

FUJITSU Software

Interstage XWand V13

A decorative horizontal band with a red-to-dark-red gradient, featuring abstract, glowing white and red lines that swirl and intersect, creating a sense of motion and technology.

Formula Editor/Data Mapping Tool

User's Guide

Windows

B1WD-3277-01ENZ0(00)
August 2013

Preface

Purpose of this Guide

This is the User's Guide for Formula Editor / Data Mapping Tool (hereafter referred to as "this function"). This manual describes how to use the definition creation and editing functions with definitions based on Formula specifications and with the mapping definitions used by data mapping functions.

Intended Readership

This manual is intended for the following readers:

- Users creating or editing definitions based on Formula specifications
- Users creating or editing the definitions used by data mapping functions

It is assumed that readers of this manual understand XBRL 2.1 specifications, Formula 1.0 specifications, and related standards specifications.

Related Manuals

None.

Export Restrictions

This software product is designed primarily for use in the country to which the software is imported. The software must not be re-exported from the country without authorization from the appropriate government authorities in accordance with the Foreign Exchange and Foreign Trade Control Law of the country. In no event will Fujitsu be liable for foreign government or third-party claims for compensation or damages whatsoever arising from the use of the software resulting in any violation of the laws and ordinances of the country or those of the relevant country. Fujitsu assumes no responsibility for any problems or damages arising from such use of the software outside the country. If Fujitsu suffers any damages as a result of user's violation of the above laws, Fujitsu reserves the right to claim compensation from the user.

Trademarks

- XBRL is a registered trademark of XBRL International, Inc. in the United States and other countries.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.
- All other company names and product names used in this manual are trademarks or registered trademarks of their respective owners.

Remarks

- This manual may be modified without notice.
- Please do not give copies of this manual to other people.
- Fujitsu is not responsible for any infringement of third-party patents of other rights arising out of use of the data in this manual.

Publication Date

First edition, August 2013

-

Contents

Chapter 1 Overview.....	1
1.1 Features.....	1
1.2 Function Overview.....	2
1.3 Icons.....	3
1.4 Menus, Submenus, and Context Menus.....	4
1.5 Starting an Application.....	5
Chapter 2 Creating and Editing Formula Definitions.....	7
2.1 Overview of Definitions.....	7
2.1.1 Definitions related to processing.....	7
2.1.2 Definition structure and overview of structural elements.....	7
2.2 Procedure for Creating Definitions.....	8
2.3 Calculation Expressions.....	8
2.4 Verification Expressions.....	10
2.4.1 Consistency verification (Consistency Assertion).....	10
2.4.2 Existence verification (Existence Assertion).....	11
2.4.3 Value verification (Value Assertion).....	13
2.4.4 Validation Message.....	14
2.4.5 Create existence assertions for each item.....	16
2.4.6 Create value assertions for each item.....	17
2.5 Variable Definitions.....	19
2.6 Filter Definitions.....	23
2.6.1 Aspect Cover Filter.....	24
2.6.2 Concept Relation Filter.....	26
2.6.3 Drag and drop definitions.....	28
2.7 Other Operations.....	29
2.7.1 Deleting an association.....	29
2.7.2 Deleting a definition.....	29
2.7.3 Setting a filter association.....	29
2.7.4 Setting a default value.....	29
2.7.5 Extended Link Role Definition.....	31
2.8 Output Definitions.....	32
2.9 Labels.....	32
2.10 Custom Functions.....	33
2.11 Checking Definition Operation.....	35
Chapter 3 Creating and Editing Data Mapping Definitions.....	38
3.1 Overview of Functions.....	38
3.1.1 Column-orientated data and row-orientated data.....	40
3.2 Creating an Item Master Taxonomy.....	41
3.2.1 Item master CSV.....	41
3.2.2 Creating an item master taxonomy.....	42
3.2.3 Customizing an item master taxonomy.....	42
3.3 Creating a CSV Output Definition.....	42
3.3.1 Output setting.....	45
3.3.2 Wildcard.....	46
3.4 Creating a Group Definition.....	47
3.4.1 Manual grouping.....	48
3.4.2 Automatic grouping.....	54
3.5 Creating a Data Mapping Definition.....	57
3.5.1 Setting items that need not be mapped.....	57
3.5.2 Setting mandatory output items.....	58
3.5.3 Data mapping definition stratification.....	58
3.6 Confirming Operation.....	59

Chapter 4 Other Functions.....	61
4.1 Setting the Editor View.....	61
4.2 Searching.....	64

Chapter 1 Overview

1.1 Features

Formula Specification is regulated as XBRL 2.1 standard extended specifications.

It is used for defining the calculation and verification logic in order to apply the logic to instances.

By adopting this specification, instance collectors and submitters can share common verification rules that are written in a standard format. This allows submitters to verify the rules before submission, and therefore improves the efficiency of submitting and collecting instances.

The Formula specifications define the description methods for the following two purpose-specific types of expressions:

- Formula

A Formula defines an expression used to output calculation results based on instance data. A new instance can be generated on the basis of an existing instance by applying a Formula definition used for calculation.

- Assertion

An Assertion defines an expression used to verify instance data. An instance can be verified by applying a Formula definition used for verification.

The data mapping function that converts instances to CSV format to facilitate database storage uses Formula specifications to perform the calculations described above, to output calculation results on the basis of instance data, then to convert the results to CSV format again to perform data searches.

The images below show examples of systems that use Formula specifications and systems that use the data mapping function.

Figure 1.1 Example of a system that uses Formula specifications

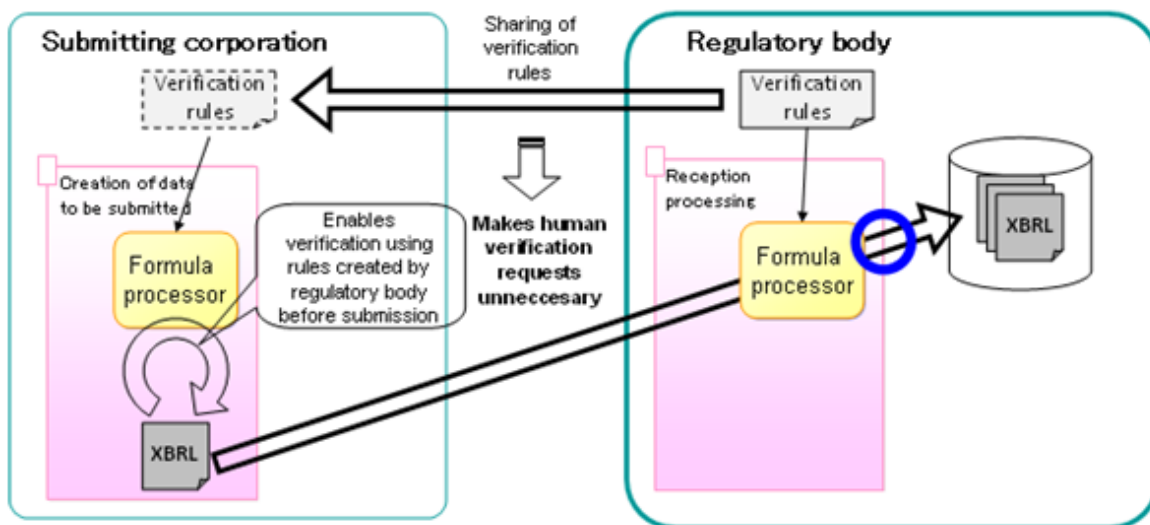
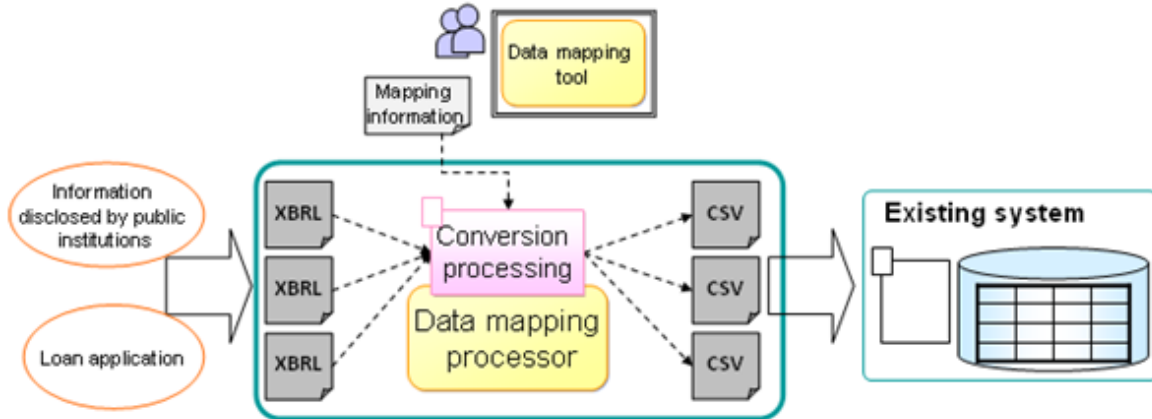


Figure 1.2 Example of a system that uses the data mapping function



This tool provides functions for graphically creating and editing definitions based on Formula specifications (hereafter referred to as Formula definitions), or for graphically creating and editing the mapping information used by the data mapping function (hereafter referred to as data mapping definitions).

Hereafter, the tool used to create Formula definitions is called the "Formula Editor", and the tool used to create data mapping definitions is called the "Data mapping tool".

Refer to Chapter 2 for information about creating Formula definitions, and to Chapter 3 for information about creating data mapping definitions.

[Note]

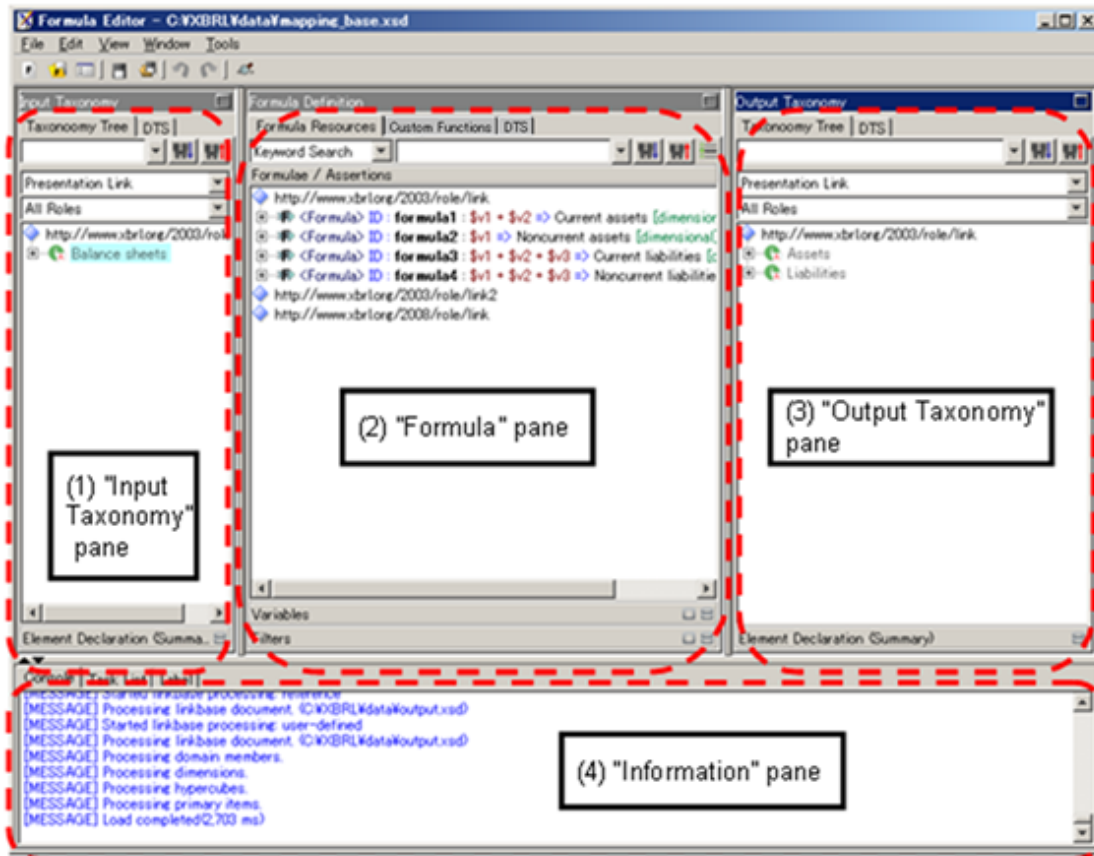
The user determines whether this tool is to be used as the "Formula Editor" or as the "Data mapping tool" at the time the tool is started. This cannot be switched after startup.

1.2 Function Overview

This tool is used to display the input and output taxonomies for calculations or data mapping, and to create and edit the calculations (using a graphical interface) and mapping definitions between those taxonomies. The tool comprises of the following panes:

- "Input Taxonomy" pane (upper left, (1))
This pane displays the input taxonomy information. The fields at the top of the pane can be used for searching by entering a keyword or for displaying a list box and switching between taxonomies.
- "Formula" pane (upper center, (2))
This pane displays the relationship expression between the elements in the input and output taxonomies, the variable definitions, the filter definitions, and the custom functions. If a definition that is being edited is not associated with an input taxonomy, the hierarchical structure of the definition files is displayed under the "DTS" tab.
- "Output Taxonomy" pane (upper right, (3))
This pane displays the output taxonomy information. The fields at the top of the pane can be used for searching by entering a keyword or for displaying a list box and switching between taxonomies.
- "Information" pane (bottom, (4))
This pane includes a console window, used for displaying messages such as file read-in and validation messages and errors, and a task list window that displays errors and warnings.

Figure 1.3 Application Window



1.3 Icons

This tool uses the following icons shown below.

Table 1.1 List of icons

Icon	Explanation	Icon	Explanation
	Formula		Imported Formula
	Formula for which the execution prohibited mark has been set		
	Consistency Assertion		Imported Consistency Assertion
	Existence Assertion		Imported Existence Assertion
	Value Assertion		Imported Value Assertion
	Assertion Set		Imported Assertion Set
	Fact Variable		Imported Fact Variable
	General Variable		Imported General Variable
	Parameter		Imported Parameter
	And Filter		Imported And Filter
	Or Filter		Imported Or Filter
	Aspect Cover Filter		Imported Aspect Cover Filter

	Concept Filter		Imported Concept Filter
	Dimension Filter		Imported Dimension Filter
	Entity Filter		Imported Entity Filter
	General Filter		Imported General Filter
	Match Filter		Imported Match Filter
	Period Filter		Imported Period Filter
	Relative Filter		Imported Relative Filter
	Segment Scenario Filter		Imported Segment Scenario Filter
	Tuple Filter		Imported Tuple Filter
	Unit Filter		Imported Unit Filter
	Value Filter		Imported Value Filter
	Indicates mapping is not required		Indicates an item for which output is mandatory
	Icon displayed for resources without associations		
	Item		Imported item
	Item for which abstract attribute is "true"		Imported item for which abstract attribute is "true"
	Tuple		Imported tuple
	Tuple for which abstract attribute is "true"		Imported tuple for which abstract attribute is "true"
	Element other than an item, tuple, or part element		Element other than an imported item, tuple, or part element
	Element other than an item, tuple, or part element for which abstract attribute is "true"		Element other than an imported item, tuple, or part element for which abstract attribute is "true"
	Hypercube Note that, if notAll is specified for arcrole, an "x" is attached in the Dimension tree.		Imported Hypercube
	Explicit Dimension Item		Imported Explicit Dimension Item
	Typed Dimension Item		Imported Typed Dimension Item
	Domain Member Only used in the Dimension tree.		Domain Member for which the usable attribute is "false" Only used in the Dimension tree.

1.4 Menus, Submenus, and Context Menus

Table 1.2 Menus and sub-menus

Menu	Submenus	Operation
File	New	Opens the "New" dialog. Specify the input taxonomy and output taxonomy information, and then open the window used for definitions. The definition being created, unless it is an independent file, can also be associated with an input taxonomy.
	Open	Opens the "Open" dialog. Specify a previously saved project file, or an existing definition file or input or output taxonomy information, then open the window used for definitions.

	Save Console Messages	Saves the warning or error messages that are output in the console window of the information pane.
	Save	Saves the definition that is currently being edited.
	Save As	Saves the definition currently being edited. Specify the save destination in the "Properties" dialog that is displayed after this menu is selected. A project file containing descriptions of relationships with referenced files or similar can also be saved.
	Property	Opens the "Properties" dialog. In this dialog, specify the file name and so on when saving definitions.
	Import Taxonomy Containing Formula Definition	Imports a definition from an extension source to the definition that is currently being edited.
	Detach Imported Taxonomy	Removes a definition that was previously imported.
	Close Current Editor	Closes the active window.
	End	Closes the application window.
Edit	Undo	Cancels the immediately previous operation.
	Redo	Reinstates the operation previously cancelled.
	Prefix Setting	Opens the "Prefix Setting" dialog that sets the prefix and name space URI.
	Default Value Setting	Opens the "Default Value Setting" dialog that sets the default value.
	Resource Reuse Setting	Opens the "Resource Reuse Setting" dialog to choose reusing the filter with the same definition or creating a new filter for a drag and drop operation.
	Create Assertion	Opens the dialog that is "Create Value Checks for Each Item" or "Create Existence Checks for Each Item" to create assertion
View	Editor View Setting	Changes the viewing settings for the information displayed in the "Formula", "Input taxonomy", and "Output taxonomy" panes.
	Select Related Concepts/Members When Formula/Assertion Is Selected	When this menu is selected, related concepts and members will be selected on the input taxonomy and output taxonomy panes when the Formula or Assertion is selected.
Window	Select Editor	Displays a list for switching to a different active editor.
Tools	Test Formulae/Assertions ("Formula Editor" only)	Applies to an instance the definition that is currently being edited so that its operation can be checked. Specify the application target instance in the dialog that is displayed when this menu is selected.
	Test mapping definition ("Data mapping tool" only)	Applies to an instance the definition that is currently being edited so that its operation can be checked. Specify the application target instance in the dialog that is displayed when this menu is selected.
	Options	Opens the "Options" dialog. Set the proxy information, output settings, URI mapping settings, search settings, tool bar settings, and so on.

1.5 Starting an Application

Use the following methods to start this tool:

If using the Formula Editor

From the start menu, select "Programs" - (purchased product) - "Tools" - "XBRL2.1" - Formula Editor".

If using the data mapping tool

From the start menu, select "Programs" - (purchased product) - "Data mapping function" - "Data mapping tool".

Chapter 2 Creating and Editing Formula Definitions

This chapter describes the procedures for creating Formula Definitions. The editing procedure, after an existing file has been opened, is the same as the creation procedure, so refer to this chapter also for editing.

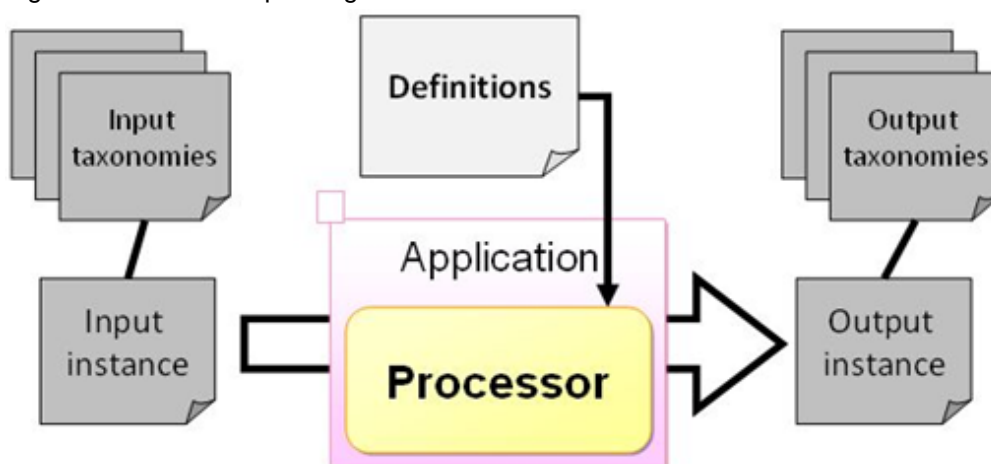
2.1 Overview of Definitions

This section gives an overview of Formula Definitions.

2.1.1 Definitions related to processing

The Formula definitions used for calculations perform calculations (based on expressions coded in the definitions) for an instance used as the input, and then outputs an instance. The figure below is a representation of the required definition relationships when processing is executed.

Figure 2.1 Relationships image for definitions



The Formula definitions used for verification only check the values for the input instance, and do not output an instance.

2.1.2 Definition structure and overview of structural elements

Formula definitions are constructed primarily from the concepts shown below and the mutual associations between them. This tool enables the concepts and mutual relationships to be connected using a graphical interface.

Concept	Explanation
Expression	This is a calculation expression or an expression used for verification. Expressions are based on XPath 2.0 specifications and are coded using specially defined variables. The following are examples: If the equity variable name is \$equity and the liabilities variable name is \$liabilities, the calculation expression for obtaining the sum of the equity and liabilities is "\$equity + \$liabilities ". The expression for verifying whether or not the equity is 10000 or greater is "\$equity > 10000".
Variable	This is a variable for use in expressions. Filters are used to specify which values from the input instance are to be associated.
Filter	A filter specifies which values from an input instance are to be associated with a variable. Various association methods are possible, such as associating using an element name, associating using a date specified in a context, and associating using a unit. Combinations of these conditions can also be set.
Conditional expression (optional)	A conditional expression is specified if conditions are attached to executing a calculation or verification. The following is an example:

If the equity variable name is \$equity and a calculation is to be executed only if the equity exceeds 10000, that condition is set as "\$equity > 10000".

2.2 Procedure for Creating Definitions

Follow the procedures below to create expression definitions using this tool:

- Investigating expressions
Investigate which values in the instance will be used, and whether calculation or verification is to be performed. Refer to Section 2.4, "Verification Expressions", for details of the procedure for creating expressions used for verification. The procedures below are for creating expressions used for calculations.
- Selection from the input taxonomy of the elements to be used for input.
In the "Input Taxonomy" pane, select the elements that are used in the expression and that constitute the input.
- Selection from the output taxonomy of output destination elements
In the output taxonomy pane, select the elements that constitute the output destination of the expression.
- Setting associations with input and output
Use drag and drop to set associations from input taxonomy elements to output taxonomy elements. This operation creates an expression for a new calculation and also, if required, automatically generates variable definitions and filter definitions for use by the expression. A variable can be added to a created expression. Drag and drop the association target variable from the variable window in the formula pane to the relevant expression. If the variable has not been defined, dragging and dropping the element from the "Input Taxonomy" pane to the expression sets the association with the expression and defines the variable and the filter. Refer to Section 2.5, "Variable Definitions", for details of variable definitions and to Section 2.6, "Filter Definitions", for details of filter definitions.
- Coding the expression
If more than one element is used as input, code the expression. Select an expression from the formula pane and right click to open the context menu. When "Edit Content" is selected in the context menu, the "Edit Content" dialog box is displayed and the calculation expression can be changed. Refer to Section 2.3, "Calculation Expressions", for details.
-

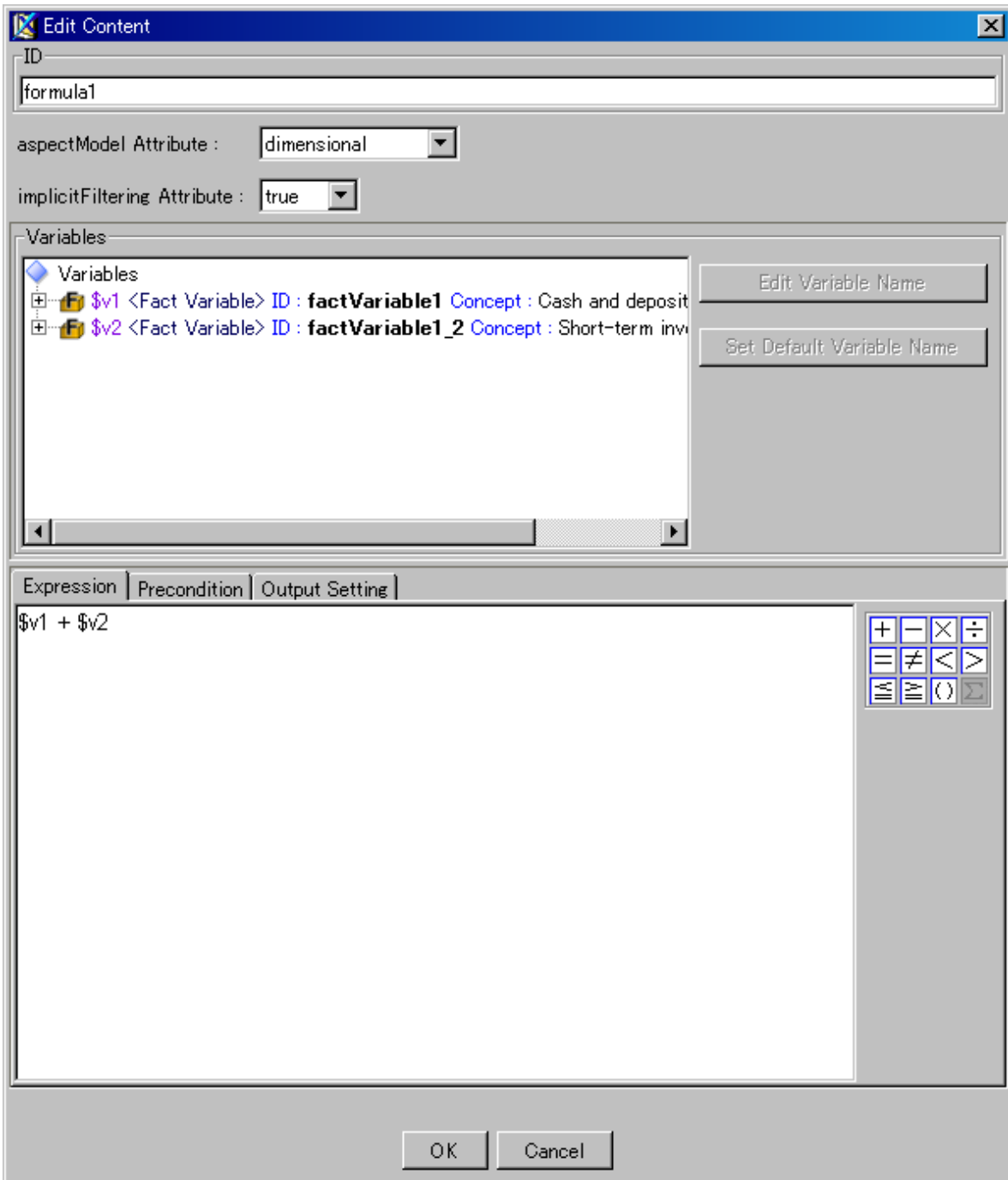
Use the above procedure to create definitions. Repeat the above procedure if more than one expression needs to be defined.

Refer to Section 2.4, "Verification Expressions", for details of the procedure for creating expressions to be used for verification.

2.3 Calculation Expressions

To refer to the detailed information of a created expression or to customize an expression, select the expression in the formula pane and right click to open the context menu. Select "Edit Content" from the context menu to display the "Edit Content" dialog box. The expression definition can be referenced and edited using this dialog.

Figure 2.2 Editing Content



The following information can be referenced and edited from the "Edit Content" dialog:

Item	Explanation
ID	The value used to uniquely identify the expression
aspectModel Attribute	In Formula specifications, the aspectModel attribute value
implicitFiltering Attribute	In Formula specifications, the implicitFiltering attribute value
Variables	Displays the variable associated with this calculation expression. The "Edit Variable Name" button can be used to rename the variable. The "Set Default Variable Name" button can be used to rename the generated variable name.

Expression	The associated calculation expression. The calculation expression can be changed.
Precondition	Specify if you want to attach conditions to execution of the calculation expression.
Output Setting	Output destination of the calculation results. Normally, an association has been set with an output taxonomy element. Refer to Section 2.8, "Output Definitions", for details of output definitions.

2.4 Verification Expressions

To create an expression to be used for verification, select from the formula pane the extended link role to which you want to add the verification expression, and right click to open the context menu. In the context menu, select "Add Formula/Assertion" and specify the type of verification you want to add.

The following table shows the verification types. In the examples below, the equity variable name is \$equity, the liabilities variable name is \$liabilities, and the assets variable name is \$assets.

Type	Explanation
Consistency verification (Consistency Assertion)	Verifies whether or not the difference between the input instance value and the result value calculated using the calculation expression is within a fixed range. For example, if the assets value is compared with the sum of the equity and liabilities, to verify whether or not the difference is within 5%, define "\$equity + \$liabilities " as the expression and, as the output, define 0.05 as the permitted difference range in the subject corresponding to assets.
Existence verification (Existence Assertion)	Verifies whether or not the value exists the specified number of times
Value verification (Value Assertion)	Verifies whether or not the expression used for the described verification is fulfilled. For example, the expression that verifies whether or not the sum of the equity and liabilities matches the assets is "\$assets = \$equity + \$liabilities ".

In addition, the context menu contains "Assertion Set" and "Formula". "Assertion Set" is used to group together the above verification expressions. Refer to Section 2.3, "Calculation Expressions", for information about "Formula".

Variables can be associated with each verification expression in the same way as for calculation expressions. Refer to Section 2.2, "Procedure for Creating Definitions", for details.

To set detailed information concerning verification expressions, select in the formula pane each of the verification definitions created as described above, and right click to open the context menu. Select "Edit Content" from the context menu to display the "Edit Content" dialog box. Definitions are set and edited in this dialog box.

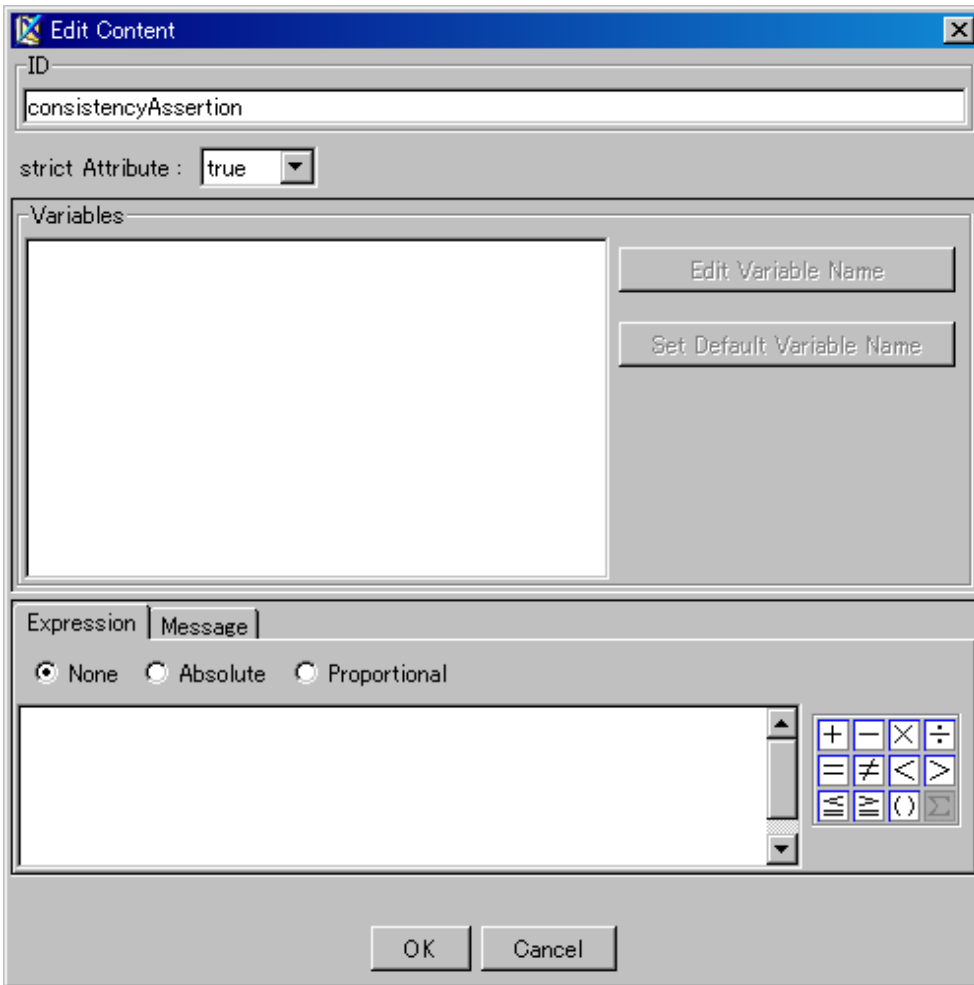
The following sections show the items that can be set and edited in each of the verification types.

There is a wizard to make an assertion easily which checks value for each item and checks its existence. The wizard can also make assertions which check for multi items.

2.4.1 Consistency verification (Consistency Assertion)

Consistency verification definitions define a calculation expression and compare the result obtained from that calculation expression with the input instance value to perform verification. Refer to Section 2.3, "Calculation Expressions", for information about defining calculation expressions. Note that a calculation expression that has already been defined can be dragged and dropped to a verification definition to associate the calculation expression with that verification expression.

Figure 2.3 Editing Content (Consistency Assertion)

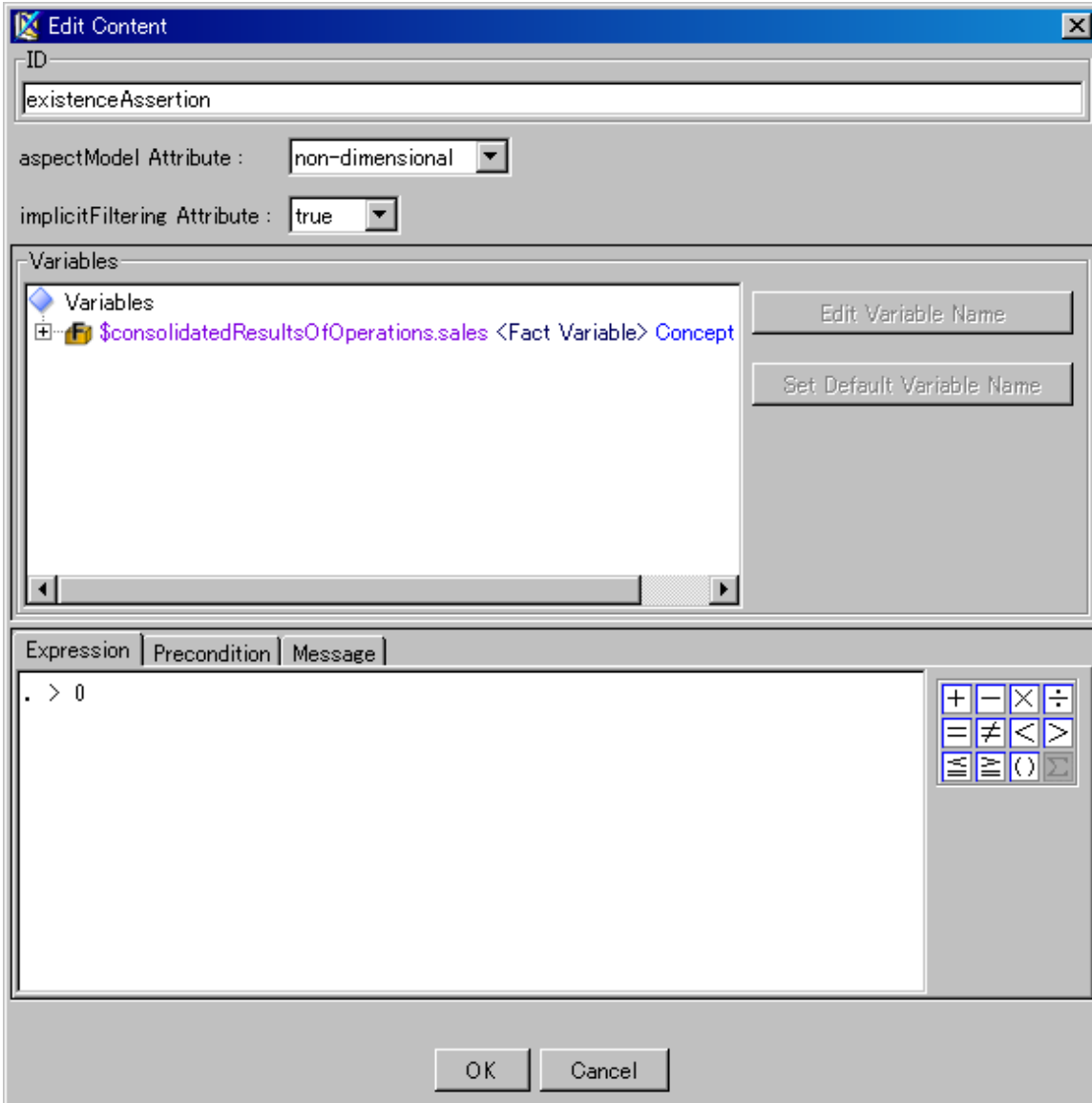


Item	Explanation
ID	The value used to uniquely identify the expression
strict Attribute	In Formula specifications, the strict attribute value
Variables	Displays the variable associated with this verification expression. The "Edit Variable Name" button can be used to rename the variable. The "Set Default Variable Name" button can be used to rename the generated variable name.
Expression	Specify the permitted range for the difference between the result provided by the calculation expression and the input instance value.
Message	Define the message that corresponds to the validation result. Refer to Section 2.4.4, "Validation Message" for information on how to define the message.

2.4.2 Existence verification (Existence Assertion)

The existence verification definition defines a calculation expression and compares the result provided by that calculation expression and the input instance value to perform verification.

Figure 2.4 Editing Content (Existence Assertion)

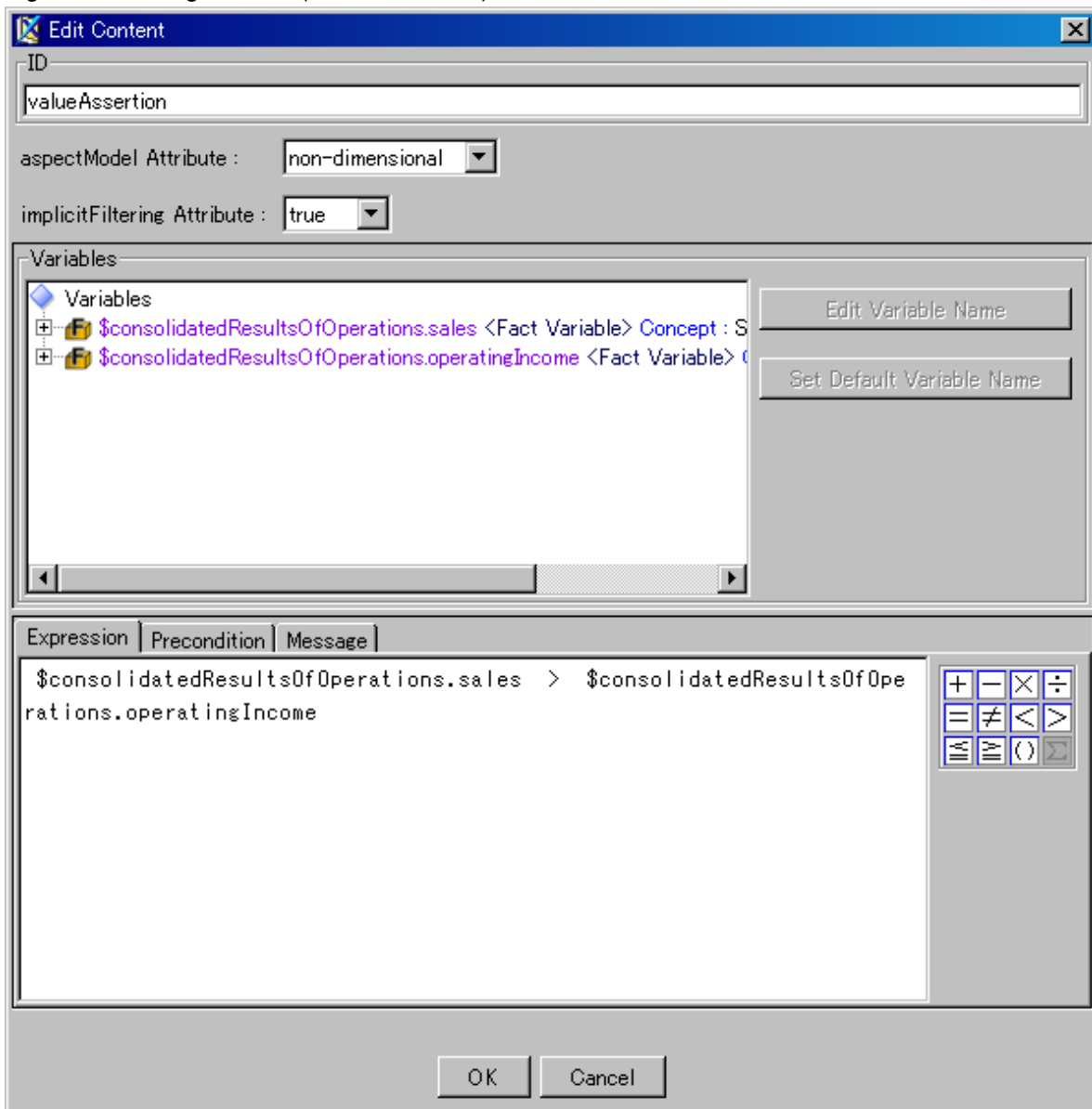


Item	Explanation
ID	The value used to uniquely identify the expression
aspectModel Attribute	In Formula specifications, the aspectModel attribute value
implicitFiltering Attribute	In Formula specifications, the implicitFiltering attribute value
Variables	Displays the variable associated with this expression. The "Edit Variable Name" button can be used to rename the variable. The "Set Default Variable Name" button can be used to rename the generated variable name.
Expression	Enter the expression to be used to verify existence. No change is required if you want to verify whether or not a value exists. Specify if you want to verify the number of values that exist.
Precondition	Specify if you want to attach conditions to execution of the expression.
Message	Define the message that corresponds to the validation result. Refer to Section 2.4.4, "Validation Message" for information on how to define the message.

2.4.3 Value verification (Value Assertion)

A value verification definition codes an expression that is used to verify whether or not specific conditions are met.

Figure 2.5 Editing Content (Value Assertion)



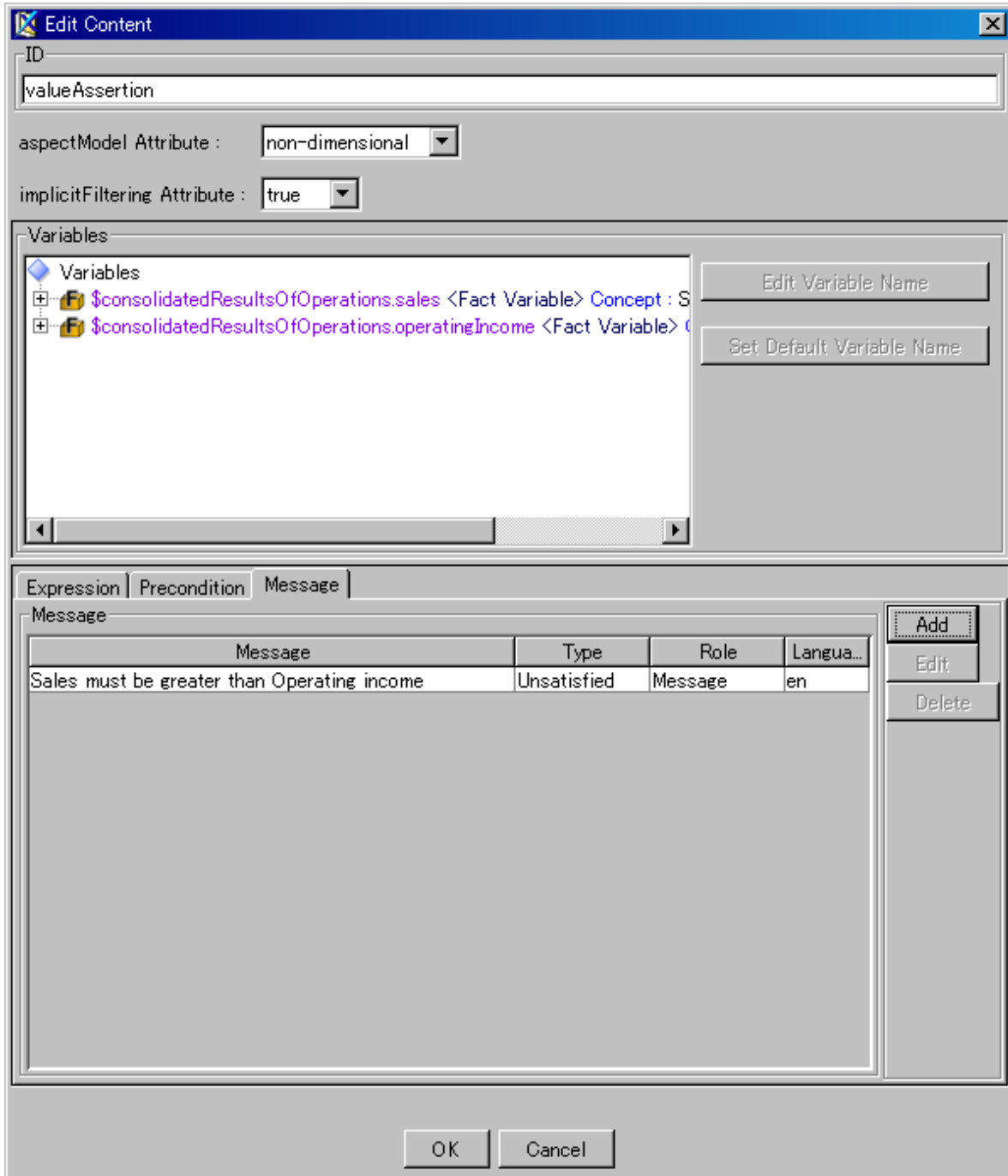
Item	Explanation
ID	The value used to uniquely identify the expression
aspectModel Attribute	In Formula specifications, the aspectModel attribute value
implicitFiltering Attribute	In Formula specifications, the implicitFiltering attribute
Variables	Displays the variable associated with this expression. The "Edit Variable Name" button can be used to rename the variable. The "Set Default Variable Name" button can be used to rename the generated variable name.
Expression	Enter the expression to be used to verify the value.
Precondition	Specify if you want to attach conditions to execution of the expression.

Message	Define the message that corresponds to the validation result. Refer to Section 2.4.4, "Validation Message" for information on how to define the message.
---------	--

2.4.4 Validation Message

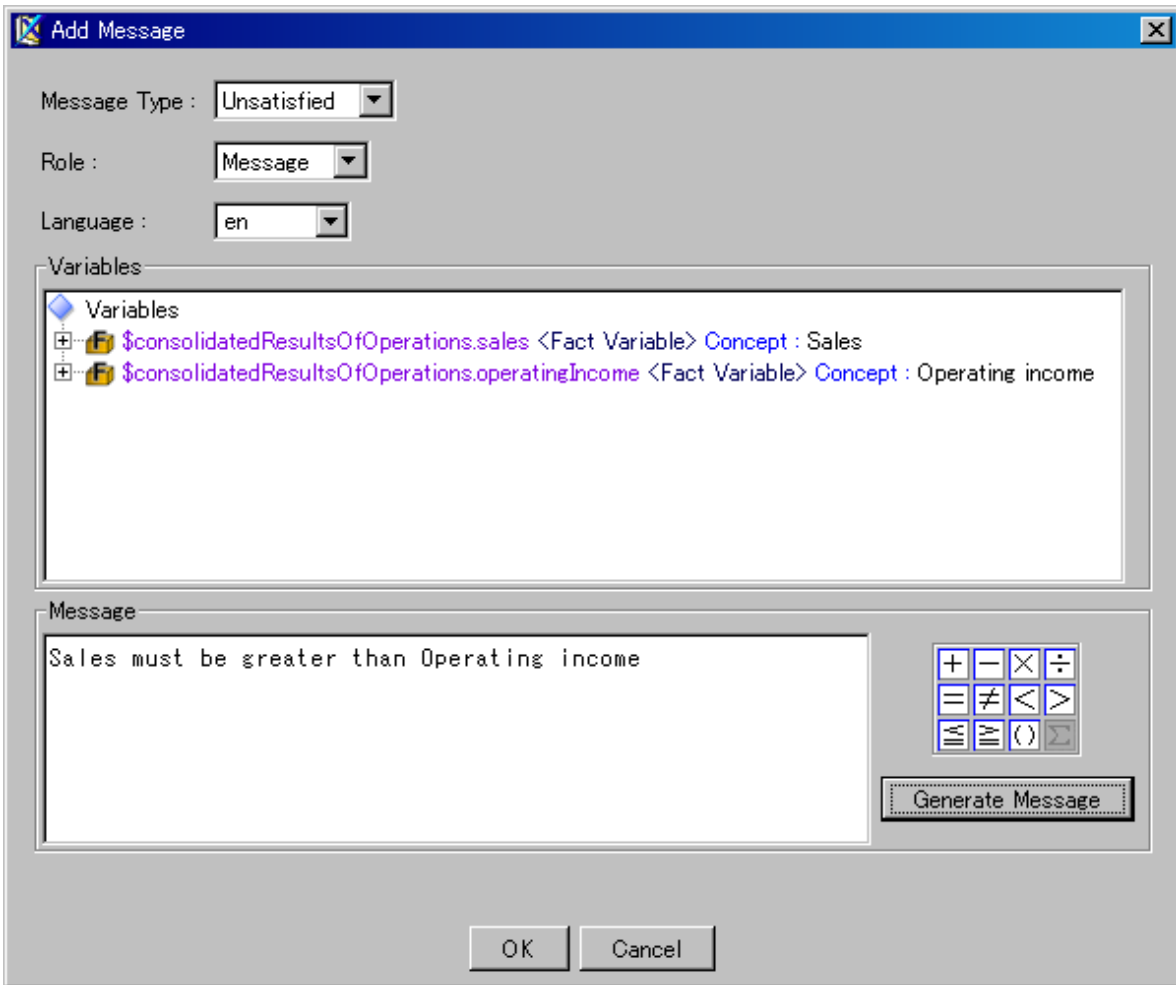
The message that corresponds to the validation result can be defined in the "Message" tab of the "Edit Content" dialog box for each validation.

Figure 2.6 "Message" Tab



By clicking the "Add", "Edit", and "Delete" buttons on the right of the "Message" tab, each message can be added, edited, or deleted respectively.

Figure 2.7 "Add Message" Dialog ("Edit Message" Dialog is the same)



Item	Explanation
Message Type	To define a message where validation has not been satisfied, select "Unsatisfied". To define a message where validation has been satisfied, select "Satisfied".
Role	Specify the message role. To define a standard message, specify "Message". To define a terse message, specify "Terse". To define a verbose message, specify "Verbose".
Language	Specify the message language.
Variables	The variables associated with the expression are displayed.
Message	Define the body of the message.
Generate Message	<p>The message can be generated from the expression automatically by clicking this button. When the language is not "ja", an English message will be generated.</p> <p>This button will be enabled when the expression is configured from the following elements:</p> <ul style="list-style-type: none"> Variables Four arithmetic operations Comparison operations Constants Parenthesis <p>When a Concept Name Filter in which a single concept is specified is associated with a variable, the variable part will be converted to the string expression defined in the concept. At this time, the</p>

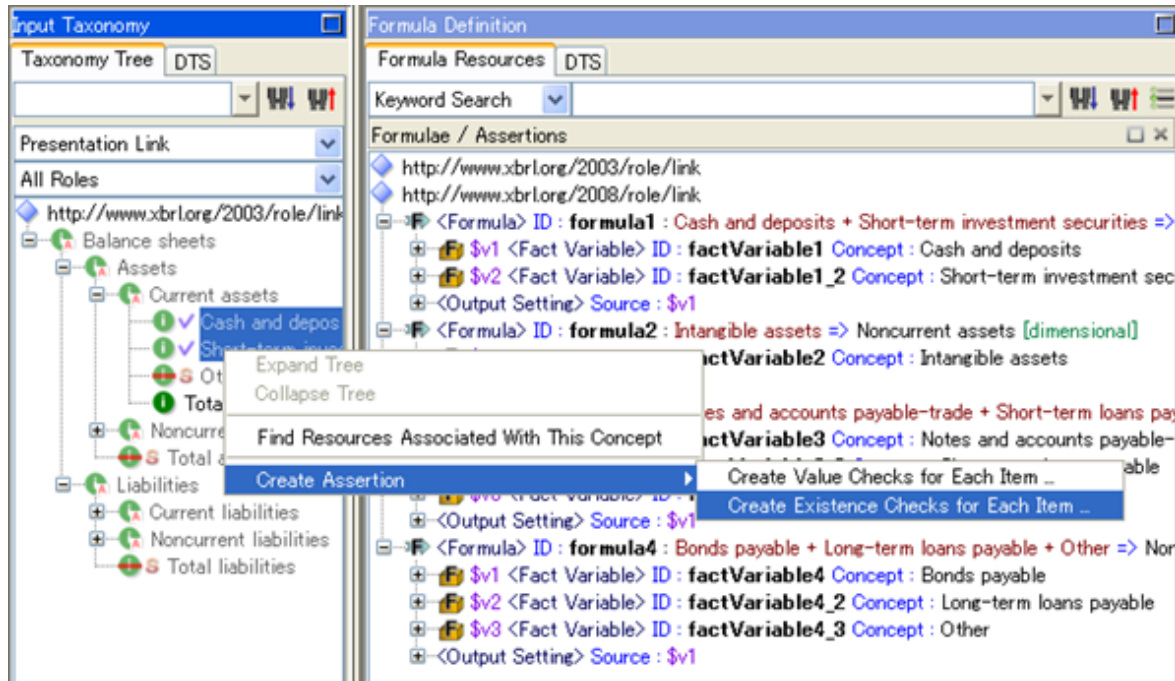
string follows the content defined in the "Input Taxonomy" tab of the "Editor View Setting" window, and the label language follows "Language" in this dialog box.

2.4.5 Create existence assertions for each item

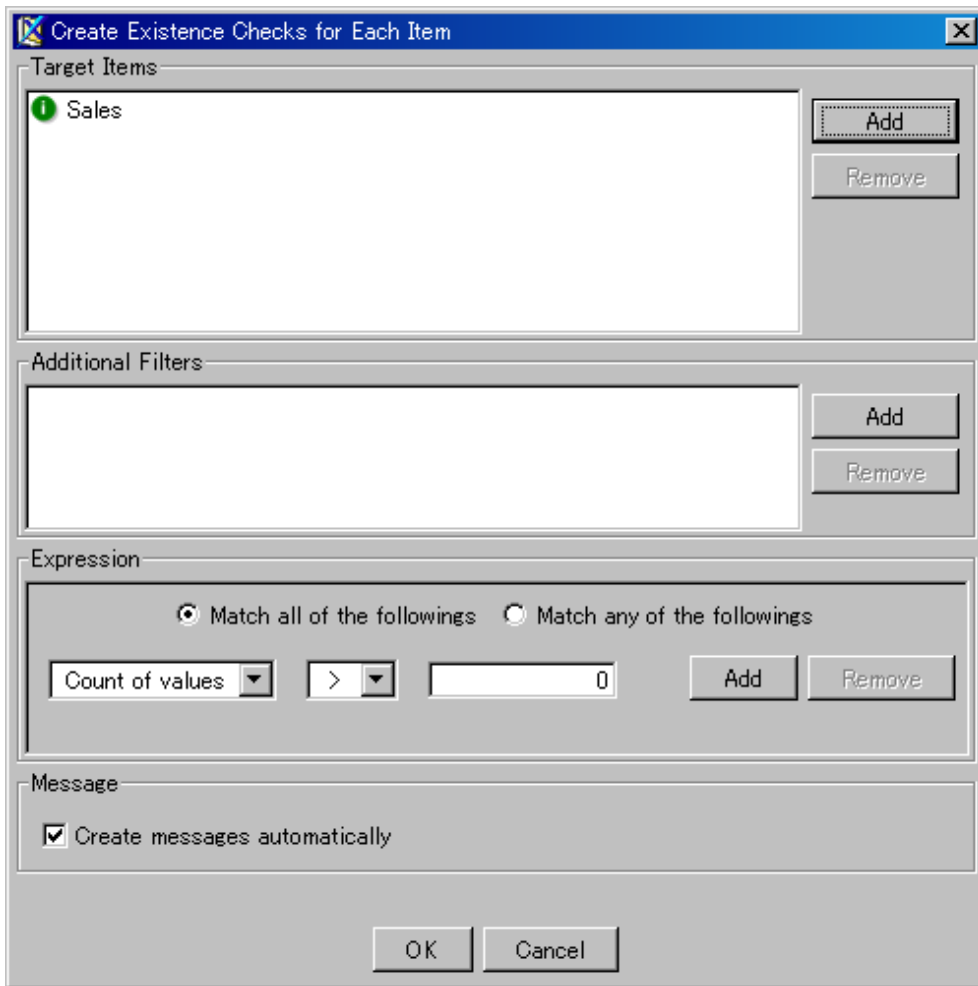
By using a wizard, you can define multiple "Existence Assertion".

To use a wizard, select items in the "Input Taxonomy" pane and right click and select "Create Existence Checks for Each Item ..." to open the "Create Existence Checks for Each Item".

Figure 2.8 Select a menu of Create Existence Checks for Each Item



The other condition can be specified in "Create Existence Checks for Each Item" dialog.



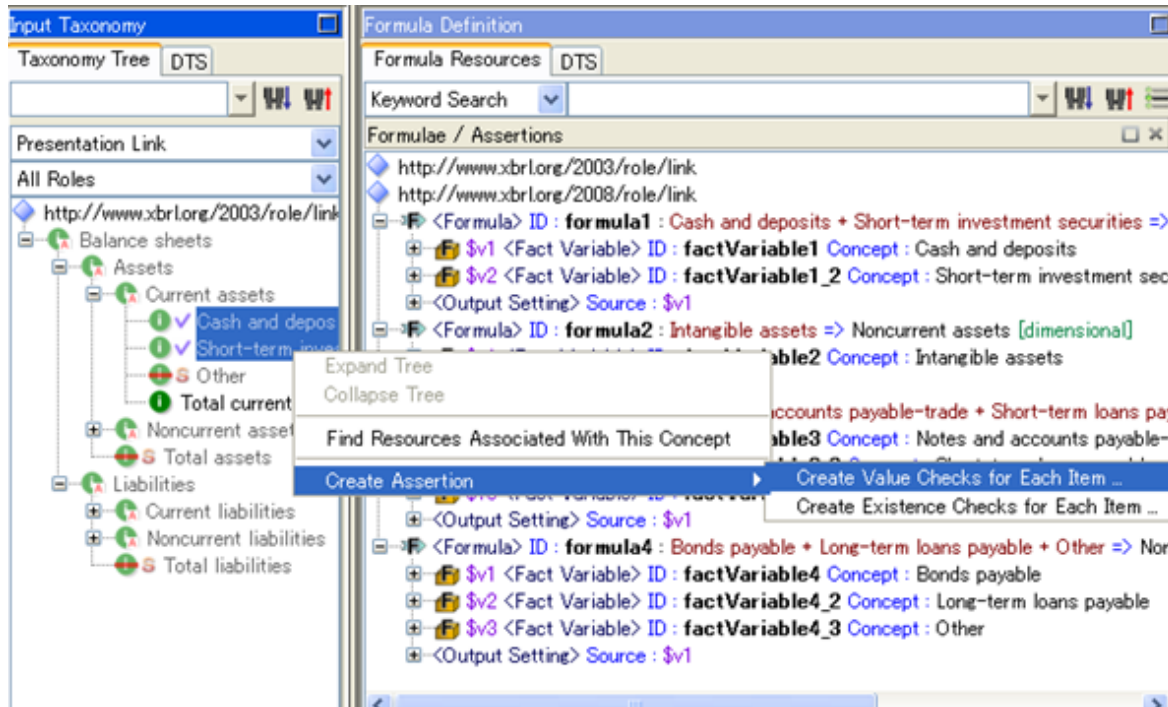
Item	Explanation
Target Items	Specify the check target items. You can add and remove items by "Add" and "Remove" button.
Additional Filters	Specify the additional filters. You can add and remove filters by "Add" and "Remove" button.
Expression	Specify the expression. You can add and remove expressions by "Add" and "Remove" button.
Create messages automatically	When this check box is selected, the message defined where validation has not been satisfied is generated automatically. The message language follows the language set in the "Input Taxonomy" tab in the "Editor View Setting" window. When the language is not "ja", an English message will be generated.

2.4.6 Create value assertions for each item

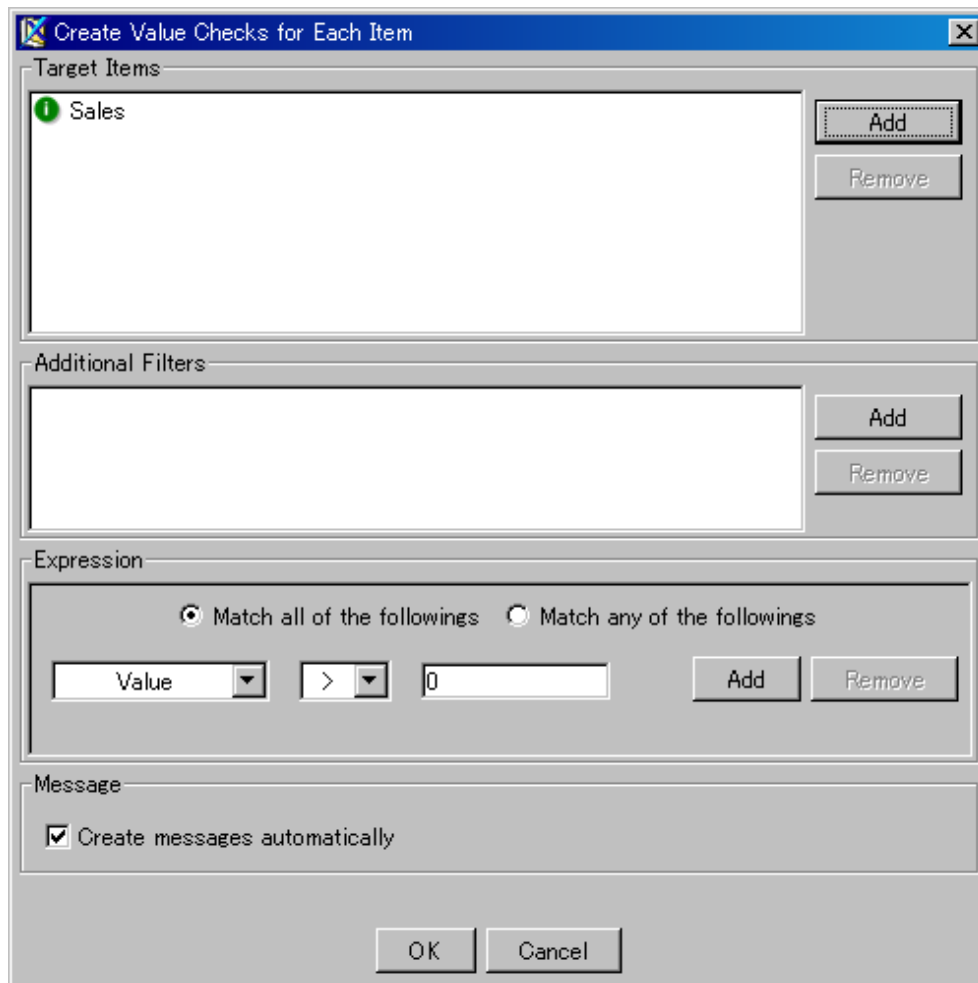
By using a wizard, you can define multiple "Value Assertion".

To use a wizard, select items in the "Input Taxonomy" pane and right click and select "Create Value Checks for Each Item ..." to open the "Create Value Checks for Each Item".

Figure 2.9 Select a menu of Create Value Checks for Each Item



The other condition can be specified in "Create Value Checks for Each Item" dialog.



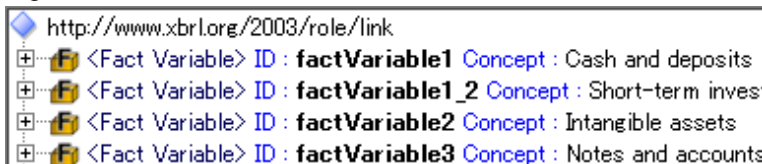
Item	Explanation
Target Items	Specify the check target items. You can add and remove items by "Add" and "Remove" button.
Additional Filters	Specify the additional filters. You can add and remove filters by "Add" and "Remove" button.
Expression	Specify the expression. You can add and remove expressions by "Add" and "Remove" button.
Create messages automatically	When this check box is selected, the message defined where validation has not been satisfied is generated automatically. The message language follows the language set in the "Input Taxonomy" tab in the "Editor View Setting" window. When the language is not "ja", an English message will be generated.

2.5 Variable Definitions

Define the variables to be used in expressions.

Variables that have already been defined can be referenced in the variables window of the formula pane. To associate a variable with an expression, select an already defined variable and drag and drop it to the expression.

Figure 2.10 Variables

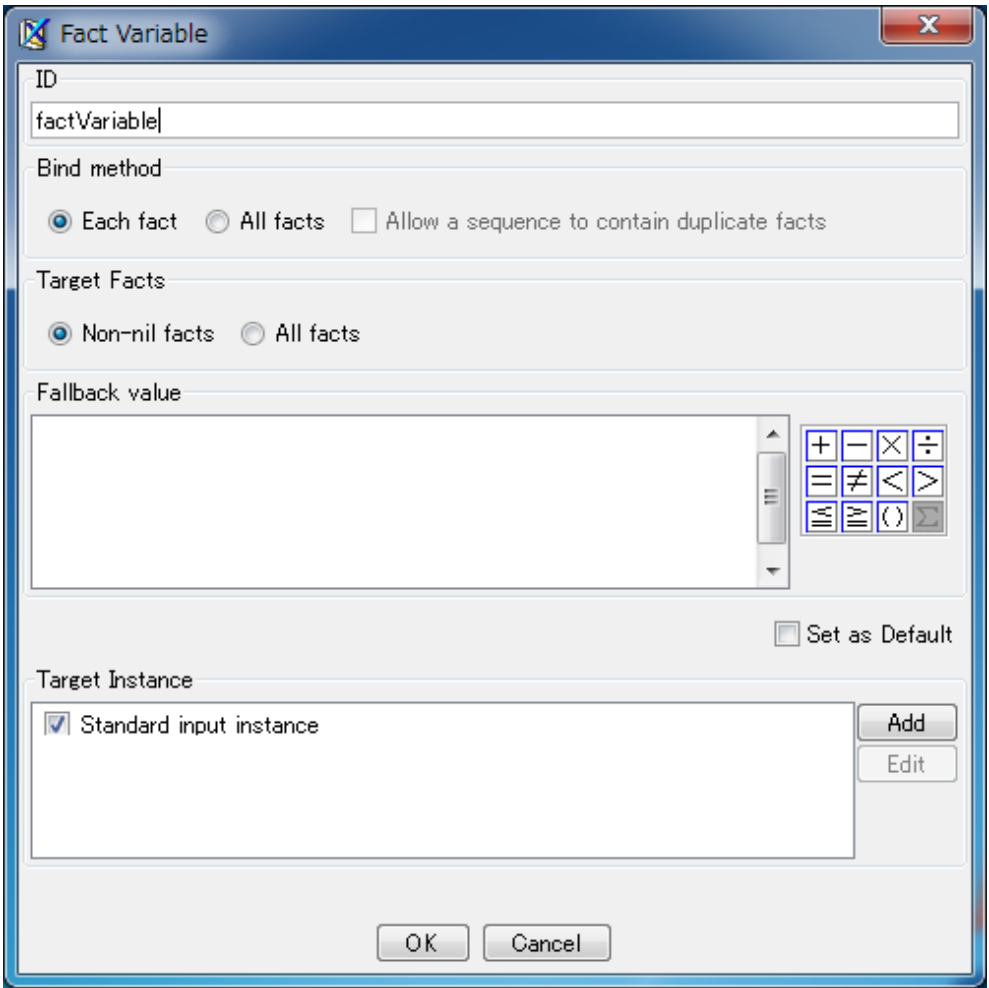


The following table shows the variable types. Select the extended link role in the variables window and right click to open the context menu. Then select "Add variable" from the context menu to enable specification of the variable you want to add.

Type	Explanation
Fact Variable	A variable that can be associated with a filter definition
General Variable	A variable used to specify the value associated with the variable in an expression on the basis of XPath 2.0 specifications. Refer to the Formula specifications or related specification documentation for details.
Parameter	A variable used to specify in a parameter a value associated with the variable. Refer to the Formula specifications or related specification documentation for details.

The following figures and tables show the definition windows for each type of variable and the content that can be defined.

Figure 2.11 Fact Variable



Item	Explanation
ID	Value used to uniquely identify the variable
Bind method	In Formula specifications, corresponds to the bindAsSequence attribute value and corresponds to "One fact" is "false".
Allow a sequence to contain duplicate facts	This corresponds to the matches attribute in the Formula specification. When this check box is selected, the setting will be "true". When this check box is not selected, the matches attribute will be unspecified. This check box is enabled when "All facts" has been specified in "Bind method".
Target Facts	In Formula specifications, corresponds to the nil attribute value and corresponds to "Non-nil facts" is "false".
Fallback value	In Formula specifications, corresponds to the fallback attribute value. If an item does not match the specified filter conditions, the value used in the calculation can be specified in an expression based on XPath 2.0 specifications.
Set as Default	After "Set as Default" checkbox is on and select "OK", the setting of "Bind method", "Target Facts", "Fallback value" is used as default.
Target Instance	Select the target input instance. In the initial state, only "Standard input instance" can be selected. Click the "Add" button to add a new target instance. Additionally, click the "Edit" button to change an instance's name. If the target instance that was added is no longer referenced by any calculation expression or Fact Variable, the target instance is automatically deleted.

Note that this item is not displayed in the Data Mapping Tool.

You can select "Edit Content" menu for multiple Fact Variables. The setting in the "Edit Content" is applied to the selected Fact Variables.

Figure 2.12 General Variable

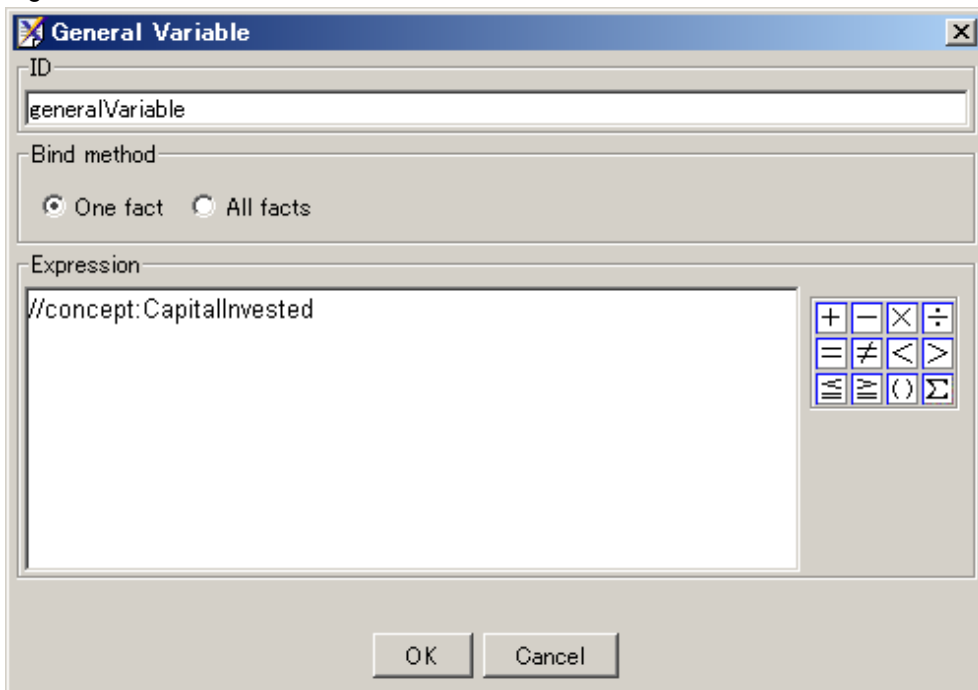
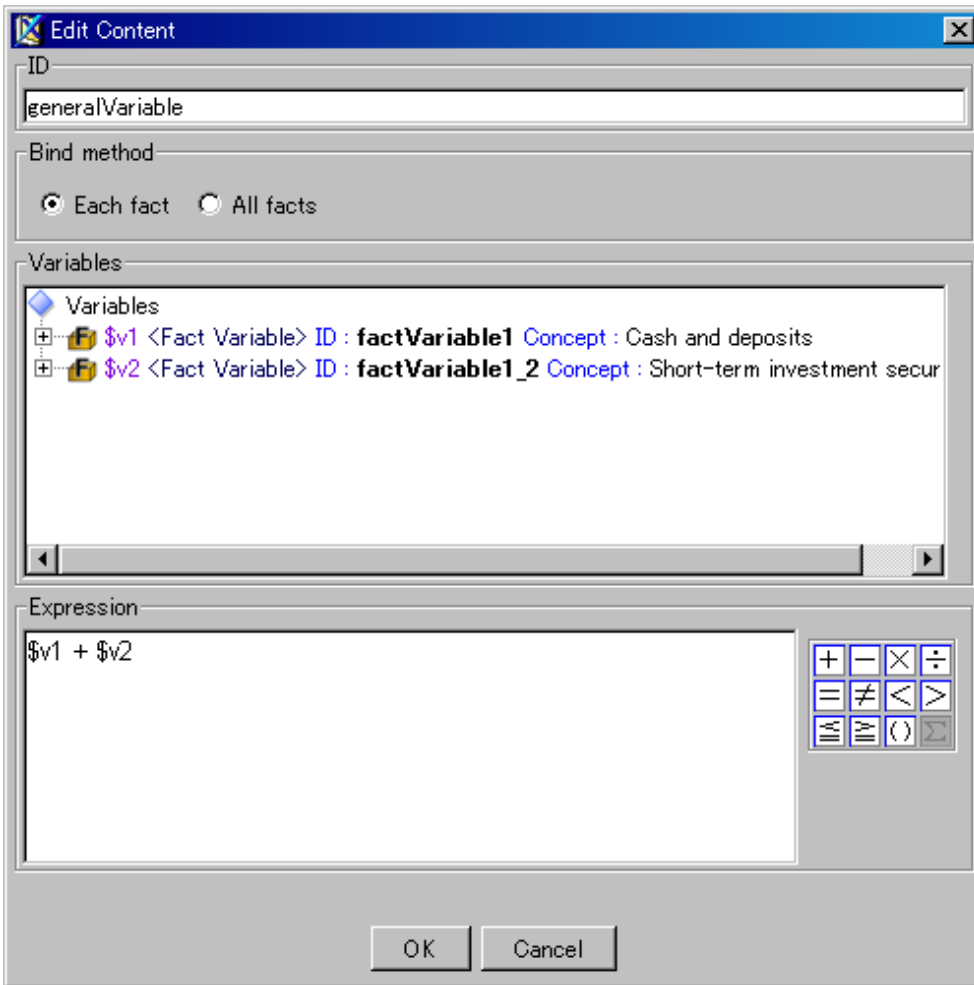
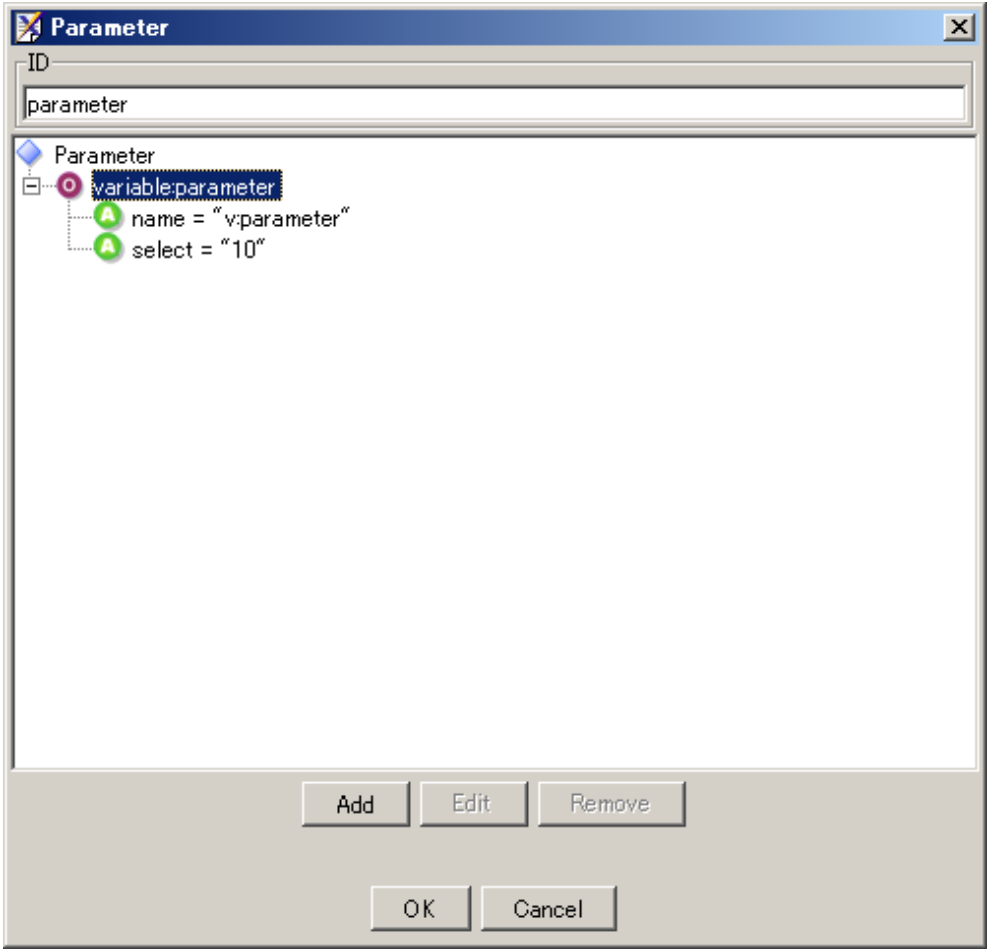


Figure 2.13 General Variable(related to formula/assertion)



Item	Explanation
ID	Value used to uniquely identify the variable
Bind method	In Formula specifications, corresponds to the bindAsSequence attribute value and corresponds to "One fact" is "false".
Variables	If the user adds a new General Variable to a formula/assertion, or the target General Variable is related to a formula/assertion, the variable list related to the formula/assertion is displayed.
Expression	Specifies the associated value in an expression based on XPath 2.0 specifications.

Figure 2.14 Parameter



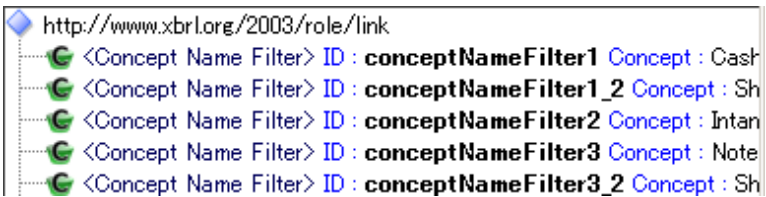
Item	Explanation
ID	Value used to uniquely identify the variable
(Tree view)	Specify any settable information in the "Add node" dialog that is displayed after the "Add" button is selected.

2.6 Filter Definitions

Define filters in order to specify associated values in variables. In general, filters are refinement conditions used to specify which input instance values are to be used.

Filters that have already been defined can be referenced in the filters window of the formula pane. A filter can be associated with a variable by selecting a filter that has already been defined and dragging and dropping it to the variable. A filter can also be associated with an expression by dragging and dropping it to a calculation expression or verification expression (except for Consistency Assertions).

Figure 2.15 Filters



The following table shows the filter types. Select the extended link role in the filters window and right click to open the context menu. Then select "Add filter" from the context menu to enable specification of the filter you want to add.

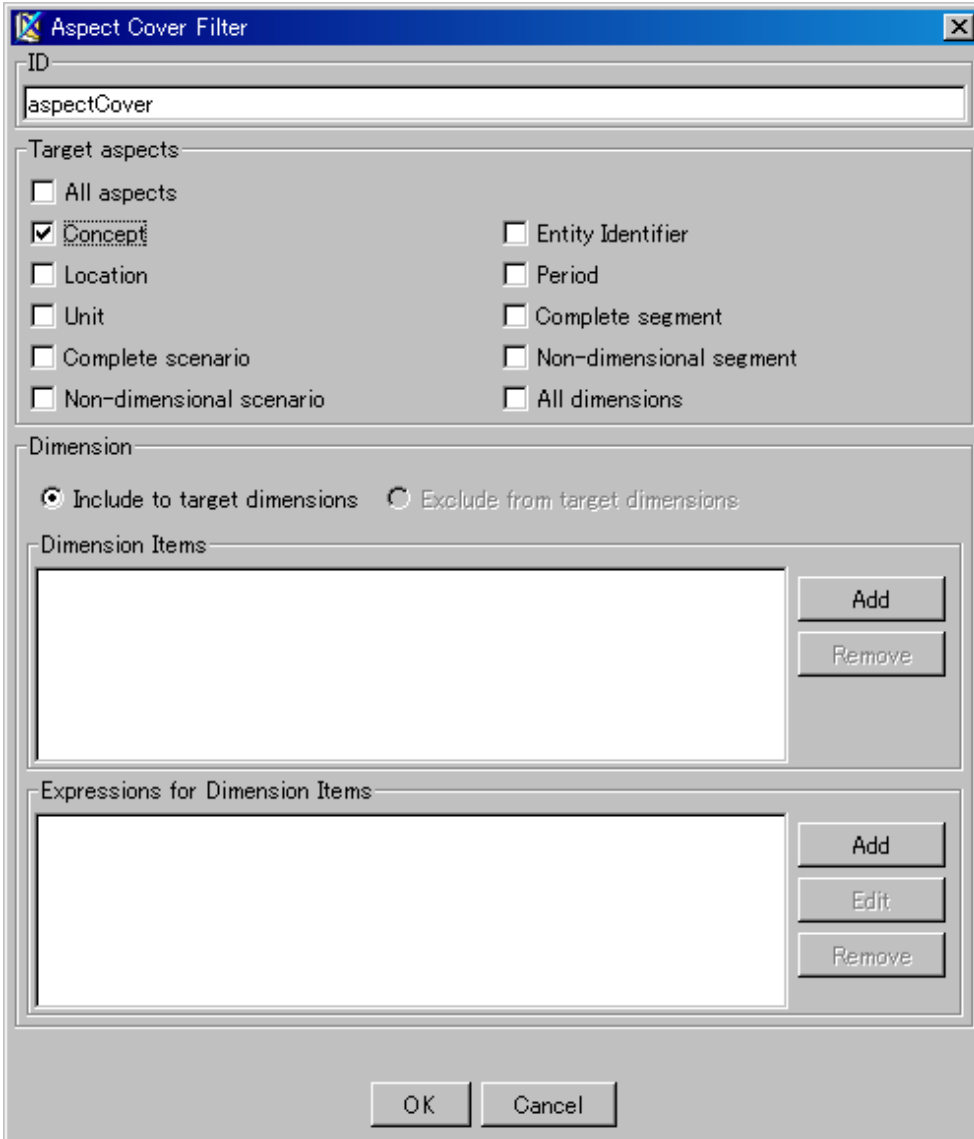
Type	Explanation
Aspect Cover Filter	This filter is for covering specific aspects. Refer to the Formula specification or related specification documentation for details.
Boolean Filter	This filter is for combining multiple filters with an AND condition or an OR condition. Refer to the Formula specifications or related specification documentation for details.
Concept Filter	This filter is for specifying element information as a refinement condition. A filter generated automatically by dragging and dropping an input taxonomy element is used as a "Concept Name Filter" specified as an element name refinement condition. Refer to the Formula specifications or related specification documentation for details.
Concept Relation Filter	This filter is for narrowing down values using relations for elements such as presentation links. Refer to the Formula specification or related specification documentation for details.
Dimension Filter	This filter is for specifying Dimension information as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Entity Filter	This filter is for specifying the Entity information of a context as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
General Filter	This filter is for specifying an expression based on XPath 2.0 specifications as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Match Filter	This filter is for using a different Fact Variable as a refinement condition and is specified specifically for an aspect. Refer to the Formula specifications or related specification documentation for details.
Period Filter	This filter is for specifying Period information as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Relative Filter	This filter is for using the aspect of another Fact Variable as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Segment Scenario Filter	This filter is for specifying a context Segment or Scenario as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Tuple Filter	This filter is for specifying Tuple information as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Unit Filter	This filter is for specifying Unit information as a refinement condition. Refer to the Formula specifications or related specification documentation for details.
Value Filter	This filter is for specifying information concerning a value as a refinement condition. Refer to the Formula specifications or related specification documentation for details.

The method used to define filters for large number of settings items is explained in the following section.

2.6.1 Aspect Cover Filter

The "Aspect Cover Filter" definition dialog box and settings items are shown below.

Figure 2.16 Aspect Cover Filter definition dialog



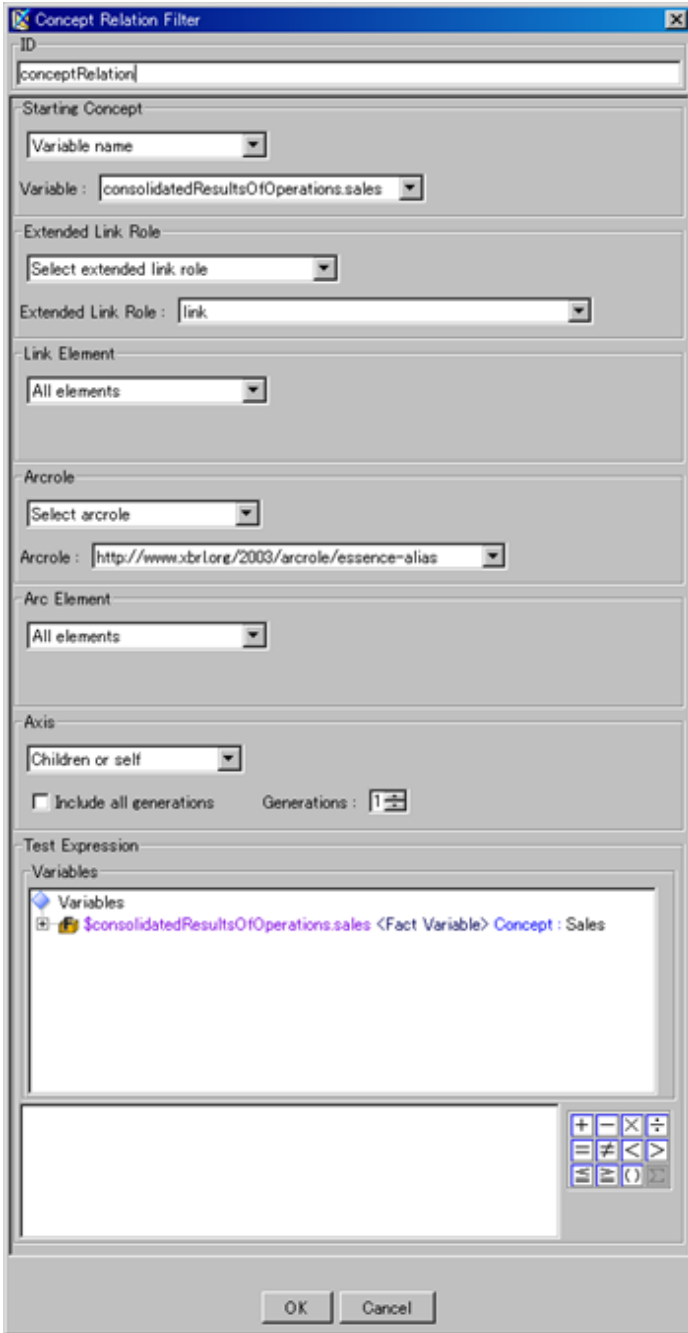
Item		Explanation
ID		Specify the filter ID.
Target Aspects	All aspects	When this check box is selected, all the aspects will be covered.
	Concept	When this check box is selected, the concept aspect will be covered.
	Entity Identifier	When this check box is selected, the entity identifier aspect will be covered.
	Location	When this check box is selected, the location aspect will be covered.
	Period	When this check box is selected, the period aspect will be covered.
	Unit	When this check box is selected, the unit aspect will be covered.
	Complete segment	When this check box is selected, the complete segment aspect will be covered.
	Complete scenario	When this check box is selected, the complete scenario aspect will be covered.
	Non-dimensional segment	When this check box is selected, the non-dimensional segment aspect will be covered.
	Non-dimensional scenario	When this check box is selected, the non-dimensional scenario aspect will be covered.

	All dimensions	When this check box is selected, all the dimension aspects will be covered.
Dimension	Include to target dimensions	When this is selected, the dimensions specified in "Dimension Item" and "Expressions for Dimension Item" will be covered. This radio button will be selected automatically when neither "All aspects" nor "All dimensions" are selected as a target aspect.
	Exclude from target dimensions	When this is selected, the dimensions specified in "Dimension Item" and "Expressions for Dimension Item" will not be covered. This radio button will be selected automatically when either "All aspects" or "All dimensions" are selected as a target aspect.
	Dimension Item	Specify the "Dimension Item" list. A dimension item can be added or removed using the "Add" and "Remove" buttons respectively.
	Expressions for Dimension Item	Specify the "Expressions for Dimension Item" list. A dimension item can be added, edited, or removed using the "Add", "Edit", and "Remove" buttons respectively.

2.6.2 Concept Relation Filter

The "Concept Relation Filter" definition dialog box and settings items are shown below.

Figure 2.17 Concept Relation Filter definition dialog



Item		Explanation
ID		Specify the filter ID.
Starting Concept	(List Box)	Specify the method used to select the starting concept. Select one of "Variable name", "Root concepts", "Specify concept name", or "Specify concept expression".
	Variable	Specify the variable name. This will be displayed if "Variable name" was selected as the method used to select the starting concept.
	Concept	Specify the concept. This will be displayed if "Specify concept name" was selected as the method used to select the starting concept.
	Expression	Specify the concept expression. This will be displayed if "Specify concept expression" was selected as the method used to select the starting concept.

Extended Link Role	(List Box)	Specify the method used to select the extended link role. Select one of "Select extended link role", "Specify extended link role", or "Specify extended link role expression".
	Extended Link Role	Specify the target extended link role. This will be displayed if "Select extended link role" or "Specify extended link role" was selected as the method used to select the extended link role.
	Expression	Specify the extended link role expression. This will be displayed if "Specify extended link role expression" was selected as the method used to select the extended link role.
Link Element	(List Box)	Specify the method used to select the link element. Select one of "All elements", "Specify element name", or "Specify element expression".
	Element Name	Specify the link element name. This will be displayed if "Specify element name" was selected as the method used to select the link element.
	Expression	Specify the link element expression. This will be displayed if "Specify element expression" was selected as the method used to select the link element.
Arcrole	(List Box)	Specify the method used to select the arcrole. Select one of "Select arcrole", "Specify arcrole", or "Specify arcrole expression".
	Arcrole	Specify the target arcrole. This will be displayed if "Select arcrole" or "Specify arcrole" was selected as the method used to select the arcrole.
	Expression	Specify the arcrole expression. This will be displayed if "Specify arcrole expression" was selected as the method used to select the arcrole.
Arc Element	(List Box)	Specify the method used to select the arc element. Select one of "All elements", "Specify element name", or "Specify element expression".
	Element Name	Specify the arc element name. This will be displayed if "Specify element name" was selected as the method used to select the arc element.
	Expression	Specify the arc element expression. This will be displayed if "Specify element expression" was selected as the method used to select the arc element.
Axis	(List Box)	Specify the axis for the concept relation.
	Include all generations	When this check box is selected, all the descendent or ancestor generations of the concept relation will be included. This can only be specified when the axis for the concept relation includes descendents or ancestors.
	Generations	Set the target number of generations for the descendents or ancestors of the concept relation. This can only be specified when the axis for the concept relation includes descendents or ancestors, and "Include all generations" is not selected.
Test Expression	Variables	List of variables relating to the Formula/Assertion to which the filter belongs.
	(Text Box)	Specify a test expression. This can be omitted.

2.6.3 Drag and drop definitions

Filters can be created using the drag and drop operation from the "Input Taxonomy" pane to the formula pane. The filters that can be created are listed below.

Drag target	Filter created	Remarks
Item	Concept Name Filter	
Tuple	Tuple Parent Filter/ Tuple Ancestor Filter	
Domain member	Explicit Dimension Filter	Enabled when "Dimension" is selected from the list in the "Input Taxonomy" pane
Explicit dimension item	Explicit Dimension Filter	

Typed dimension item	Typed Dimension Filter
----------------------	------------------------

The behavior varies depending on where the drag target is dropped, as described below.

Drop destination	Behavior
Calculation expression or verification expression (except for Consistency Assertions)	The filter is created, and the new Fact Variable associated with it is also associated with the calculation expression and the verification expression. Note that if the drag target is dropped while holding down the Ctrl key, the filter is directly associated with the calculation expression and the verification expression of the drop destination.
Fact Variable/Boolean Filter	The filter is created and associated with the Fact Variable/ Boolean Filter of the drop destination.
Variables window	The filter is created, and the new Fact Variable associated with it is added.
Filters window	The filter is created, and added.

When an item or a domain member is dragged and dropped using "Resource Reuse Setting" on the "Edit" menu, if the filter definition is the same, it is possible to reuse that filter or switch so that a new filter is always created.

2.7 Other Operations

2.7.1 Deleting an association

To delete an association between definitions, such as the association between a verification expression and a calculation expression, the association between an expression and a variable, or the association between a variable and a filter, select the children of the association you want to delete in the hierarchical structure displayed in the "Formula" pane, and right click to open the context menu. The association can be changed by selecting "Delete association" in the context menu.

2.7.2 Deleting a definition

To delete a definition, select the definition that you want to delete and right click to open the context menu. The definition can be deleted by selecting "Delete definition" in the context menu.

2.7.3 Setting a filter association

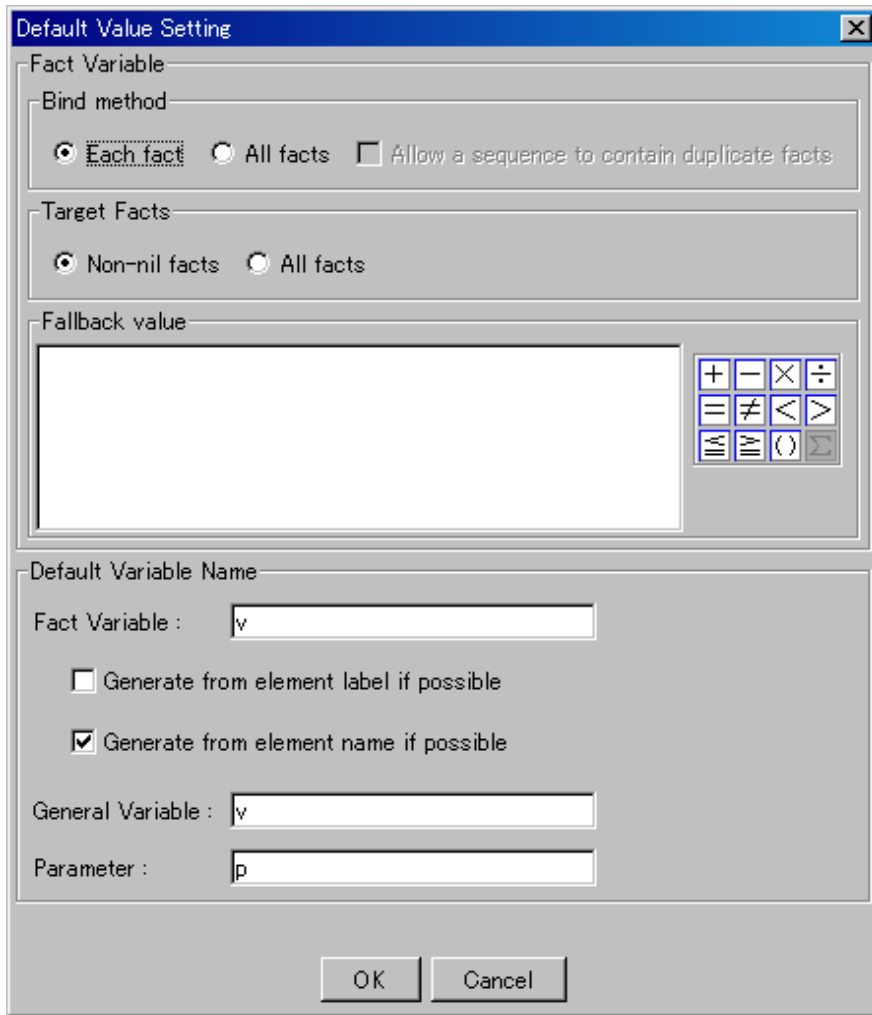
The following table shows the settings that can set when associating a variable or similar with a filter.

Menu name	Explanation
Edit cover attribute	Select this menu item to toggle between the "true" and "false" values for the cover attribute. "uncovered" corresponds to "false". Refer to the Formula specifications or related specification documentation for details.
Edit complement attribute	Select this menu item to toggle between the "true" and "false" values for the complement attribute. "complement" corresponds to "true". Refer to the Formula specifications or related specification documentation for details.

2.7.4 Setting a default value

To specify a default value, select "Default Value Setting" in the edit menu. This setting is used when you create a new variable.

Figure 2.18 Default Value Setting



Setting of Fact Variable

The value of "Bind method", "Target Facts", "Fallback value" defined in this dialog is used as default setting.

Item	Explanation
Bind method	In Formula specifications, corresponds to the bindAsSequence attribute value and corresponds to "One fact" is "false".
Allow a sequence to contain duplicate facts	This corresponds to the matches attribute in the Formula specification. When this check box is selected, the setting will be "true". When this check box is not selected, the matches attribute will be unspecified. This check box is enabled when "All facts" has been specified in "Bind method".
Target Facts	In Formula specifications, corresponds to the nil attribute value and corresponds to "Non-nil facts" is "false".
Fallback value	In Formula specifications, corresponds to the fallback attribute value. If an item does not match the specified filter conditions, the value used in the calculation can be specified in an expression based on XPath 2.0 specifications.

Setting of Default Variable Name

The value of "Default Variable Name" defined in this dialog is used as default variable name.

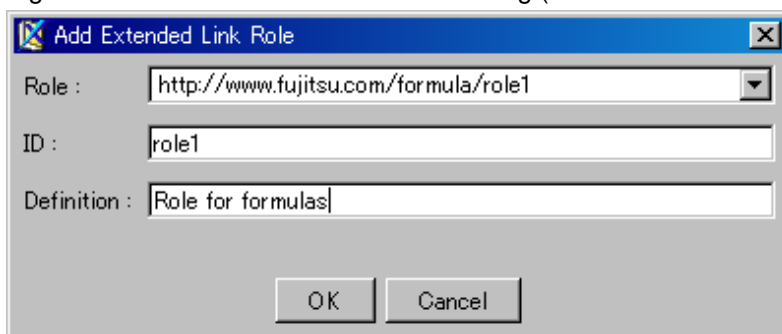
Item	Explanation
------	-------------

Fact Variable	"v" is set at the first time. If Fact Variable is related to a formula/assertion, this value and sequential number is set as the default variable name.
Generate from element label if possible	When this checkbox is on, the name of a Fact Variable will be the label of the element of Concept Name Filter if the Fact Variable is related to a Concept Name Filter. If both this checkbox and the following "Generate from element name if possible" is on, this setting is selected. The label is used by the label role and language defined in the "Input Taxonomy" tab of "Editor View Setting"(4.1 Setting the Editor View). If the label contains the characters which can not be used as variable name like space character, hyphen(-) and so on, the character is replaced as "_". If the label starts the character which can not be used as the start character of the variable name like number, the character "_" is inserted as the first character.
Generate from element name if possible	When this checkbox is on, the name of a Fact Variable will be the name attribute value of the element of Concept Name Filter if the Fact Variable is related to a Concept Name Filter. If not, the name of a Fact Variable is generated according to the "Fact Variable" field setting.
General Variable	"v" is set at the first time. If General Variable is related to a formula/assertion, this value and sequential number is set as the default variable name.
Parameter	"p" is set at the first time. If Parameter is related to a formula/assertion, this value and sequential number is set as the default variable name.

2.7.5 Extended Link Role Definition

The extended link role can be added, edited, or removed using the right-click menu of the formula pane. Using the extended link role, formulae and assertions can be defined as separate categories.

Figure 2.19 "Add Extended Link Role" dialog ("Edit Extended Link Role" is the same)



Item	Explanation
Role	Specify the extended link role to be defined.
ID	Specify the ID of the extended link role to be defined.
Definition	Specify the legend of the extended link role to be defined. This field can be omitted.

To create a formula, assertion, variable, or filter that will belong to a specific extended link role, right-click on the target extended link role of the formula pane then select each additional menu.

2.8 Output Definitions

Use the "Output setting" tab of the "Edit Content" dialog box to define the output definitions for a calculation expression.

Figure 2.20 Output Settings

The following table shows the information that can be referenced and edited from the Output Setting tab.

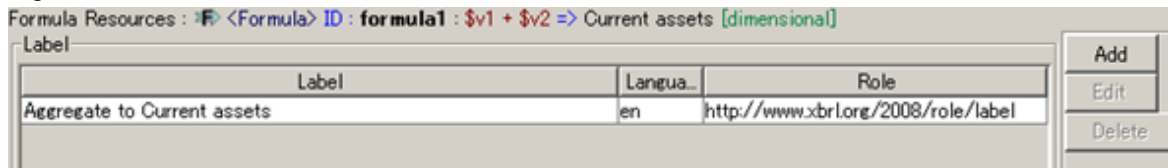
Item	Explanation
source Attribute	In Formula specifications, the source attribute value of the formula element.
precision/decimals Attribute	Specify the precision attribute or the decimals attribute of the output item.
Aspects	In Formula specifications, this is the aspects element information. Specify the output destination information. When an expression is created by dragging and dropping from an input taxonomy element to an output taxonomy element, an aspect that specifies the output taxonomy element name is generated. To add an aspect, select the "Add Aspect" button and select the type of aspect you want to add.
Target Instance	Select the target output instance. In the initial state, only "Standard output instance" can be selected. Click the "Add" button to add a new target instance. Additionally, click the "Edit" button to change an instance's name. If the target instance that was added is no longer referenced by any calculation expression or Fact Variable, the target instance is automatically deleted. Note that this item is not displayed in the Data Mapping Tool.

2.9 Labels

Under Formula specifications, labels can be set for expressions, variables, and filters. This tool enables labels to be set for expressions and variables, and the labels can be used in displays.

To set a label, use the "Label" tab in the "Information" pane.

Figure 2.21 "Label" tab

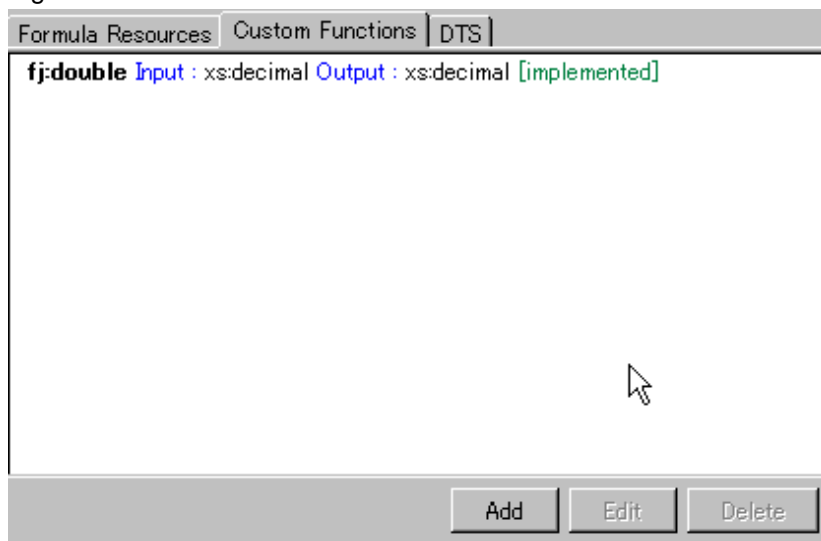


Select the target for which you want to set a label, then select the "Add" button of the "Label" tab. The "Add label" dialog is displayed. Specify the label content, language, and role.

2.10 Custom Functions

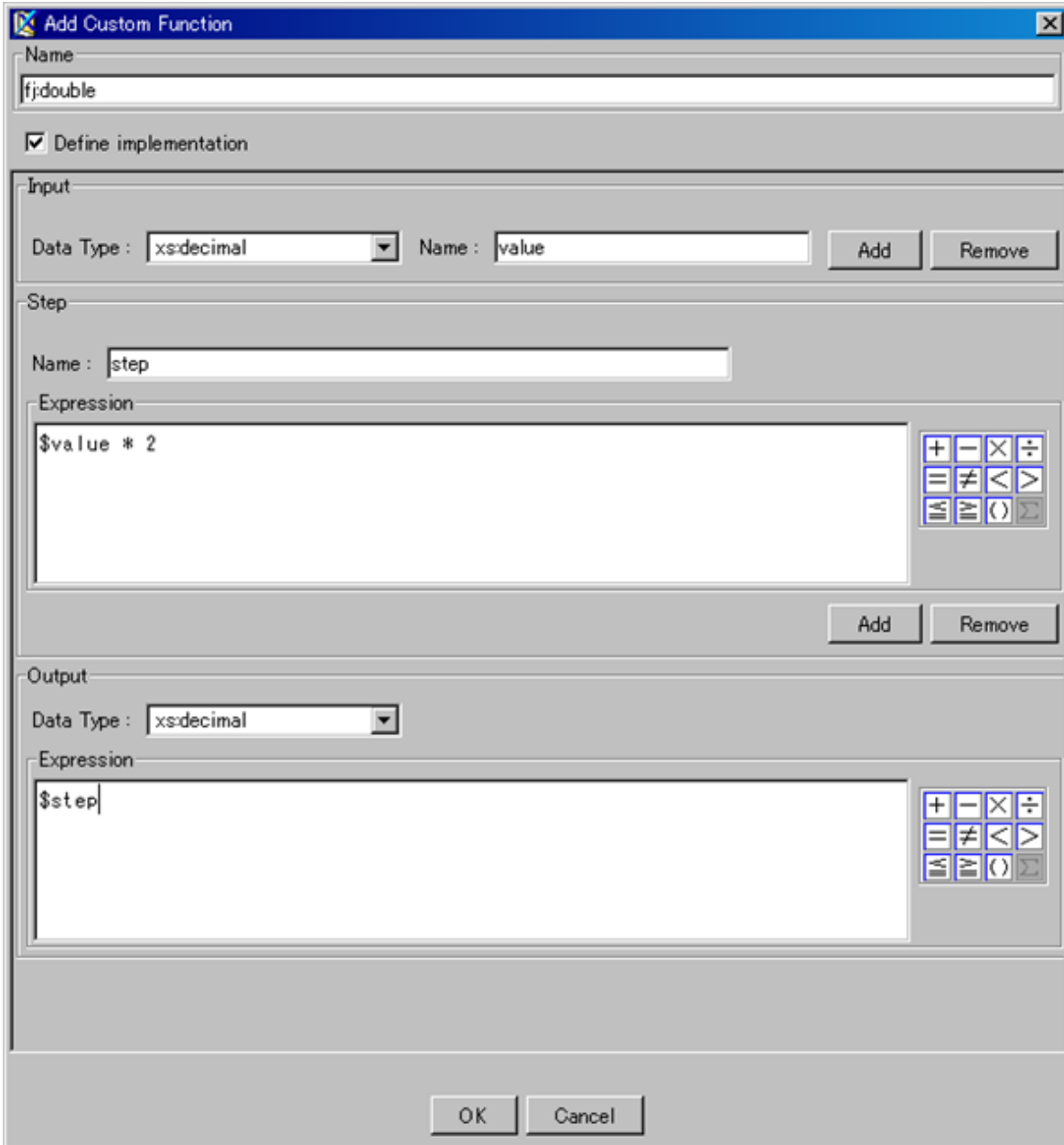
User-specific functions can be defined in the "Custom Functions" tab. The functions that are defined can be used with the calculation and validation expressions. In the "Custom Functions" tab, a list of the defined custom functions is displayed. Additionally, a custom function can be added, edited, or deleted using the right-click menu or the buttons in the bottom right of the tab.

Figure 2.22 "Custom Functions" tab



When the custom function is added or edited, the following dialog box is displayed.

Figure 2.23 "Add Custom Function" dialog box ("Edit Custom Function" is the same)



Item		Explanation
Name		Specify the custom function name with a prefix. Names without a prefix cannot be used.
Define implementation		Select this check box to define the custom function interface and implementation. When this check box is not selected, only the interface will be defined.
Input		Define the custom function input argument. Multiple arguments can be defined, or the argument can also be omitted.
	Name	Specify the argument data type.
	Expression	Specify the argument name. This is only displayed when the "Define implementation" check box is selected.
	Add	Add the argument.
	Remove	Remove the argument.

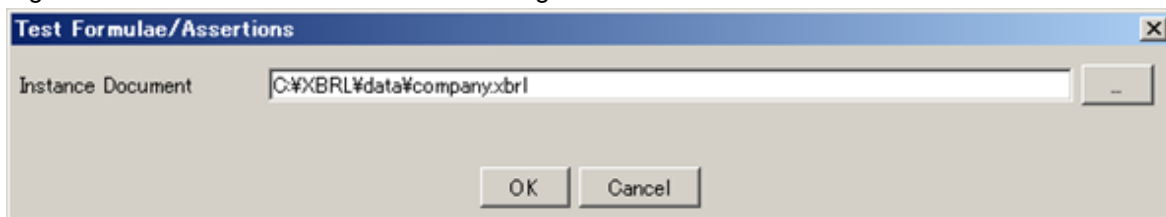
Step		This is used to assign the current result of the calculation to the variable. Multiple steps can be defined, or the argument can also be omitted. Note that the step is only displayed when the "Define implementation" check box is selected.
	Name	Specify the step name.
	Expression	Specify the step calculation expression.
	Add	Add the step.
	Remove	Remove the step.
Output	Data Type	Define the output data type.
	Expression	Define the output calculation expression. This is only displayed when the "Define implementation" check box is selected.

2.11 Checking Definition Operation

A created definition can be applied to an instance to check its operation.

When "Test Formulae/Assertions" is selected from the Tools menu, the "Test Formulae/Assertions" dialog is opened. In this dialog, specify the instance to which the definition is to be applied. Note that when there are multiple target input instances, their corresponding fields are displayed. Specify each instance. However, when an instance is both target input and output instance, no field is displayed.

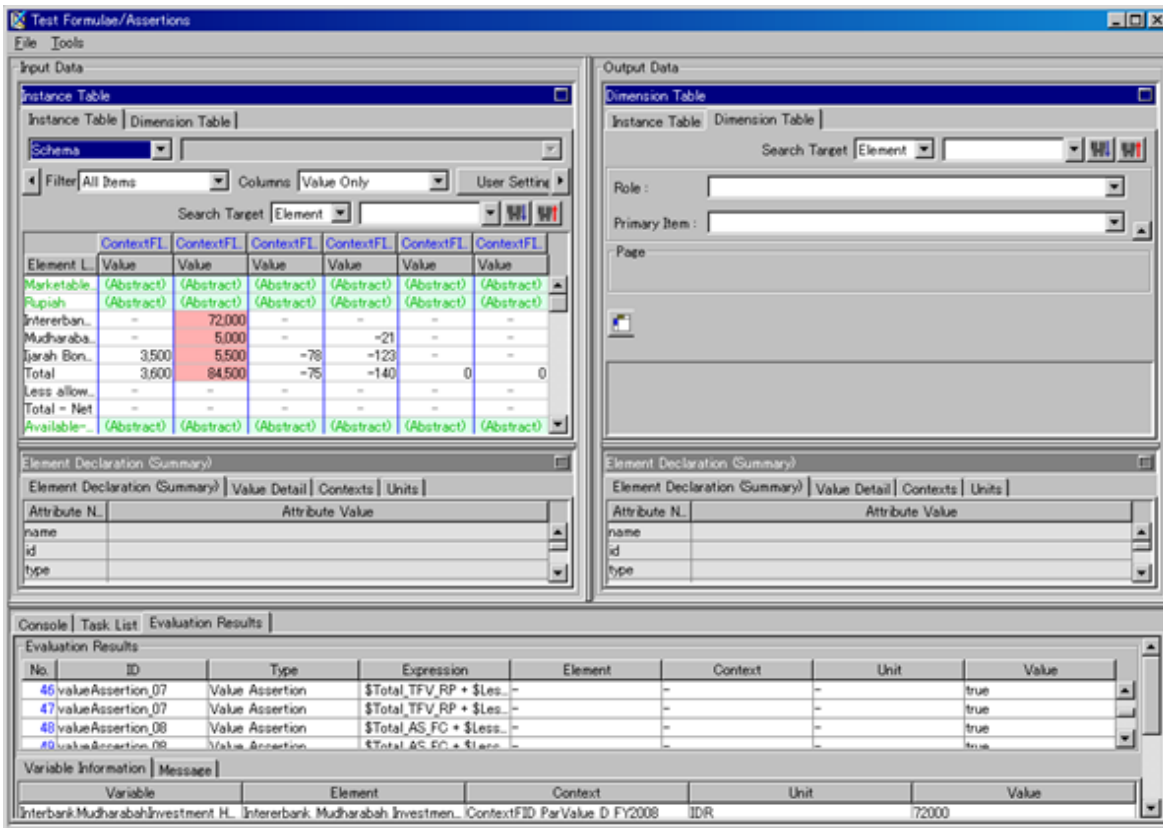
Figure 2.24 "Test Formulae/Assertions" dialog



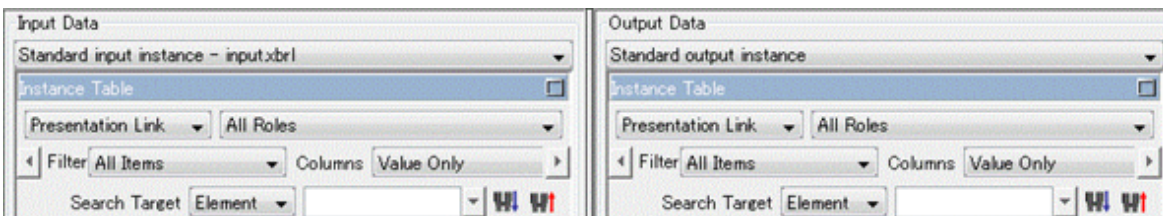
When "OK" is clicked, the definition currently being edited is applied to the specified instance and the application results are displayed.

Note that, when the formula and assertion definitions are mixed, only formulae will be evaluated immediately after the "OK" button is clicked. To evaluate the assertions, select "Evaluate Assertion" in the "Test Formulae/Assertions" dialog box.

Figure 2.25 Application results display



In the application results window, the input instance is displayed on the left, and the application results instance is displayed on the right. If there are multiple target input or output instances, a list box appears and the instances to be displayed can be switched, as shown below.



The "Evaluation results" tab in the lower part of the window displays the expression evaluated by the apply process and the variable values assigned to each expression at the time of evaluation. The user can refer to this information to check the expression that was operating and check whether or not the variables assigned at that time were as expected.

If a parameter is related to formula or assertion, the dialog to input parameter value is displayed.

Figure 2.26 Parameter Setting dialog

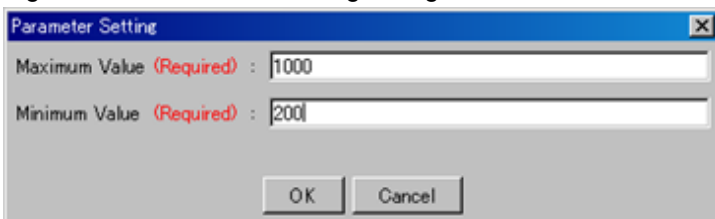


Table 2.1 Menus and submenus

Menu	Submenu	Description
File	Save Console Messages	"All": Saves to a file all the messages that are output in the console of "Information" pane.

		"Error Messages": Saves to a file the error messages that are output in the console of "Information" pane.
	Save Output Instance	Saves the instance displayed in "Output Data".
Tools	Update	Reloads the created definitions. It is used when you made any updates to the definitions after the testing.
	Evaluate Formulae	Reloads the instance document and applies only calculation formula definitions to the instance. It is used when you made any updates to the instance.
	Evaluate Assertion	Reloads the instance document and applies only assertion formula definitions to the instance. It is used when you made any updates to the instance.

The messages with the role "Message" of the selected assertion result in the evaluation results are shown in the "Message" tab.

Figure 2.27 Message tab

Variable Information		Message
Message		Language
Sales must be greater than Operating Income		en

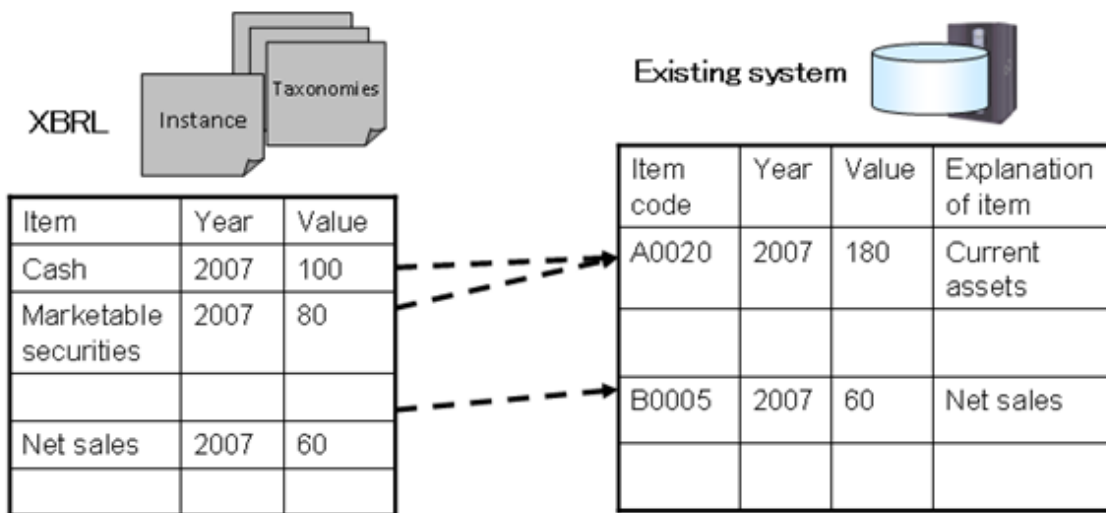
Chapter 3 Creating and Editing Data Mapping Definitions

This chapter describes the procedure for creating data mapping definitions. With the exception of opening an existing file, the editing procedure is the same, so refer to the description of how to create these definitions.

3.1 Overview of Functions

The data mapping function is used to store instance information in a database (RDB) in existing systems. Generally, in existing systems, financial items are managed using a data format that differs from that of taxonomies. The following figure is a representative diagram showing the difference.

Figure 3.1 Differences between XBRL and an existing system data format



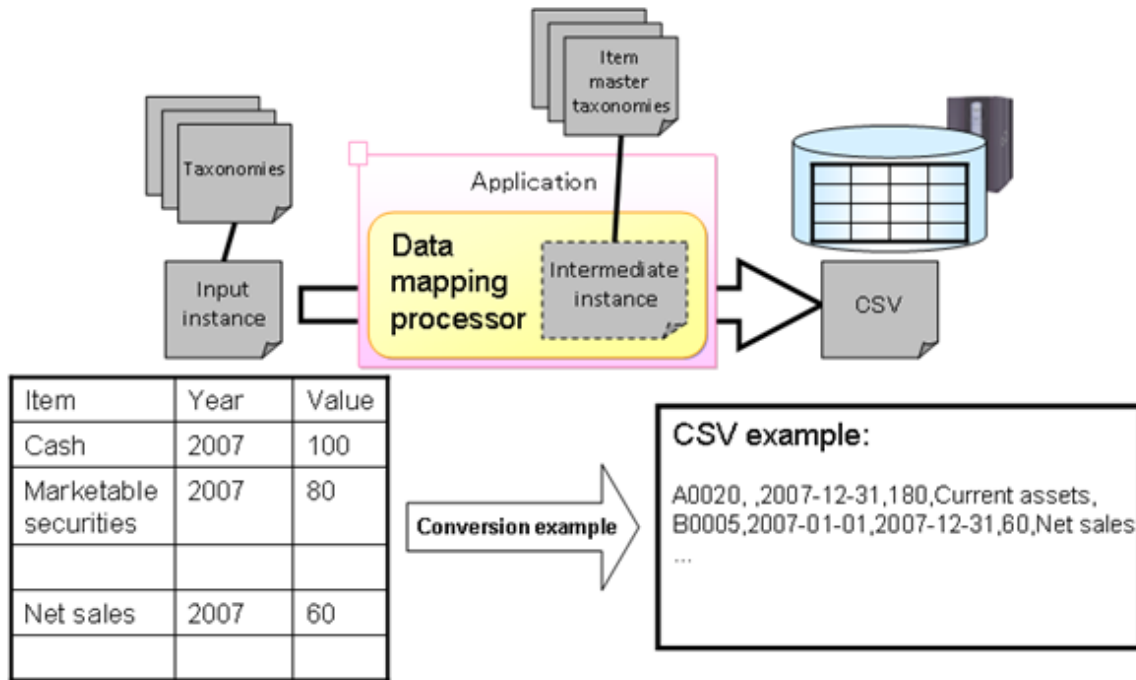
For these cases, the data mapping functions provide a function for converting instances to a format (CSV) that facilitates storage in a database.

The data mapping function uses the following processing flow to achieve the above:

- Conversion from XBRL to XBRL
An instance (hereafter referred to as an input instance) is converted to an instance (hereafter referred to as an intermediate instance) that conforms to a taxonomy (hereafter referred to as an item master taxonomy) that is based on the item master that the existing system uses to manage data.
- Conversion from XBRL to CSV
The intermediate instance is converted to CSV.

The following figure is a representative diagram.

Figure 3.2 Flow of conversion by the data mapping function

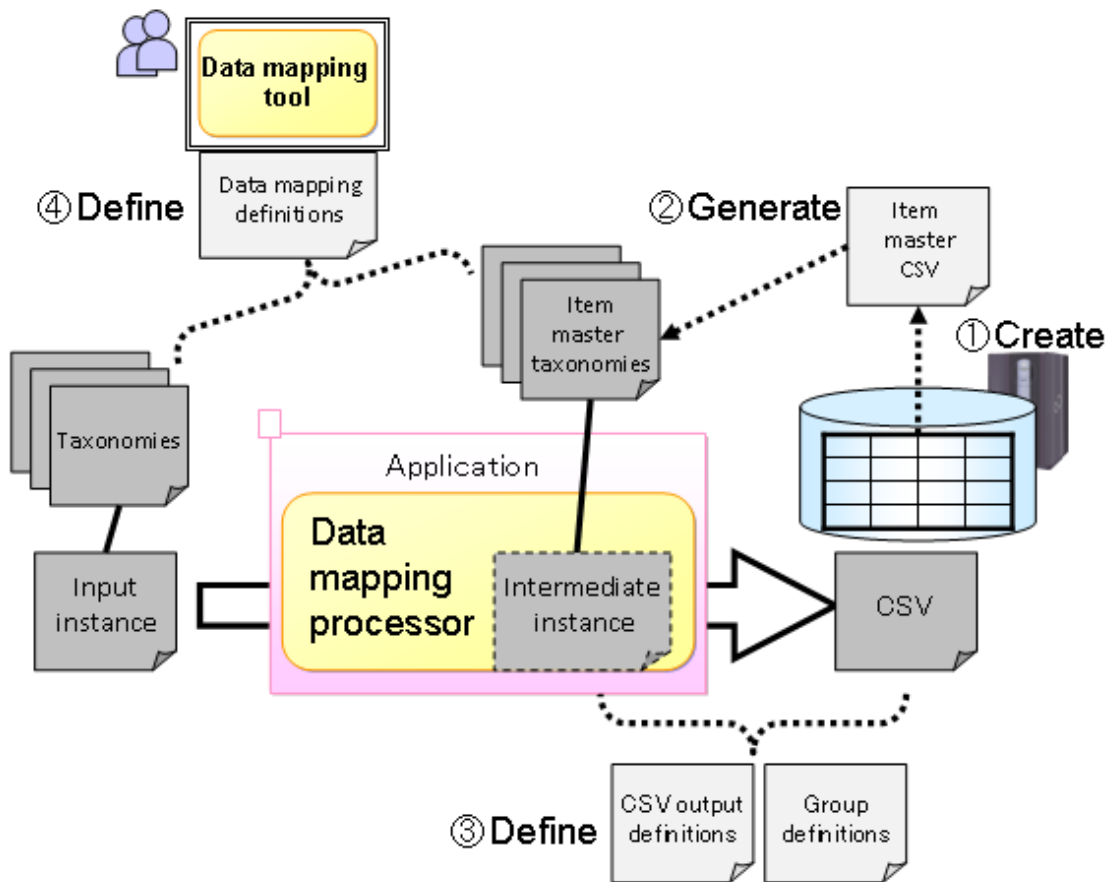


In order to achieve the above, the following are created:

- Item master taxonomy
The item master that is managed by the existing database is created in CSV format (hereafter referred to as the item master CSV), and then an item master taxonomy is created based on this CSV. Refer to Section 3.2, "Creating an Item Master Taxonomy", for details.
- CSV output definition
This definition is used to convert the intermediate instance to CSV format. Refer to Section 3.3, "Creating a CSV Output Definition", for details.
- Group definition
This is a definition of context groups which are referenced from some output items in a CSV output definition. This definition is used to output row-orientated data, which is explained later. Refer to Section 3.5, "Creating a Data Mapping Definition", for details.
- Data mapping definition
This definition is used to convert the input instance to the intermediate instance. Refer to 3.6, "Creating a Data Mapping Definition", for details.

The following figure is a representative diagram showing the relationship between the created definitions and conversion.

Figure 3.3 Relationships between definitions



3.1.1 Column-orientated data and row-orientated data

The data mapping function can output the following two types of CSV.

- Column-orientated data

This is a data format where item values are recorded in columns. Only one item value of an intermediate instance is recorded in one row of a CSV. That is, the number of rows of the CSV is equal to that of items of the instance. This is a default data format. (Refer to Section 3.3, "Creating a CSV Output Definition", for details)

[Sample: column-orientated]

```
2008-03-31,JPY,101,20000000000,Current Assets
2008-06-30,JPY,101,90000000000,Current Assets
2008-03-31,JPY,102,50000000000,Fixed Assets
2008-06-30,JPY,102,30000000000,Fixed Assets
```

- Row-orientated data

This is a data format where item values are recorded in rows. Multiple item values of an intermediate instance are recorded in one row of a CSV. The number of rows of the CSV is equal to that of context groups defined. This data format is available only if the row-orientated output option is enabled in the output setting. (Refer to Section 3.3, "Creating a CSV Output Definition" and Section 3.4, "Creating a Group Definition", for details)

[Sample: row-orientated]

```
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,90000000000,30000000000
```

3.2 Creating an Item Master Taxonomy

An item master taxonomy is created from an item master CSV file. The data mapping function provides a tool for creating the item master taxonomy from the item master CSV file.

3.2.1 Item master CSV

Create an item master CSV file in the following format:

Column number	Item	Explanation
1	Item label	<p>This is the label name that corresponds to the item code. The item label is set in the item master taxonomy as the label of the element that corresponds to the item. Specify an empty string if a label is not required.</p> <p>Note that the language that is set in this label is the language used by the system.</p>
2	Item type	<p>Use the local name of the item types (xbrli:~) prescribed in the XBRL 2.1 specifications to specify the item value type. For example, set <code>monetaryItemType</code> for an amount of money. If an empty string is specified, the value is processed as the string type. The item type is set in the item master taxonomy as the type of the element that corresponds to the item.</p>
3	Date type for financial items	<p>Specify either "instant" or "duration" as the date type for the item. This is normally set as follows:</p> <p>[instant]</p> <ul style="list-style-type: none"> - Balance sheet items - Items that express beginning of year and end of year fluctuating statements such as cash flow statement and equity <p>[duration]</p> <ul style="list-style-type: none"> - Statements of income items - Fluctuating statements such as cash flow statement and equity items other than the above <p>If an empty string is specified, the date is processed as the "instant" type. The date type is set in the item master taxonomy as the <code>periodType</code> attribute of the element that corresponds to the item.</p>
4	Debit or credit type for financial items	<p>Specify the "debit/credit" type of the item as either "debit" or "credit".</p> <p>If an empty string is specified, nothing is set. The debit or credit type is set in the item master taxonomy as the <code>balance</code> attribute of the element that corresponds to the item.</p>
5	Item code	<p>Specify the item code name managed in the database. In the item master taxonomy, set this information in the label of the element that corresponds to the item. In the role attribute of the label, <code>http://xml.fujitsu.com/xbrl/mapping/codeLabel</code> is set, and the language used by the system is set as the language.</p> <p>If an empty string is specified, an error occurs.</p>
6	Supplementary item code (can be omitted)	<p>Specify a supplementary item code managed by the database. This can be omitted. In the item master taxonomy, set this information in the label of the element that corresponds to the item.</p>

The following are examples of CSV:

Current Assets,monetaryItemType,instant,debit,101,

Fixed Assets,monetaryItemType,instant,debit,102,
 Current Liabilities,monetaryItemType,instant,credit,201,
 Fixed Liabilities,monetaryItemType,instant,credit,202,

3.2.2 Creating an item master taxonomy

Use the csv2tax.bat command provided by the data mapping function to create an item master taxonomy from the created item master CSV. The csv2tax.bat command is stored in the folder below. The Java execution environment is required to execute this command.

\$(product installation folder)\XWand-tk\tools\datamapping\csv2tax.bat

Note that the default installation folder for this product is "C:\Program Files\ISXWand".

Specify the following arguments when executing this command:

csv2tax.bat (item master CSV file) (XML schema file name of the item master taxonomy) (label link file name of the item master taxonomy)

The following is an example of executing this command:

csv2tax.bat c:\in\master.csv c:\out\master.xsd c:\out\master-label.xml

3.2.3 Customizing an item master taxonomy

The Taxonomy Editor can be used to define a presentation link or similar in order to make it easy to view a created item master taxonomy.

In addition, when wildcard references for items are used in a CSV output definition, orders of elements and presentation links affect orders of items of an output CSV, so please edit the item master taxonomy to obtain intended output. For example, if you want to output items in the proper order such as "code_101", "code_102", and "code_103", you need to define the following presentation links. (Refer to Section 3.3.2, "Wildcard", for details)



3.3 Creating a CSV Output Definition

A CSV output definition is used to define which data is written in a certain column of an output CSV when converting an intermediate instance to the output CSV.

When column-orientated output option is used, one item of an intermediate instance is recorded in one row of a converted CSV. That is, the number of rows of the converted CSV is equal to that of items of the intermediate instance. In addition to item values, other information such as item codes and item context dates can be recorded in one row. Column-orientated output option is a default option.

When row-orientated output option is used, multiple items of an intermediate instance are recorded in one row of a converted CSV. The number of rows of the converted CSV is equal to that of context groups defined. To use the row-orientated output option, the value "SINGLE" must be set as a "RowType" in an output setting. (Refer to Section 3.3.1, "Output setting", for details)

Create a CSV output definition in the following text format:

First column output item name, second column output item name, third column output item name, ...

The following output item names can be specified in CSV output definitions:

Output item name	Explanation
Value	Outputs the item value
Code Code(index number)	Outputs the item code that is defined in the item master taxonomy. The item code is obtained from the label of the item that has the resource role attribute value of "http://xml.fujitsu.com/xbrl/mapping/codeLabel". If there is no item code, an empty string is output.

Output item name	Explanation
	<p>If there is more than one item code, index numbers can be specified as shown below (starting with index 0):</p> <p>First: Code(0), second: Code(1), ...</p>
<p>Label</p> <p>Label(language)</p>	<p>Outputs the item label. The label language ([RFC 3066]) can be specified as follows:</p> <p>Japanese: Label ("ja"), English: Label ("en"), ...</p> <p>If the language is not specified, the standard label found first is output. If the language is specified, the standard label in the specified language that is found first is output.</p>
Name	Outputs a local name of an item.
Namespace	Outputs a namespace URI of an item.
ContextId	Outputs an id attribute value of a context which is referenced from an item.
UnitId	Outputs an id attribute value of a unit which is referenced from an item. For a non-numeric item, an empty string is output.
Instant	If the item context date is "instant", the value at that point in time is output. If the context date is "duration" or "forever", an empty string is output.
StartDate	If the item context date is "duration", the value at the start date is output. If the context date is "instant" or "forever", an empty string is output.
StartDateOrInstant	If the item context date is "instant", the value at that point in time is output, and if the context date is "duration", the value at the start date is output. If the context date is "forever", an empty string is output.
EndDate	If the item context date is "duration", the value at the end date is output. If the context date is "instant" or "forever", an empty string is output.
EndDateOrInstant	If the item context date is "instant", the value at that point in time is output, and if the context date is "duration", the value at the end date is output. If the context date is "forever", an empty string is output.
Measure	For a numeric item, the item unit is output. For a non-numeric item, an empty string is output.
IdentifierScheme	Outputs the scheme attribute value of the identifier element of the context referenced by the item.
IdentifierValue	Outputs the identifier element of the context referenced by the item.
Segment	Outputs, in XML text, the segment of the context referenced by the item. However, the name space declaration is not output.
Scenario	Outputs, in XML text, the scenario of the context referenced by the item. However, the name space declaration is not output.
FileName	Outputs the input instance filename.
<p>Ref(item identifier, output item name)</p> <p>Ref(item identifier, context identifier, output item name)</p>	<p>Specify an item to be referenced using identifiers and output a value of the item using the specified output item name.</p> <p>An item identifier identifies an item defined in an item master taxonomy in the format "{name space name}local name", "local name", or wildcard. (Refer to Section 3.3.2, "Wildcard", for details)</p> <p>A context identifier identifies a context referenced from an item in the format "@context group ID" or wildcard. (Refer to Section 3.3.2, "Wildcard", and Section 3.4, "Creating a Group Definition", for details)</p> <p>If no context identifier is specified, the first item, which appears in an intermediate instance, will be selected.</p> <p>Any string other than an empty string can be specified as an output item name.</p>

Output item name	Explanation
	<p>For example, if Ref("code_101",Value) is specified, the value of the item, whose local name is "code_101", is output.</p> <p>For example, if Ref("code_101","@cg_1",Value) is specified, the value of the item, whose local name is "code_101" and which references the context belonging to the context group "cg_1", is output.</p> <p>If no item is identified by the identifiers, an empty string is output. However, in the case that the output item name is one of the "Code, Label, FileName, Ref, ContextRef, and "string", an available value can be output even if no item is identified.</p> <p>This output item name is mainly used when the row-orientated output option is selected.</p>
ContextRef(context identifier, output item name)	<p>Specify a context to be referenced using identifiers and output a value of the context using the specified output item name.</p> <p>A context identifier identifies a context in the format "@context group ID" or wildcard. (Refer to Section 3.3.2, "Wildcard", and Section 3.4, "Creating a Group Definition", for details)</p> <p>Any string other than an empty string can be specified as an output item name. In addition, an empty string is output if one of the "Value, Code, Label, and Measure" is specified.</p> <p>For example, if ContextRef("@cg_1",StartDate) is specified, the start date of the first context, which belongs to the context group "cg_1", is output.</p> <p>If no context is identified by the identifiers, an empty string is output. However, in the case that the output item name is one of the "FileName, Ref, ContextRef, and "string", an available value can be output even if no context is identified.</p> <p>This output item name is mainly used when the row-orientated output option is selected.</p>
Cat(value 1, value 2)	<p>Outputs the concatenation of value 1 and value 2. If value 1 does not exist, value 2 is output. If value 2 does not exist, value 1 is output. If neither value exists, an empty string is output.</p> <p>For example, if Cat("code_",Code) is specified and the item code is 101, "code_101" is output. If no item code exists, "code_" is output.</p>
Or(value 1, value 2)	<p>If both value 1 and value 2 exist, value 1 is output. If value 1 does not exist, value 2 is output. If value 2 does not exist, value 1 is output. If neither value exists, an empty string is output.</p> <p>For example, if Or(Label("en-US"),Label("en")) is specified, the label whose language is "en-US" is output if it exists, otherwise, the label whose language is "en" is tried to be output.</p>
"string"	Outputs the string embraced with the double-quotation marks.
(empty string)	Outputs an empty string (null string) by specifying an empty string as the output item name.

Note: If an argument is numeric, specify the argument without double-quotation marks like Code(1). If an argument is non-numeric, specify the argument with double-quotation marks like Label("ja").

If a value to be output contains linefeed codes, the value is output after the linefeed codes are removed. For example, even if a label value of an item contains linefeed codes, "Label" outputs the label value which does not contain the linefeed codes.

A linefeed code can be written at the end of a line in a CSV output definition as follows.

output item name (1,1), output item name (1, 2), output item name (1, 3), ...

output item name (2,1), output item name (2, 2), output item name (2, 3), ...

output item name (3,1), output item name (3, 2), output item name (3, 3),

* (row, column)

In this case, a linefeed code is also contained at the end of a line in an output CSV. When column-orientated output option is used, one item of an intermediate instance is written as multiple lines of the output CSV. The number of all lines of the output CSV is equal to "the number of items x the number of lines of a CSV output definition." When row-orientated output option is used, the number of all lines of the output CSV is equal to "the number of lines of a CSV output definition + the number of lines added due to context groups," which does not depend on the number of items of an intermediate instance. (Refer to Section 3.4, "Creating a Group Definition", for details of context groups)

Available output item names are different between column-orientated output and row-orientated output. In each option, the following output item names can be specified.

Data Format	Available Output Item Names
Column-orientated	All output item names explained above.
Row-orientated	"FileName, Ref, ContextRef, Or, "string", (empty string)" can be specified. An empty string is output if other names are specified. Note that Output item names which are specified as arguments of Ref or ContextRef are interpreted based on the specification explained at "Explanation" of the above table.

[Sample: column-orientated]

If you want to output dates (end date or instant) at the first column, units at the second column, item codes at the third column, item values at the fourth column, and English labels at the fifth column, you need to define the CSV output definition as follows.

EndDateOrInstant, Measure, Code, Value, Label("en")

If the number of items of an intermediate instance is four, the following CSV is output.

```
2008-03-31,JPY,101,20000000000,Current Assets
2008-06-30,JPY,101,9000000000,Current Assets
2008-03-31,JPY,102,5000000000,Fixed Assets
2008-06-30,JPY,102,3000000000,Fixed Assets
```

[Sample: row-orientated]

If you want to output the string "Date" at the first column of the first row, item labels at the second and later columns of the first row, dates (end date or instant) at the first column of the second and later rows, and item values at the second and later columns of the second and later rows, you need to define the CSV output definition using linefeed codes as follows. (Refer to Section 3.3.1, "Output setting", for details of the first row)

[RowType: SINGLE]

"Date", Ref("-presentation", Label)

ContextRef("@|", EndDateOrInstant), Ref("-presentation", "@|", Value)

If the number of items of an intermediate instance is four and that of context groups is two, the following CSV is output. (Refer to Section 3.4, "Creating a Group Definition", for details of context groups)

```
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
```

3.3.1 Output setting

In a CSV output definition, output settings can be described before output item names appear in the following format.

[Key: Value]

Available keys and values are as follows.

Key	Value	Explanation
RowType	MULTIPLE	Output item names are applied for each item of an intermediate instance. That is, the column-orientated option is used. (Default option)
	SINGLE	Output item names are applied only once regardless of the number of items of an intermediate instance. That is, the row-orientated option is used.

[Sample: row-orientated]

If you use the row-orientated output option, you need to define the CSV output definition as follows.

[RowType: SINGLE]

"Date", Ref("-presentation", Label)

ContextRef("@|", EndDateOrInstant), Ref("-presentation", "@|", Value)

If the number of items of an intermediate instance is four and that of context groups is two, the following CSV is output. (Refer to Section 3.4, "Creating a Group Definition", for details of context groups)

```
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
```

If "RowType" is not specified or "MULTIPLE" is specified for "RowType" in the above CSV output definition, the following CSV is output.

```
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,9000000000,3000000000
```

Output item names are applied for each item of an intermediate instance (that is, they are applied four times in the above example), so the same values are output redundantly.

For this reason, **if you want to output the row-orientated date using the output item names such as Ref, ContextRef, and so on, you need to specify "SINGLE" for "RowType"**.

On the other hand, **if you want to output the column-orientated data using the output item names such as Value, Label, and so on, you need to omit "RowType" or specify "MULTIPLE" for "RowType"**.

3.3.2 Wildcard

The following wildcard can be specified for the arguments "item identifier" or "context identifier" of the output item names "Ref" and "ContextRef". (Refer to Section 3.4, "Creating a Group Definition", for details of context groups)

Argument	Wildcard	Explanation
item identifier	"-" (hyphen)	All items are recorded in the same row (row-orientated) according to the order of elements defined in an item master taxonomy (document order). Note that abstract elements are omitted. That is, the number of columns to be added to the output CSV is equal to that of non-abstract elements defined.
	"-presentation"	All linked items are recorded in the same row (row-orientated) according to the order of presentation links defined in an item master taxonomy (document order). The target link is the link whose extended link role is "http://www.xbrl.org/2003/role/link" and whose arcrole is "http://www.xbrl.org/2003/arcrole/parent-child". Note that the output order is determined after the rules of prohibiting, overriding, and ordering links are applied. If same items appear in different locations in the presentation link tree, only the first one is output. Abstract elements are omitted. That is, the number of columns to be added to the output CSV is equal to that of valid linked elements. If no links are defined, no columns are added to the output CSV.
	"-presentation(role)"	This is almost same as "-presentation", but you can specify the target extended role of links as the argument "role".
context identifier	"@ "	All context groups are recorded in the same column (column-orientated) according to the order of context groups define (document order).

Argument	Wildcard	Explanation
		That is, the number of rows to be added to the output CSV is equal to that of context groups defined. Note that if no context groups are defined, each context defined in an intermediate instance is regarded as a context group who has the context.

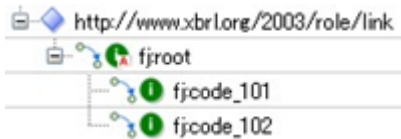
For example, suppose that the CSV output definition is defined using the wildcard "-presentation" and "@|" as follows.

```
[RowType: SINGLE]
```

```
"Date", Ref("-presentation", Label)
```

```
ContextRef("@|", EndDateOrInstant), Ref("-presentation", "@|", Value)
```

Suppose also that presentation links are defined in an item master taxonomy as follows, where the element "root" is an abstract element.



In this case, the wildcard "-presentation" is expanded as the element names "code_101" and "code_102" to row orientation, and two "Ref" columns are added. As a result, the CSV output definition becomes equivalent to the following definition.

```
[RowType: SINGLE]
```

```
"Date", Ref("code_101", Label), Ref("code_102", Label)
```

```
ContextRef("@|", EndDateOrInstant), Ref("code_101", "@|", Value), Ref("code_102", "@|", Value)
```

Suppose also that the context groups "cg_1" and "cg_2" are defined in the group definition, and the context group "cg_1" has the context whose date is the instant "2008-03-31", and the context group "cg_2" has the context whose date is the instant "2008-06-30". (Refer to Section 3.4, "Creating a Group Definition", for details of context groups)

```
cg_1: { the context "c1" whose data is the instant "2008-03-31" }
```

```
cg_2: { the context "c2" whose date is the instant "2008-06-30" }
```

In this case, the wildcard "@|" is expanded as the context groups "cg_1" and "cg_2" to column orientation, and two rows which contain "ContextRef" are added. As a result, the CSV output definition becomes equivalent to the following definition.

```
[RowType: SINGLE]
```

```
"Date", Ref("code_101", Label), Ref("code_102", Label)
```

```
ContextRef("@cg_1", EndDateOrInstant), Ref("code_101", "@cg_1", Value), Ref("code_102", "@cg_1", Value)
```

```
ContextRef("@cg_2", EndDateOrInstant), Ref("code_101", "@cg_2", Value), Ref("code_102", "@cg_2", Value)
```

Suppose also that the following values are specified for an intermediate instance.

```
<code_101 contextRef="c1">20000000000</code_101>
```

```
<code_101 contextRef="c2">90000000000</code_101>
```

```
<code_102 contextRef="c1">50000000000</code_102>
```

```
<code_102 contextRef="c2">30000000000</code_102>
```

As a result of the conversion, the following CSV is output.

```
Date,Current Assets,Fixed Assets
2008-03-31,20000000000,5000000000
2008-06-30,90000000000,30000000000
```

3.4 Creating a Group Definition

A group definition is used to create context groups that are referenced from "Ref" or "ContextRef" in a CSV output definition.

A context group is a group which bundles multiple contexts together. For example, if you want to output the item which refers to the context "cd" of the duration [2008-04-01, 2009-03-31] and the item which refers to the context "ci" of the instant "2009-03-31" into the same row, define one context group which contains these two contexts, and use it to be referred from a CSV output definition.

The following CSV can be output by using a context group.

Value of Item 1 Referring to "cd", Value of Item 2 Referring to "cd", Value of Item 3 Referring to "ci", ...

The following two methods are prepared for creating context groups. You need to choose either one depending on the situation.

-
- Manual grouping
This is a method for creating context groups by specifying groups and its member contexts. The number of context groups is determined in creating a group definition. This method is used if the number of contexts defined in an intermediate instance and details of them are known, or the number of rows of an output CSV and details of them are determined beforehand, or detailed control of the output CSV is required. Refer to Section 3.4.1, "Manual grouping", for details.
- Automatic grouping
This is a method for creating context groups by specifying groups and criteria for determining member contexts. The number of context groups is not determined in creating a group definition, and is determined in converting an instance to CSV. This method is used if the number of contexts defined in an intermediate instance and details of them are not known, or the number of rows of an output CSV and details of them are not determined beforehand, or detailed control of the output CSV is not required. Refer to Section 3.4.2, "Automatic grouping", for details.

A group definition is described in XML, whose format is explained later. Note that the following notations are used in the explanation.

Notation	Meaning
A B C	A, B, or C.
A?	A or nothing.
A*	Zero or more A.
A+	One or more A.
()	A group of notations. Ex.: (A B)* means zero or more (A or B).
<element name>	An XML element.
@attribute name	An XML attribute.
{ }	A group of contexts. Ex.: { c1, c2 } represents the group containing the context "c1" and "c2".
xs:type name	A type definition which is defined in the XML schema specification (W3C). Refer to "XML Schema Part 2: Datatypes Second Edition - http://www.w3.org/TR/xmlschema-2/ " for details.

3.4.1 Manual grouping

This is a method for creating context groups by specifying groups and its member contexts. The number of context groups is determined in creating a group definition. This method is used if the number of contexts defined in an intermediate instance and details of them are known, or the number of rows of an output CSV and details of them are determined beforehand, or detailed control of the output CSV is required.

The following conditions are described to define a group.

Item	Explanation
Group ID	Describes an ID for identifying a group.
Context Reference Condition	Describes conditions for identifying contexts to be referenced. The following conditions can be described in a group definition, and contexts which satisfy the conditions are grouped to the same group. Multiple conditions can be described in one group definition. <ol style="list-style-type: none"> 1. Context ID, Date, identifier, scheme, Segment, and Scenario can be specified in string or regular expression formats. 2. Date can be specified in absolute or relative date formats. 3. Segment or Scenario can be specified in XML format.

The group ids defined here can be used as context identifiers in CSV output definitions. **If you use the manual grouping, specify group ids or wildcard as context identifiers.**

The above definition is described in the following XML format. All elements and attributes other than "XML Fragment" have no namespace name. An error is reported if the XML structure or attribute value does not match the following definition.

Element / Attribute Name	Attribute	Child Elements / Value	Explanation
<groups>	-	<contextGroup>*	A root element
<contextGroup>	@id	<contextRef> <and> <or>	A context group definition
@id	-	xs:ID	A group id
<contextRef>	@name, @type?, @value?, @range?	XML Fragment	A context reference condition. Note that "XML Fragment" can be specified if @name is "Segment" or "Scenario", and @type is "XML". Otherwise, "XML Fragment" is ignored.
@name	-	"ContextId" "StartDate" "EndDate" "Instant" "Forever" "StartDateOrInstant" "EndDateOrInstant" "Identifier" "Scheme" "Segment" "Scenario"	A name of target information
@type	-	"String" "Regex" "Absolute" "Relative" "XML"	A format of a value of target information. Note that "Absolute" and "Relative" can be specified if @name is "StartDate", "EndDate", "Instant", "StartDateOrInstant", or "EndDateOrInstant". Otherwise, "Absolute" and "Relative" are ignored. "XML" can be specified if @name is "Segment" or "Scenario". Otherwise, "XML" is ignored. If @name is "Forever", @type can be omitted.
@value	-	xs:string	A value of target information. If @name is "Segment" or "Scenario", and @type is "XML", @value can be omitted. If @name is "Forever", @value can be omitted.
@range	-	xs:string	An acceptable error range. @range can be specified If @name is "StartDate", "EndDate", "Instant", "StartDateOrInstant", or "EndDateOrInstant". Otherwise, @range is ignored.
<and>	-	(<contextRef> <and> <or>)+	Matches contexts which satisfy all of the conditions defined in child elements.
<or>	-	(<contextRef> <and> <or>)+	Matches contexts which satisfy at least one of the conditions defined in child elements.
XML Fragment	-	-	Child elements of Segment or Scenario.

The following table explains the specifiable values and its meanings for attributes of the context reference condition <contextRef>.

@name	Reference Key	@type	@value	@range
ContextId	id attribute	"String"	id attribute value	-
		"Regex"	regular expression of id attribute value	-
StartDate	start date	"String"	absolute date	-
		"Regex"	regular expression of absolute date	-

		"Absolute"	absolute date	An acceptable error range of date.
		"Relative"	relative date	An acceptable error range of date.
EndDate	end date	"String"	absolute date	-
		"Regex"	regular expression of absolute date	-
		"Absolute"	absolute date	An acceptable error range of date.
		"Relative"	relative date	An acceptable error range of date.
Instant	instant	"String"	absolute date	-
		"Regex"	regular expression of absolute date	-
		"Absolute"	absolute date	An acceptable error range of date.
		"Relative"	relative date	An acceptable error range of date.
Forever	forever	-	-	-
StartDateOrInstant	start date or instant	"String"	absolute date	-
		"Regex"	regular expression of absolute date	-
		"Absolute"	absolute date	An acceptable error range of date.
		"Relative"	relative date	An acceptable error range of date.
EndDateOrInstant	end date or instant	"String"	absolute date	-
		"Regex"	regular expression of absolute date	-
		"Absolute"	absolute date	An acceptable error range of date.
		"Relative"	relative date	An acceptable error range of date.
Identifier	identifier value	"String"	identifier value	-
		"Regex"	regular expression of identifier value	-
Scheme	identifier scheme	"String"	scheme value	-
		"Regex"	regular expression of scheme value	-
Segment	segment	"String"	XML text. Note that a simple text match is used for a QName comparison.	-
		"Regex"	Regular expression of XML text. Note that a simple text match is used for a QName comparison.	-

		"XML"	Omit @value and describe XML fragments as children of <contextRef>.	-
Scenario	scenario	"String"	XML text. Note that a simple text match is used for a QName comparison.	-
		"Regex"	Regular expression of XML text. Note that a simple text match is used for a QName comparison.	-
		"XML"	Omit @value and describe XML fragments as children of <contextRef>.	-

The following table explains the specifiable values and its meanings for @type and @value.

@type	Explanation
String	Describe a string for @value. Full match is performed.
Regex	Describe a Java regular expression for @value. Partial math is performed.
Absolute	Describe an absolute value for @value. @range can be specified.
Relative	Describe a relative value for @value. @range can be specified.
XML	Omit @value and describe XML fragments as children of <contextRef>. XML fragment match (s-equal) is performed.

The following table explains the specifiable values and its meanings for @type and @value when @name of <contextRef> is "StartDate", "EndDate", "Instant", "StartDateOrInstant", or "EndDateOrInstant".

@type	@value
String	Describe a date in the format "xs:date xs:dateTime". Format: xs:date xs:dateTime Ex. : "2010-03-31" -> This matches contexts whose date is "2010-03-31".
Regex	This is almost same as String, but you can use a regular expression.
Absolute	This is almost same as String, but a semantic comparison is performed instead of a simple string comparison. For example, The start date 2009-04-01 and 2009-04-01T00:00:00 are regarded to be same.
Relative	A date is described by a relative value compared to the latest date of end dates or instant dates in the xs:duration format. Note that a positive period has no meaning and is regarded to be a negative value. Format: xs:duration Ex. : "-P1Y" -> This matches contexts whose date is one year before the latest date.

An acceptable error range can be described for @range when @type is "Absolute" or "Relative". Contexts that are in the rage are grouped to the same corresponding group.

@range
An acceptable error range on the basis of @value is described in the xs:duration format. Note that a positive period and a negative period are regarded to be same. Format: xs:duration

@range

Ex.: "P7D"

-> This matches contexts whose date is within seven days from @value.

[Sample: group definition]

- This matches contexts whose id attribute value contains "Consolidated" and start date is "2009-04-01".

```
<groups>
  <contextGroup id="cg_1">
    <and>
      <contextRef name="ContextId" type="Regex" value=".*Consolidated.*"/>
      <contextRef name="StartDate" type="Regex" value="^2009-04-01$"/>
    </and>
  </contextGroup>
</groups>
```

- This matches contexts whose id attribute value contains "Consolidated" or start date is "2009-04-01".

```
<groups>
  <contextGroup id="cg_1">
    <or>
      <contextRef name="ContextId" type="Regex" value=".*Consolidated.*"/>
      <contextRef name="StartDate" type="Regex" value="^2009-04-01$"/>
    </or>
  </contextGroup>
</groups>
```

- This matches contexts whose end date or instant date is within seven days from "2010-03-31".

```
<groups>
  <contextGroup id="cg_2">
    <contextRef name="EndDateOrInstant" type="Absolute" value="2010-03-31" range="P7D"/>
  </contextGroup>
</groups>
```

- This matches contexts whose end date or instant date is within seven days from one year before the latest date.

```
<groups>
  <contextGroup id="cg_3">
    <contextRef name="EndDateOrInstant" type="Relative" value="-P1Y" range="P7D"/>
  </contextGroup>
</groups>
```

- This matches contexts that have the specified segment.

```
<groups>
  <contextGroup id="cg_4">
    <contextRef name="Segment" type="Regex" value=".*m:RedWine.*"/>
  </contextGroup>
</groups>
```

Note: @value is compared as a simple string, so a namespace declaration for the prefix "m" is **not required**.

- This matches contexts that have the specified segment.

```
<groups xmlns:xbrldi="..." xmlns:m="...">
  <contextGroup id="cg_4">
    <contextRef name="Segment" type="XML">
      <xbrldi:explicitMember ...>m:RedWine</xbrldi:explicitMember>
      <xbrldi:explicitMember ...>m:Barbados</xbrldi:explicitMember>
    </contextRef>
  </contextGroup>
</groups>
```


Note: The child elements of <contextRef> are compared as an XML fragment, so a namespace declaration for the prefix "m" is required.

[XML Schema: group definition]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="groups">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="contextGroup"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="contextGroup">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="contextRef"/>
        <xs:element ref="and"/>
        <xs:element ref="or"/>
      </xs:choice>
      <xs:attribute name="id" use="required" type="xs:ID"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="contextRef">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:any/>
      </xs:choice>
      <xs:attribute name="name" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ContextId"/>
            <xs:enumeration value="StartDate"/>
            <xs:enumeration value="EndDate"/>
            <xs:enumeration value="Instant"/>
            <xs:enumeration value="Forever"/>
            <xs:enumeration value="StartDateOrInstant"/>
            <xs:enumeration value="EndDateOrInstant"/>
            <xs:enumeration value="Identifier"/>
            <xs:enumeration value="Scheme"/>
            <xs:enumeration value="Segment"/>
            <xs:enumeration value="Scenario"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="String"/>
            <xs:enumeration value="Regex"/>
            <xs:enumeration value="Absolute"/>
            <xs:enumeration value="Relative"/>
            <xs:enumeration value="XML"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="value" type="xs:string"/>
      <xs:attribute name="range" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="and">
    <xs:complexType>
```

```

    <xs:choice maxOccurs="unbounded">
      <xs:element ref="contextRef" />
      <xs:element ref="and" />
      <xs:element ref="or" />
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="or">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="contextRef" />
      <xs:element ref="and" />
      <xs:element ref="or" />
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>

```

3.4.2 Automatic grouping

This is a method for creating context groups by specifying groups and criteria for determining member contexts. The number of context groups is not determined in creating a group definition, and is determined in converting an instance to CSV. This method is used if the number of contexts defined in an intermediate instance and details of them are not known, or the number of rows of an output CSV and details of them are not determined beforehand, or detailed control of the output CSV is not required.

The following context grouping criteria are described to define groups.

Item	Explanation
Context Grouping Criteria	<p>Describes context grouping criteria. The following criteria can be specified, and contexts that have the same value based on the criteria are grouped to the same group. Multiple criteria can be described in a group definition.</p> <p>1. Grouped by Date, identifier, scheme, Segment, or Scenario</p>

The groups created by the automatic grouping satisfy the following conditions.

- Each context belongs to at least one group. There are no contexts that belong to no groups. Note that the targets of this grouping are contexts in an intermediate instance not in an input instance.
- No groups are sub groups of other groups. For example, the groups { c1, c2, c3 } and { c1, c3 } cannot be created at the same time.
- Each group has at least one context. There are no groups that have no contexts.
- One context may belong to multiple groups that are not sub groups of each other. For example, the groups { c1, c3 } and { c2, c3 } may be created at the same time.

The context groups created by the automatic grouping has the group ID "cg_{n}" which is automatically assigned. Note that {n} is a index of groups, which is an integer started from 1.

Ex.: cg_1, cg_2, ...

The group IDs assigned here can be specified as context identifiers in a CSV output definition. But the number of groups and the order of groups cannot be known preliminarily, so it is not recommended to use group IDs in a CSV output definition. **Use wildcard as a context identifier, if you use the automatic grouping.**

The above definition is described in the following XML format. All elements and attributes other than "XML Fragment" have no namespace name. An error is reported if the XML structure or attribute value does not match the following definition.

Element / Attribute Name	Attribute	Child Elements / Value	Explanation
<groups>	-	<contextCls>?	A root element
<contextCls>	-	<contextId> <and>	A context group definition

Element / Attribute Name	Attribute	Child Elements / Value	Explanation
<contextId>	@name, @range?	-	A context grouping criterion
@name	-	"ContextId" "Period" "StartDateOrInstant" "EndDateOrInstant" "Identifier" "Scheme" "Segment" "Scenario"	A criterion name
@range	-	xs:string	An acceptable error range. @range can be specified if @name is "Period", "StartDateOrInstant", or "EndDateOrInstant". Otherwise, @range is ignored.
<and>	-	<contextId>+	Contexts which satisfy all of the conditions defined in child elements are grouped to the same group. If the <and> element has multiple <contextId> elements, "Period", "StartDateOrInstant", and "EndDateOrInstant" cannot be specified for @name at the same time.

The following table explains the specifiable values and its meanings for attributes of the context grouping criterion <contextId>.

@name	Criterion	@range
ContextId	Grouped by id attribute value. Note that there are no contexts that have the same id attribute value with other contexts, so groups are created for each context.	-
Period	Grouped by date. Contexts that have a same period or same instant are grouped to the same group. A context which has a period and a context which has an instant are not grouped to the same group.	An acceptable error range of date.
StartDateOrInstant	Grouped by date. In addition to the Period condition, contexts whose start date and instant are same are grouped to the same group. A context which has a period and a context which has an instant may be grouped to the same group.	An acceptable error range of date.
EndDateOrInstant	Grouped by date. In addition to the Period condition, contexts whose end date and instant are same are grouped to the same group. A context which has a period and a context which has an instant may be grouped to the same group.	An acceptable error range of date.
Identifier	Grouped by identifier value	-
Scheme	Grouped by identifier scheme	-
Segment	Grouped by segment	-

Scenario	Grouped by scenario	-
----------	---------------------	---

An acceptable error range can be described for @range when @name of <contextId> is "Period", "StartDateOrInstant", or "EndDateOrInstant". Contexts that are in the range are grouped to the same group.

@range
<p>An acceptable error range on the basis of @value is described in the xs:duration format. The range specified here means acceptable maximum difference of dates of contexts in the same group. Note that a positive period and a negative period are regarded to be same.</p> <p>If the automatic grouping cannot be deterministic, an error is reported. For example, if a group has the contexts "c1" and "c2", and the context "c3" is not yet grouped, and "c1" and "c3" are in the acceptable error range, however "c2" and "c3" are not in the acceptable error range, then it is not deterministic that "c1" and "c3" can be grouped to the same group or not.</p> <p>Format: xs:duration</p> <p>Ex. : "P7D"</p> <p>-> Contexts whose dates are within seven days are grouped to the same group.</p>

[Sample: group definition]

- Contexts whose periods, instants, or end date and instant are within seven days are grouped to the same group.

```
<groups>
  <contextCls>
    <contextId name="EndDateOrInstant" range="P7D" />
  </contextCls>
</groups>
```

- Contexts whose identifiers, schemes, segments, and scenarios are same and periods, instants, or end date and instant are same are grouped to the same group.

```
<groups>
  <contextCls>
    <and>
      <contextId name="EndDateOrInstant" />
      <contextId name="Identifier" />
      <contextId name="Scheme" />
      <contextId name="Segment" />
      <contextId name="Scenario" />
    </and>
  </contextCls>
</groups>
```

[XML Schema: group definition]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="groups">
    <xs:complexType>
      <xs:choice minOccurs="0">
        <xs:element ref="contextCls" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="contextCls">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="contextId" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:element ref="and" />
    </xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="contextId">
    <xs:complexType>
        <xs:attribute name="name" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="ContextId" />
                    <xs:enumeration value="Period" />
                    <xs:enumeration value="StartDateOrInstant" />
                    <xs:enumeration value="EndDateOrInstant" />
                    <xs:enumeration value="Identifier" />
                    <xs:enumeration value="Scheme" />
                    <xs:enumeration value="Segment" />
                    <xs:enumeration value="Scenario" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="range" type="xs:string" />
    </xs:complexType>
</xs:element>
<xs:element name="and">
    <xs:complexType>
        <xs:choice maxOccurs="unbounded">
            <xs:element ref="contextId" />
        </xs:choice>
    </xs:complexType>
</xs:element>
</xs:schema>

```

3.5 Creating a Data Mapping Definition

This section describes the data mapping definitions from the taxonomy that is in accordance with the instance to be processed, to the item master taxonomy.

These definitions handle the processing target instance as input, and the intermediate instance as output, to make Formula definitions. For an overview of definitions for this purpose, refer to Section 2.1, "Overview of Definitions". For the procedure for creating definitions, refer to Section 2.2, "Procedure for Creating Definitions". For the calculation expression definitions when more than one element is the input, refer to Section 2.3, "Calculation Expressions".

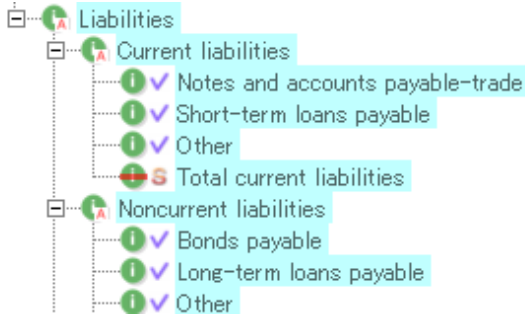
3.5.1 Setting items that need not be mapped

The data mapping function presumes that one or more data mapping definitions have been defined for each of the items that appear in the input instance. If there is an item that does not have a data mapping definition, a warning is posted to the application during data mapping processing.

Define in advance that data mapping is not required for items such as those that do not need to be processed by an existing system. This means that, if an anomaly appears in an input instance (for example, if an extraordinary loss allowance appears), the fact that there is no "data mapping definition" nor a "definition indicating that mapping is not required" enables applications to detect the presence of the anomaly in order to post a warning.

In order to define that mapping is not required for an input taxonomy item, select the item in the input taxonomy pane and right click to open the context menu. In the context menu, select "Set the mapping not required mark" to define that the item does not require mapping. The following example window shows the results of defining that mapping is not required for "Total current liabilities".

Figure 3.4 Example of window when the mark indicating that mapping is not required has been set



3.5.2 Setting mandatory output items

The data mapping function can define that output of a value in the intermediate instance is mandatory. When a data mapping definition exists, but data is not output because an expected item does not exist in the input instance, this definition enables this to be detected.

In order to define an output taxonomy item as a mandatory output item, select the item in the output taxonomy pane and right click to open the context menu. In the context menu, select "Set the output mandatory mark" to define that output of a value is mandatory for this item. The following example window shows the results of defining that output is mandatory for "Current assets".

Figure 3.5 Example of window when the mark indicating mandatory output has been set



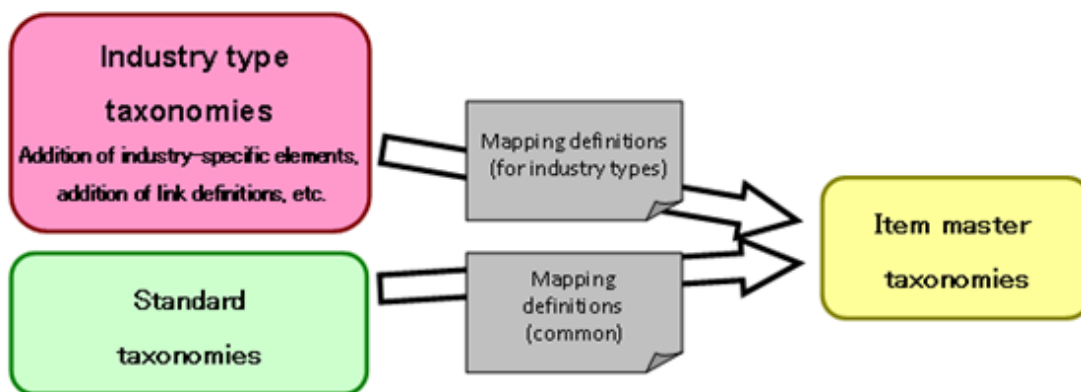
3.5.3 Data mapping definition stratification

A standard taxonomy can be extended to define an industry type taxonomy, an industry type taxonomy can be extended to define a corporate taxonomy, and so on. By additionally defining only the differences, a taxonomy can be extended to define another taxonomy.

The data mapping function can stratify the data mapping definitions to suit the hierarchical structure of taxonomies and define data mapping definitions of only the differences.

The following figure is a representative diagram.

Figure 3.6 Data mapping definition stratification



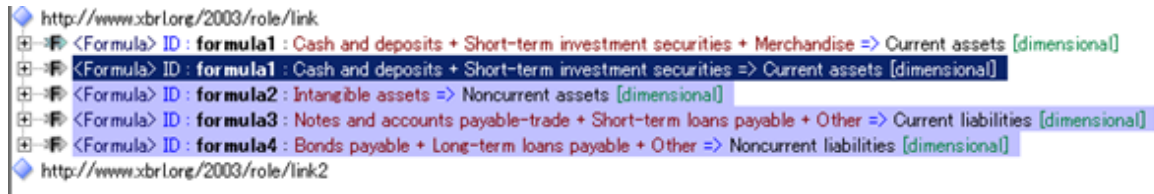
Use the following procedures to create definitions to achieve the above:

- Creating data mapping definitions for a standard taxonomy
Create data mapping definitions between the standard taxonomy and the item master taxonomy. Use these as the "Mapping definitions (common)".

- Creating data mapping definitions for an industry type taxonomy
Create data mapping definitions between the industry type taxonomy and the item master taxonomy. To create these definitions, select "Import taxonomy including formula definitions" from the File menu, then specify "Mapping definition (common)". This results in the expressions defined by "Mapping definition (common)" also being displayed in the formula pane. The expressions for data mapping definitions that have been read in are displayed in a different color and cannot be edited.

The following figure shows an example of this window.

Figure 3.7 Results of reading in stratified data mapping definitions

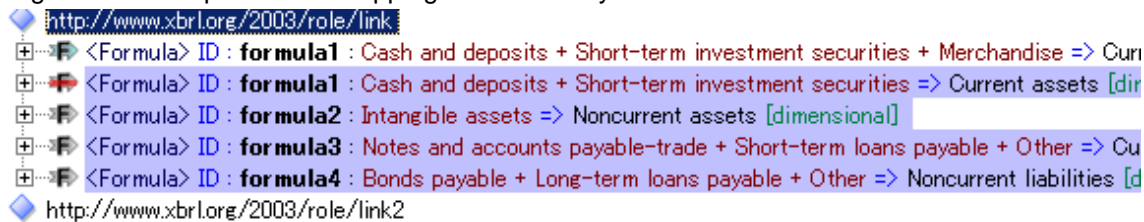


With stratified data mapping definitions, in some cases the extension source data mapping definitions are overwritten by the extended side. In order to overwrite a definition, use the following procedures to create the definition:

- Prohibiting an extension source data mapping definition
Select the expression that you want to prohibit and right click to open the context menu. In the context menu, select "Set execution prohibited mark" to ensure that the expression is not executed.
- Creating a data mapping definition to replace a prohibited data mapping definition
Create a new definition using the normal procedure.

The following window example shows that execution prohibited has been set for the extension source data mapping definition (the definition with the formula1 ID that is shown with a blue background) and a new formula1 definition has been created.

Figure 3.8 Example of data mapping overwritten by the extended side

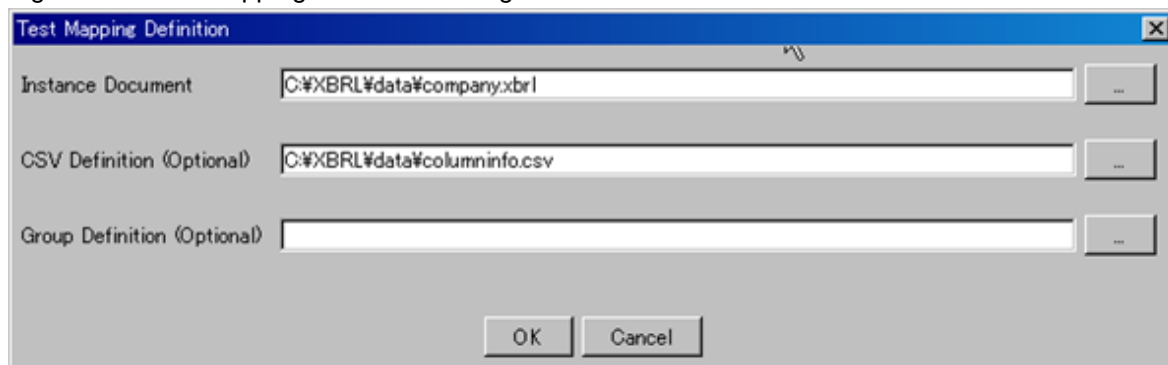


3.6 Confirming Operation

A created definition can be applied to an instance to check its operation.

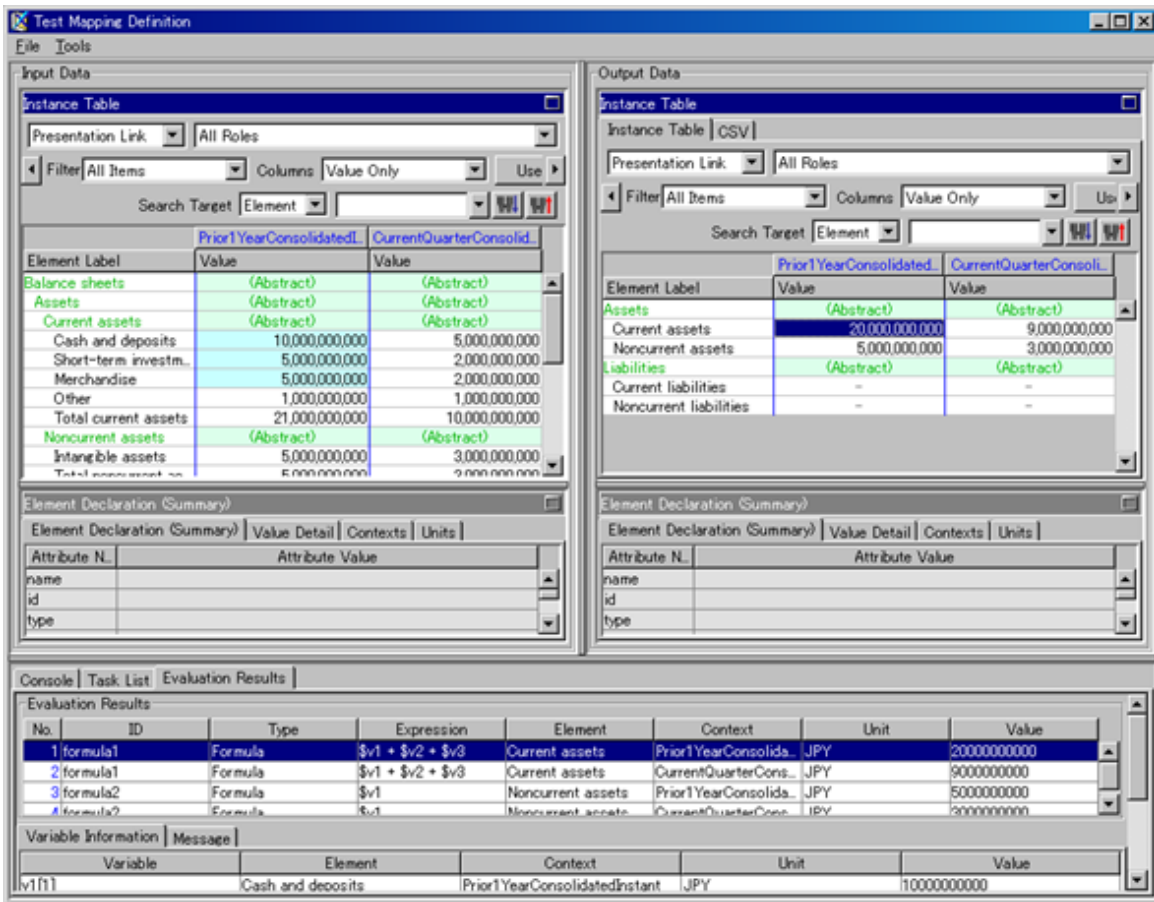
When "Test mapping definition" is selected from the Tools menu, the "Test Mapping Definition" dialog is opened. In this dialog, specify the apply target instance, the CSV output definition, and the group definition.

Figure 3.9 "Test Mapping Definition" dialog



When "OK" is clicked, the definition currently being edited is applied to the specified instance and the results are displayed.

Figure 3.10 Application results display



In the application results window, the input instance is displayed on the left, and the application results are displayed on the right. If a CSV output definition was specified, the definition is executed up to the CSV conversion. The results of the CSV conversion can be checked using the "CSV" tab.

The "Evaluation Results" tab in the lower part of the window displays the expression evaluated by the apply process and the variable values assigned to each expression at the time of evaluation. The user can refer to this information to check the expression that was operating and check whether or not the variables assigned at that time were as expected.

The menus and submenus of this window is as follows:

Table 3.1 Menus and submenus

Menu	Submenu	Description
File	Save Console Messages	"All": Saves to a file all the messages that are output in the console of "Information" pane. "Error Messages": Saves to a file the error messages that are output in the console of "Information" pane.
	Save Output Instance	Saves the instance displayed in "Output Data".
Tools	Update	Reloads the created definitions. It is used when you made any updates to the definitions after the testing.
	Evaluate Formulae	Reloads the instance document and applies only calculation formula definitions to the instance. It is used when you made any updates to the instance.
	Evaluate Assertion	Reloads the instance document and applies only assertion formula definitions to the instance. It is used when you made any updates to the instance.

Chapter 4 Other Functions

When "Options" is selected under the Tools menu, the "Options" dialog opens. This dialog can be used to customize the tool operation. The contents set under each tab are the same as for the Taxonomy Editor and the Instance Creator. Refer to either of the following manuals for details of each option:

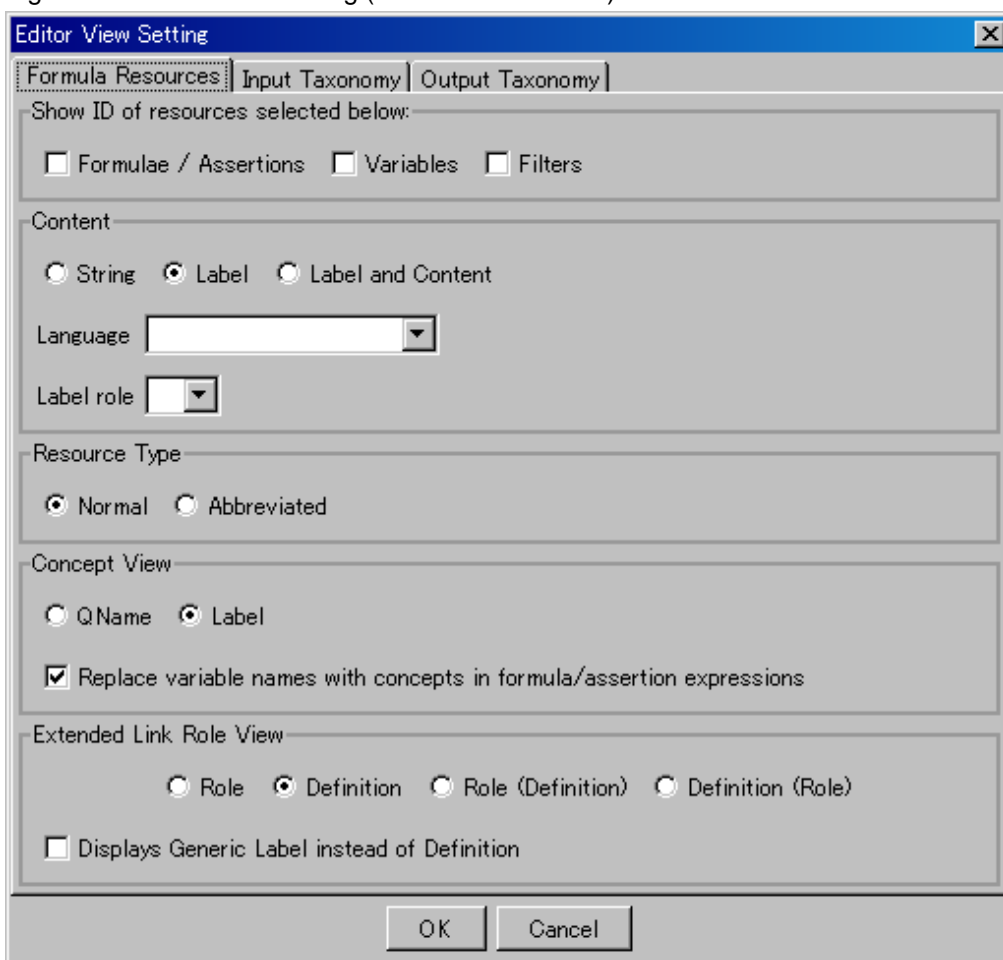
- Taxonomy Editor User's Guide
- Instance Creator User's Guide

4.1 Setting the Editor View

The display format used to display a definition can be customized.

From the View menu on the menu bar, select "Editor View Setting". Enter the settings in the "Editor View Setting" dialog.

Figure 4.1 Editor View Setting (Formula Resources)

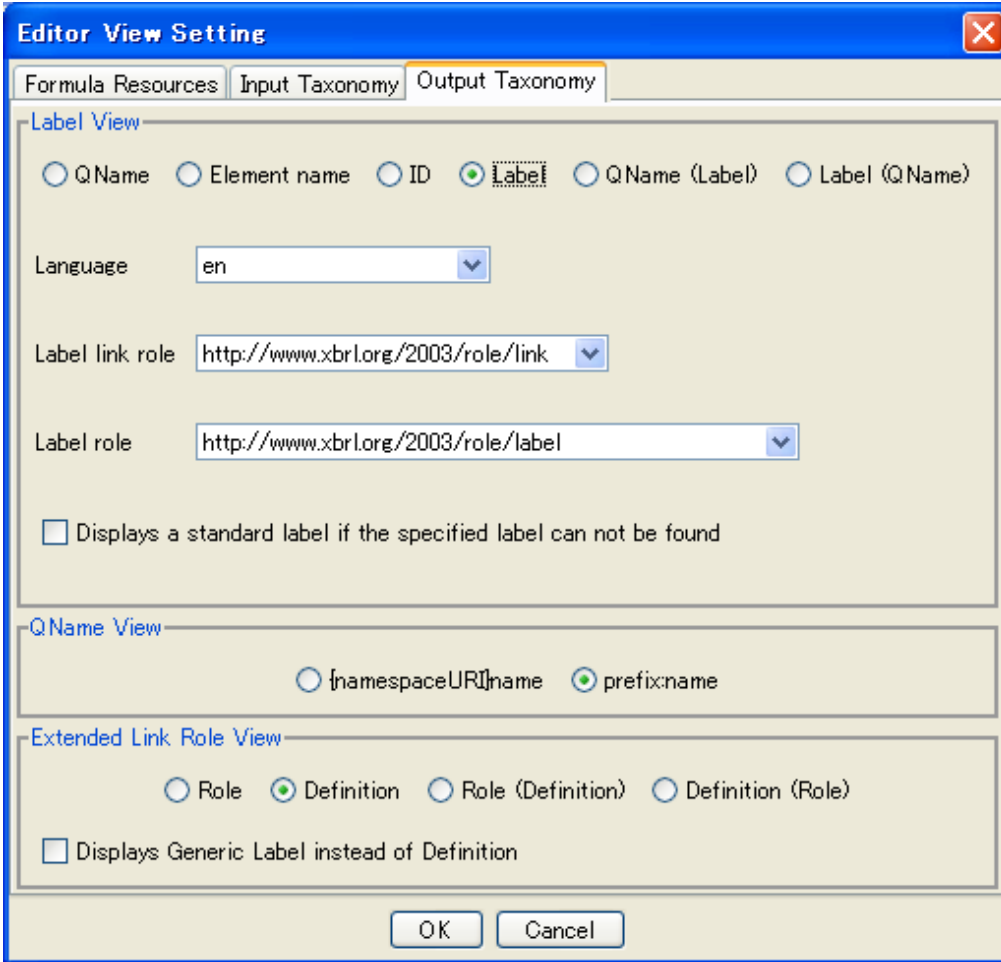


The format of the "Formula" pane can be customized.

	Item	Explanation
Show ID of resources selected below	Formulae/Assertions	Specify whether or not to display the ID attribute value when displaying calculation expressions and verification expressions.

	Variables	Specify whether or not to display the ID attribute value when displaying variables.
	Filters	Specify whether or not to display the ID attribute value when displaying filters.
Content	(Preference)	Specify whether to display a "String", "Label", or "Label and Content" when displaying contents.
	Language	Specify the language used when a label is displayed.
	Label role	Specify the role when a label is displayed.
Resource Type		Specify the type of display format for displays.
Concept View		Specify the display format for concepts (elements).
	Replace variable names with concepts in formula/assertion expressions	Specify whether or not to replace variables in expression displays with concepts.
Extended Link Role View	Role	Displays the role.
	Definition	Displays the content of the definition attribute displayed as the definition element value in the schema roleType element. If a definition element is not defined, the role is displayed.
	Role (Definition)	Displays the role and the definition.
	Definition (Role)	Displays the definition and the role.
	Displays Generic Label instead of Definition	When generic labels are defined for the roleType element, the application displays the generic label with the selected language in the label view.

Figure 4.2 Editor View Setting (Input Taxonomy and Output Taxonomy)



		Item	Explanation
Label View		QName	Uses the qualifier to display an element.
		Element name	Uses the element name to display an element.
		ID	Uses the ID to display an element.
		Label	Uses the label name to display an element (default).
		QName (Label)	Uses the qualifier and label to display an element.
		Label (QName)	Uses the label and qualifier to display an element.
		Language	The language used to display label names.
		Label link role	The label link role being used.
		Label role	The label role being used.
		Displays a standard label if the specified label cannot be found.	Specifies the operation when the label specified in the role is not found.
QName View		{namespaceURI}name	namespaceURI + name used in displays.
		prefix:name	Prefix + name used in displays (default).
Extended Link Role View		Role	Displays the role.

Definition	Displays the content of the definition attribute displayed as the definition element value in the schema roleType element. If a definition element is not defined, the role is displayed.
Role (Definition)	Displays the role and the definition.
Definition (Role)	Displays the definition and the role.
Displays Generic Label instead of Definition	When generic labels are defined for the roleType element, the application displays the generic label with the selected language in the label view.

4.2 Searching

The searches shown in the following table can be performed from the "Formula" pane.

Method	Explanation
Keyword Search	The search is performed using the specified keyword. Specify the keyword in the input field next to the list box.
Concept Search	A concept search searches for definitions associated with the specified concept. Specify the concept in either the "Input Taxonomy" or "Output Taxonomy" pane, then right click and select "Find Resources Associated With This Concept".
Resource Search	A resource search searches for definitions associated with the specified resource. Specify the resource in the "Formula" pane, then right click and select "Find Resource".

Figure 4.3 Filter button (formula pane)



In addition, the filter button can be used to display in the various "Formula" pane windows only those targets that match the search conditions.

The searches shown in the following table can be performed from the "Input Taxonomy" pane and the "Output Taxonomy" pane.

Method	Explanation
Keyword Search	The search is performed using the specified keyword. Specify the keyword in the input field in the upper part of the window.
Concept Search	In the "Formula" pane, select "Concept Name Filter", then right click and select "Find Concept". This searches for the selected item in the "Input Taxonomy" pane. In the "Formula" pane, select the "Output Setting" of the calculation expression, then "Aspects", then "Concept". Right click and the select "Find Concept". This searches for the selected item in the "Output Taxonomy" pane.