:all genres//genre:Movie Genre]**			**D 0**	**35,805**	**both**	**OK**
25	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Science Fiction]	1		D 0	18,150	inst	

You completed Dimensions Lesson Five. Proceed to "Lesson Six: Creating XBRL Instance Documents with Dimensional Contexts" on page 342.

Lesson Six: Creating XBRL Instance Documents with Dimensional Contexts

Entering in the data for an instance document can be time consuming and impractical. The XBRL Taxonomy Designer Import option enables you to populate instance documents with data.

Note: Completed files created from this lesson can be found at `samples\tutorial\dimensions`.

➤ To use the Import function to import dimensional data:

1 **Create an instance document.**

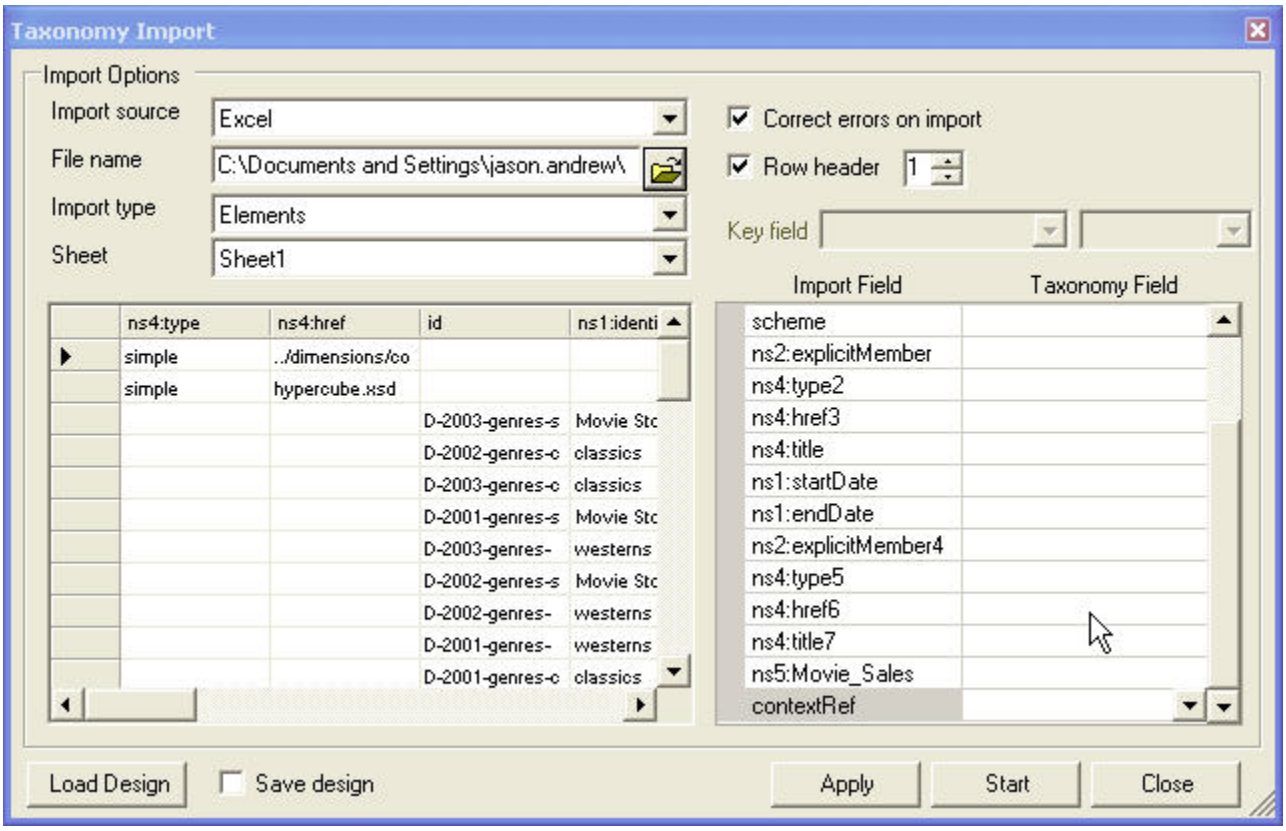
2 **Load the reference taxonomy file** `xbrldi-2006.xsd`.

3 Load the **Hypercube taxonomy** created in Lesson Three.

4 Select **File**, and then **Import**.

5 Use the **Browse** option to browse to `Movie_Sales.xsl`.

This is a Microsoft Excel file with all of the data from the example entered.

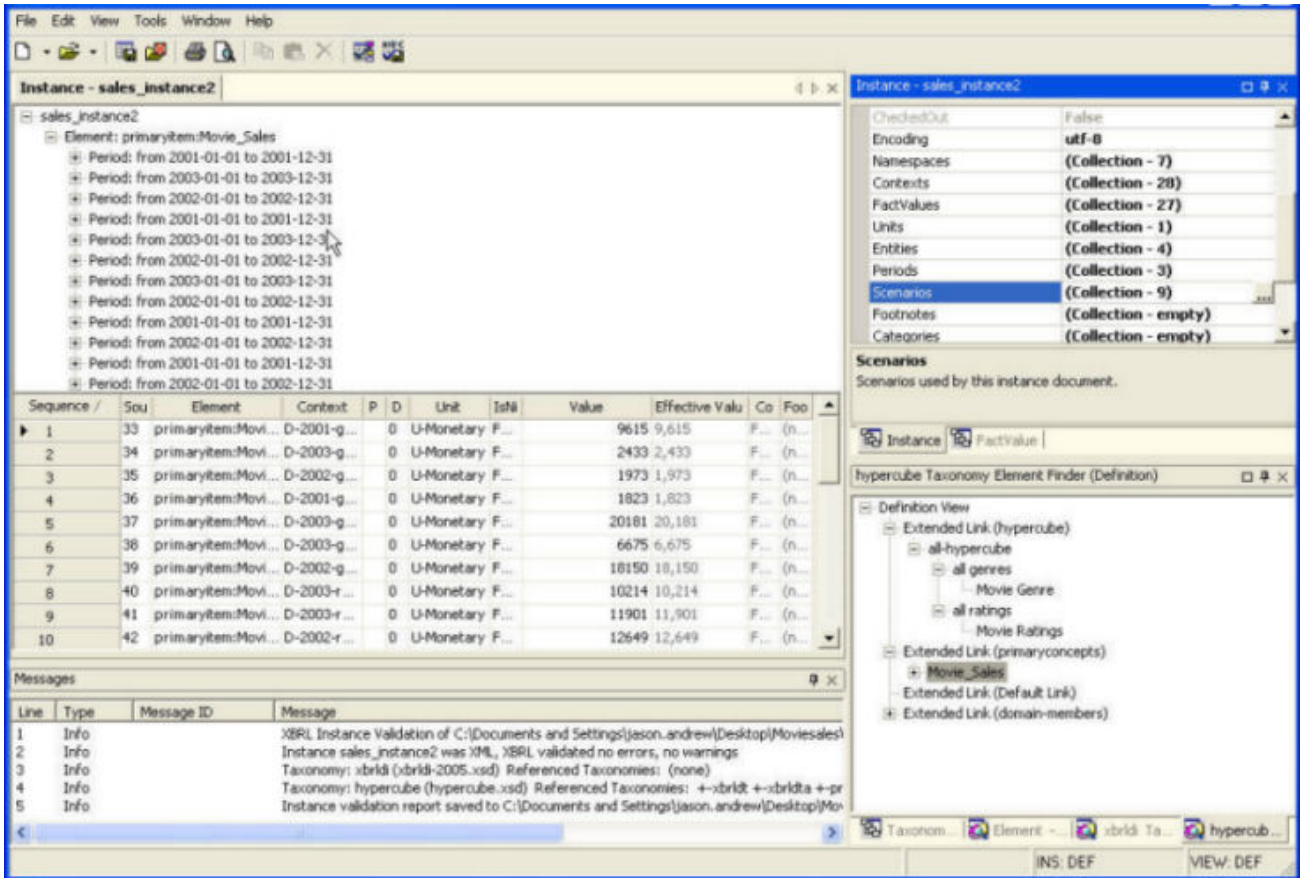


6 Click **Apply**.

7 For Import Type, select **units** and click **Apply**.

8 For Import Type, select **facts** and click **Apply**.

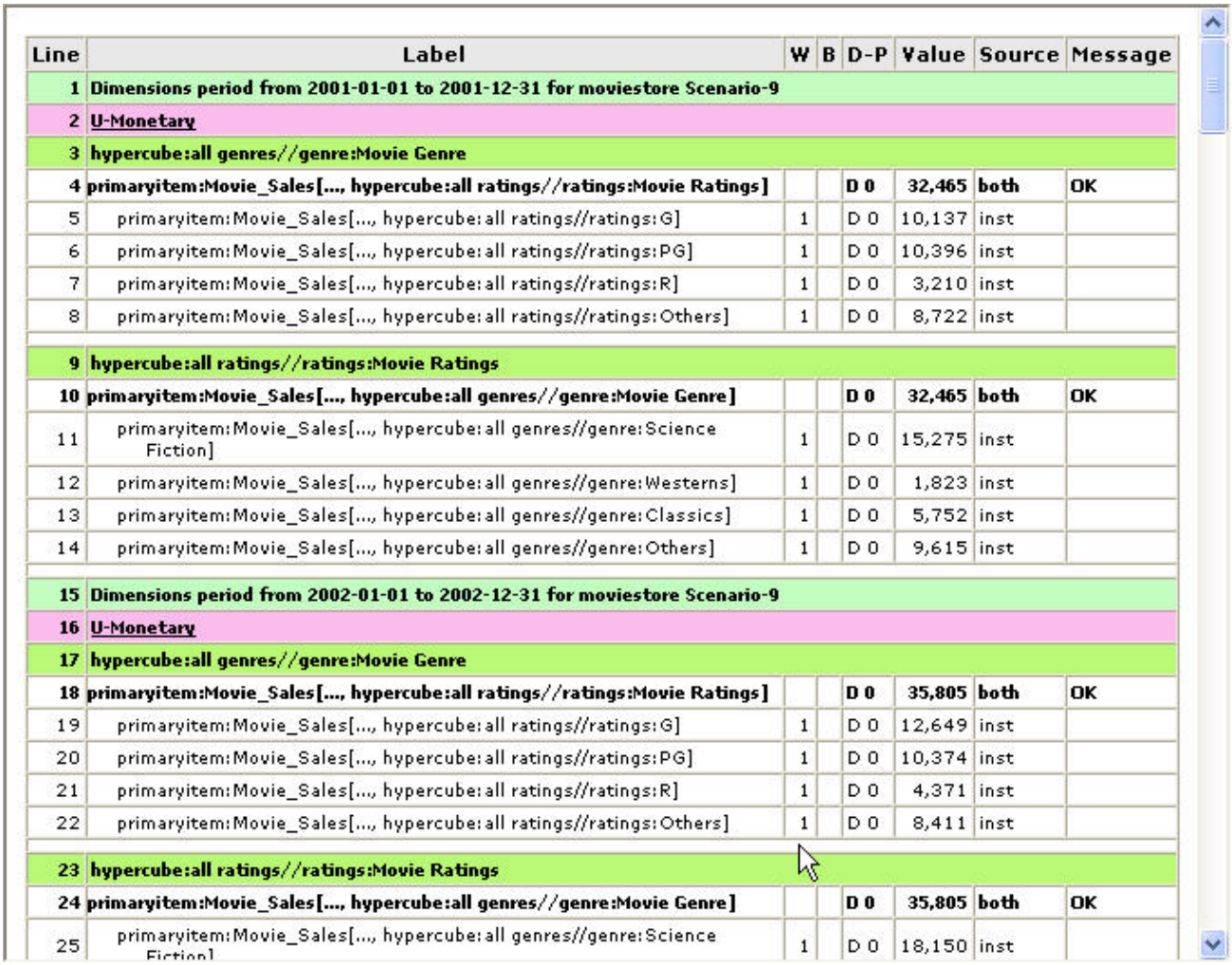
9 Click **Start**.



10 Select the **XBRL Validation** box and the **Show dimensions audit trail** box.

11 Click **Run Validation**.

See "Reading a Calculation Trace" on page 151.



Line	Label	W	B	D-P	Value	Source	Message
1	**Dimensions period from 2001-01-01 to 2001-12-31 for moviestore Scenario-9**						
2	**U-Monetary**						
3	**hypercube:all genres//genre:Movie Genre**						
4	**primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Movie Ratings]**			**D 0**	**32,465**	**both**	**OK**
5	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:G]	1		D 0	10,137	inst	
6	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:PG]	1		D 0	10,396	inst	
7	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:R]	1		D 0	3,210	inst	
8	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Others]	1		D 0	8,722	inst	
9	**hypercube:all ratings//ratings:Movie Ratings**						
10	**primaryitem:Movie_Sales[..., hypercube:all genres//genre:Movie Genre]**			**D 0**	**32,465**	**both**	**OK**
11	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Science Fiction]	1		D 0	15,275	inst	
12	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Westerns]	1		D 0	1,823	inst	
13	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Classics]	1		D 0	5,752	inst	
14	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Others]	1		D 0	9,615	inst	
15	**Dimensions period from 2002-01-01 to 2002-12-31 for moviestore Scenario-9**						
16	**U-Monetary**						
17	**hypercube:all genres//genre:Movie Genre**						
18	**primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Movie Ratings]**			**D 0**	**35,805**	**both**	**OK**
19	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:G]	1		D 0	12,649	inst	
20	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:PG]	1		D 0	10,374	inst	
21	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:R]	1		D 0	4,371	inst	
22	primaryitem:Movie_Sales[..., hypercube:all ratings//ratings:Others]	1		D 0	8,411	inst	
23	**hypercube:all ratings//ratings:Movie Ratings**						
24	**primaryitem:Movie_Sales[..., hypercube:all genres//genre:Movie Genre]**			**D 0**	**35,805**	**both**	**OK**
25	primaryitem:Movie_Sales[..., hypercube:all genres//genre:Science Fiction]	1		D 0	18,150	inst	

You completed Dimensions Lesson Six. Proceed to "Lesson Seven: Creating a Multi-Hypercube Taxonomy with a Restricted Hypercube" on page 345.

Lesson Seven: Creating a Multi-Hypercube Taxonomy with a Restricted Hypercube

Now that you've created a complex hypercube, you want to go a step further. What if you decide to restrict certain elements from your video store? For example, you noted that Movies with the Category Rating Other do not sell as well as other movies. Further, some of these movies are not suitable for adults and you wish to restrict these movies.

Note: Completed files created from this lesson can be found at `samples\tutorial\dimensions`.

➤ To create a restricted hypercube taxonomy:

1 **Open the taxonomy Hypercube from Lesson Four.**

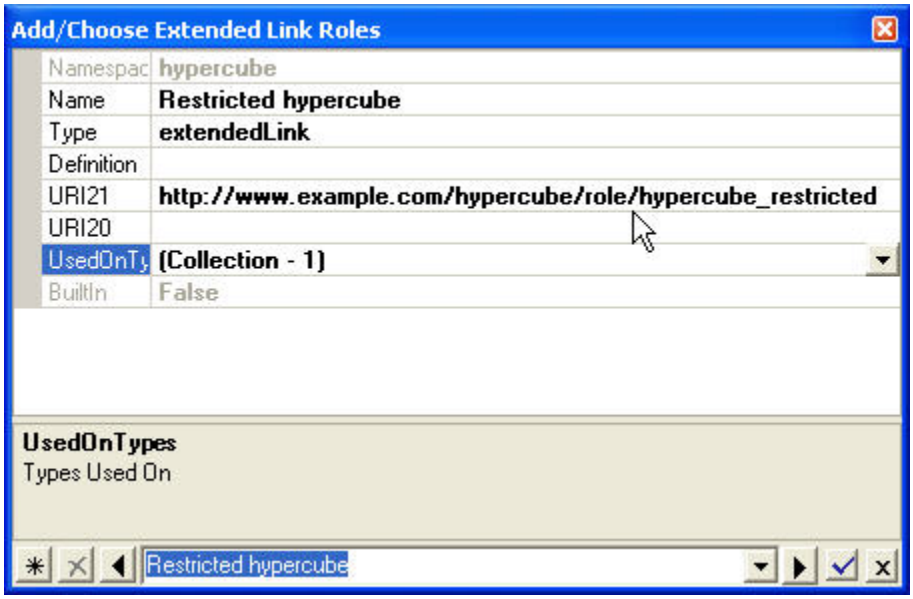
2 Save the Taxonomy as `hypercube_restrict.xsd`.

We are going to modify elements from the hypercube taxonomy to create a restricted, multihypercube example.

3 Switch to **Definition View** with all of the roles shown.

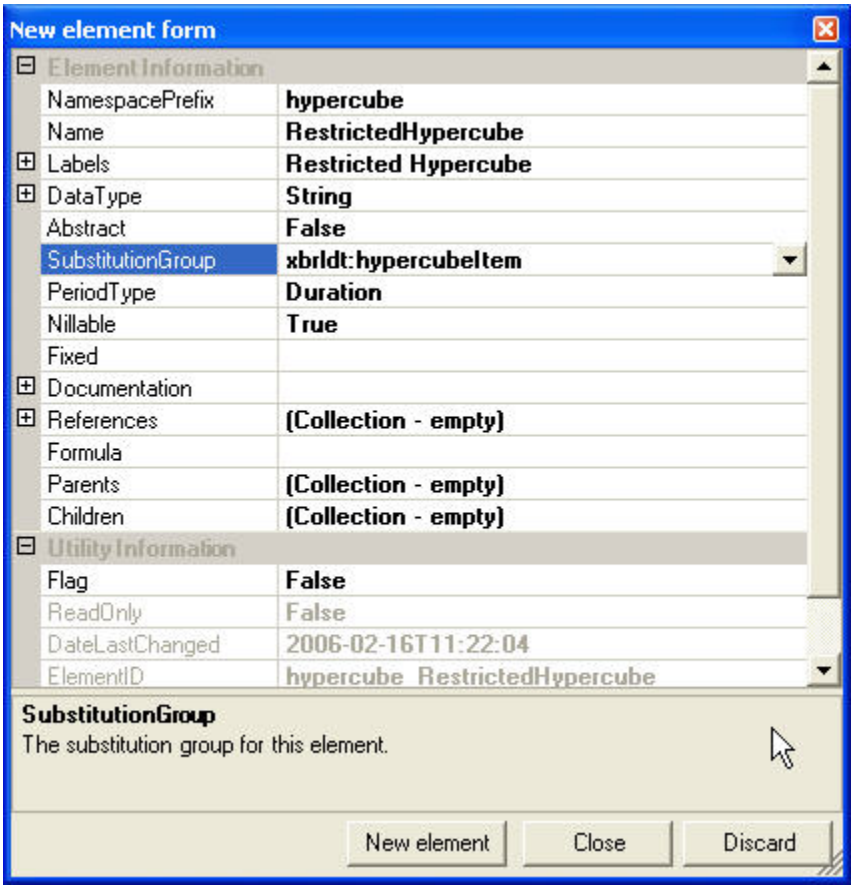
4 Add an **Extended Link** called **Hypercube_restricted**.

5 Alter the **URI** to **hypercube_restricted**.



6 Click the **check mark** button to save this extended link.

7 Add a child element named **Restricted Hypercube** to the Hypercube_restricted extended link.



This is the second hypercube in this example that restricts one of the dimensions.

8 Change the **Substitution Group** to **xbrldt: hypercube-item**.

9 Change the **Abstract** value to **True**.

10 Drag **Restricted Hypercube** from the Element list to be a child of **hypercube-restricted**.

11 Drag the **Move Ratings** element from the Element List to be a child of the **Restricted Hypercube**.

12 Select **hypercube-dimension** as the **arcrole**.

Note: You can also drag elements with ALT pressed (drag-link icon) from a neighboring tree view branch (like the other hypercube).

13 Drag the **Ratings-Other** element from the element list and make it a child of **Movie Ratings**.

14 Select the **Dimension-Domain** for the **arcrole**.

15 Drag the **Restricted Hypercube** to be a child of Movie Sales under the hypercube extended link.

16 Select **xbrldt: not all**for the **Arc-role**.

17 Validate the taxonomy to ensure that everything works properly and then save the taxonomy.

Formulas Tutorial: Creating Simple Formulas to Validate Data

This tutorial teaches users about Formulas, XBRL, and XBRL Taxonomy Designer.

While a few formulas may be created using the Oracle Hyperion Disclosure Management XBRL Taxonomy Designer drag and drop formula builder with concepts in a presentation view, it can be limited to one formula per concept. To overcome this we create an extension taxonomy to create containers for a formula or ratio.

Note: Business rules are one of the most valuable features of XBRL. The ability to validate data and develop ratios for analytics is extremely valuable. Companies can run formulas against their financial reports before submission to regulators and also run private formulas they generated to validate internal policies or create analytical ratios. An advanced course in business rule generation is available

➤ To create a simple formula to validate data:

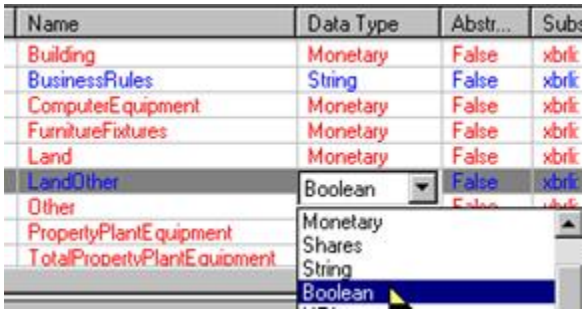
1. Create a taxonomy.
2. In the Taxonomy Property window, change the **namespace prefix** to **ci-br**.
3. Change the **namespace identifier** to **http://www.example.com/ci-businessrules**.
4. Select **File**, then **Save As**, and then enter `CIBusinessRules.xsd`.
5. Create a reference taxonomy through the Reference Taxonomies property grid using the Basic Calculation taxonomy.

   The Reference Taxonomies property grid now shows.
6. Select **Turn Off**, then **View**, and then **Namespace Prefix** to toggle it off so that when you edit concepts you are not editing the namespace prefix.
7. Select **File**, then **Save**, and then close the taxonomy,
8. Open **ci-businessrules** taxonomy and notice that BasicCalculation taxonomy is now a referenced taxonomy to your ci-businessrules taxonomy.

   Note: If you want to use colors to distinguish between the reference taxonomy (BasicCalculation) and your taxonomy cibusinessrules, select Tools, and then Options and move colors of your choice. Save the taxonomy and reload it.

   Now let's edit some elements and create elements where we can store formulas.
9. Right-click the element **Property, Plant and Equipment** to edit and add the word (View).

   This separates the concepts from the containers we create to store formulas.
10. Add the element **BusinessRules** as a concept at the same level as Property, Plant and Equipment (View)
11. Add a concept under Business Rules and call it **Land > Other.**
12. Set the data type to **Boolean**.



13 Select **Land > Other** in the presentation view.

14 Select **Business Rules** and change **Abstract** to **True**.

15 Select **Formula**, and then select **[...]** from the element property grid.

16 Select **ci:Land** from the presentation relationship view, drag it to the formula editor.

17 Enter **>**.

18 Select **ci:Other** from the presentation relationship view, drag it to the formula editor.

19 Select to validate the formula.

If the formula is valid, a message is displayed in the messages window

If the formula is invalid an error is displayed

20 Select **[x]** to close the formula editor

21 Select the element property window and note the formula expression,

22 In the Taxonomy Property window, select **Linkbases**, and then **[...]**.

23 Select the formula linkbase and note the FormulaType property.

The default is Disclosure Management (for the Oracle Hyperion Disclosure Management propriety format), this can be changed to XBRL 1.0 (for the XBRL international formula 1.0 specification).

Note the formula linkbase that is added; note the expression that is detailed and the element that the formula is associated with.

24 Select **Tools**, and then **XBRL Validation** to verify that the taxonomy is XBRL valid.

25 Select **[x]**, select **File**, then **Save**, and enter `CIBusinessRules.xsd`.

Let's create an instance document for our extended taxonomy and validate the business rules. We can do this quickly by using the **Tools Taxonomy Generate Business Rules** option.

26 Enter a name for the instance document; for example, **ciBusinessRulesInstance**.

27 Select **All elements** in the selected extended links.

28 For the Monetary amount, select **$100**.

A value of $100 is entered into the instance for each concept. Because there are 5 concepts the total should be $500 and be reported in the instance document.

29 Click **Start** and **Close**.

Now let's validate the instance document and the business rule, but before we do let's change some values so the validation triggers some errors.

30 Close the taxonomy and open the instance document you just created.

31 Run the business rule validation from the menu toolbar.

32 Change values for land to less than $100 and run validation again.

An XBRL element can contain the attribute substitution group containing one of these default values— xbrl:item, or xbrl:tuple; or it may also substitute for another element of the same type.

- **Concept**—Concepts are defined in two equivalent ways. In a syntactic sense, a concept is an XML Schema element definition, defining the element to be in the item element substitution group or in the tuple element substitution group. At a semantic level, a concept is a definition of kind of fact that can be reported about the activities or nature of a business activity.
- **Item**—An item is an element in the substitution group for the XBRL item element. It contains the value of the simple fact and a reference to the context (and unit for numeric items) needed to correctly interpret that fact. When items occur as children of a tuple, they must also be interpreted in light of the other items and tuples that are children of the same tuple. There are numeric items and non-numeric items, with numeric items being required to document their measurement accuracy and units of measurement.
- **Tuple**—A tuple is an element in the substitution group for the XBRL tuple element. Tuples are used to bind together the parts of a compound fact. Those constituent parts

are themselves, facts but they must be interpreted in light of each-other. For example, the name, age and compensation of a director of a company must be grouped together to be correctly understood.

Sample screenshots

Substitute for xbrl:Item because the element is a distinct item reported instance document

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyItem	MyItem	Monetary	False	xbrli:item

Substitute for xbrl:Tuple because the tuple groups together other items

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyTuple	MyTuple	Tuple		xbrli:tuple

Substitute for another element of the same type, if they are not the same type an error occurs. This technique enables you to substitute for elements of the same type, and enables you to restrict the original concepts definition.

Prefix	Label	Name	Data Type	Abstract	Substitution Group
new	MyItem	MyItem	Monetary	False	xbrli:item
new	Another...	AnotherI...	Monetary	False	new:MyItem

Index

E

F

H

I

L

M

N

O

P

R

S

T

U

V

W

X