

Metadata Manual

SIMPROCESS

Release 1



CACI

EVER VIGILANT

Copyright © 2007-2012 CACI, INC.-FEDERAL.

All rights reserved. No part of this publication may be reproduced by any means without written permission from CACI.

The information in document is believed to be accurate in all respects. However, CACI cannot assume the responsibility for any consequences resulting from the use thereof. The information contained herein is subject to change. Revisions to this publication or new editions of it may be issued to incorporate such change.

SIMPROCESS is a registered trademark of CACI, INC.-FEDERAL.

Organization of the SIMPROCESS Documentation Set 5

CHAPTER 1

SIMPROCESS Metadata 7

What is Metadata? 7

What are Metamodels? 7

Metamodel - Metadata Relationship 8

CHAPTER 2

SIMPROCESS Metamodel Editor 9

Launching the Metamodel Editor 9

Metamodel Editor Description 10

Metamodel Editor Operation 12

Creating a Metamodel 17

CHAPTER 3

Assigning Metamodels 30

Assign Metamodel... Menu Item 30

Assign Metamodel Operation 31

Reconciling Metamodels 34

Updating Metamodels 37

CHAPTER 4

Entering Metadata 39

Prerequisites 39

Metadata Entry Point 39

Metadata Dialog 41

Metadata Search 57

CHAPTER 5

***BPEL Metadata* 60**

Overview 60

BPEL Process Metamodel..... 64

BPEL Activity Metadata 67

BPEL Export 75

Organization of the SIMPROCESS Documentation Set

The SIMPROCESS documentation set consists of four manuals:

- *Getting Started With SIMPROCESS*
- *SIMPROCESS User's Manual*
- *SIMPROCESS Metadata Manual*
- *SIMPROCESS OrgModel Manual*

Getting Started

The *Getting Started With SIMPROCESS* manual is a must for first time SIMPROCESS users. This manual can also be used for evaluation purposes. Chapter 1 provides an overview of Process Modeling and Analysis and the SIMPROCESS product. Chapter 2 provides system requirements and installation instructions. Chapters 3 and 4 of the *Getting Started With SIMPROCESS* manual provide a tutorial, and Chapter 5 provides a description of the demonstration and reference models.

User's Manual

The *User's Manual* can be opened directly from the **Help/SIMPROCESS Manuals** menu. It is divided into three parts. Part A is an excellent reference for beginners and casual users. This

part contains detailed documentation of the basic and intermediate functions of SIMPROCESS. Chapters 1 and 2 provide SIMPROCESS terminology and basics. Chapter 3 provides a detailed description of SIMPROCESS Statistical Constructs and their use. Chapter 4 describes in detail how the SIMPROCESS Activity Modeling blocks are used. Chapter 5 describes the use of Resources. Graphical Modeling Constructs are covered in Chapter 6. Chapter 7 is dedicated to Activity-based Costing, and Chapter 8 covers the Output Reports for analysis.

Part B is a reference intended for advanced users of SIMPROCESS. This part contains detailed documentation of the programming and library management functions in SIMPROCESS Professional Edition. Chapter 9 documents Reusable Templates and Library Management. Chapter 10 covers the Advanced SIMPROCESS constructs such as attributes, expressions, and time stamps. Chapter 11 wraps-up the advanced features of SIMPROCESS with descriptions of the complex features of the Generate activity and Downtime Schedules for Resources.

Part C describes the integrated tools included with SIMPROCESS Professional. Chapter 12 of this manual provides an introduction to data analysis and ModelFit. Chapter 13 covers using the SIMPROCESS Database, while Chapter 14 discusses using the Experiment Manager. How to do optimization using OptQuest is in Chapter 15, and Chapter 16 discusses SIMPROCESS Dashboards. Chapter 17 explains how to create and extract model bundles. Chapter 18 discusses how to create and view custom reports, and Chapter 19 describes the use of Scenarios and Scenario Reports.

Metadata Manual

The Metadata Manual describes how to build and edit SIMPROCESS metamodels, assign metamodels to a SIMPROCESS model, and enter metadata in a SIMPROCESS model.

OrgModel Manual

The OrgModel Manual describes how to build and edit SIMPROCESS Organization and Resource Models (OrgModels) and assign OrgModels to a SIMPROCESS model.

CHAPTER 1

SIMPROCESS Metadata

This chapter presents an overview of metadata and SIMPROCESS metamodels.

What is Metadata?

Metadata is data about data. From a SIMPROCESS perspective, metadata is a collection of customer specific unique data attributes for SIMPROCESS model components (Activity, Entity Type, Resource, Pad, Connector, Model, and Swimlane). (See the *SIMPROCESS User's Manual* for more information on SIMPROCESS models.)

What are Metamodels?

Metamodels are collections of metadata definitions. The key functional value that metamodels offer is the capability to define unique sets of data definitions associated with SIMPROCESS model components. For example, a metamodel could be created specifying data routinely used in an organization. The metadata within the metamodel could relate to SIMPROCESS Activities that represent activities within the organization. These data definitions could then be used in SIMPROCESS Activities to capture values for those data items. Another example is the definition of industry-standard data such as DOD's CADM for DODAF views. Likewise, a vertical industry may also have specific data that is pertinent to a process model such as Sarbanes-Oxley financial requirements or HIPAA health care requirements.

Metamodels are independent of any given SIMPROCESS model so that they can be associated with multiple SIMPROCESS models. Likewise, multiple metamodels can be used within a single SIMPROCESS model. It is a many-to-many relationship.

Metamodel - Metadata Relationship

A SIMPROCESS metamodel consists of data item definitions. Metamodels are created using the SIMPROCESS Metamodel Editor ([Chapter 2](#)). The data item definitions in metamodels are used within a SIMPROCESS model to create metadata for a SIMPROCESS model component (Activity, Entity Type, Resource, Pad, Connector, Model, or Swimlane). One or more metamodels can be assigned to a SIMPROCESS model ([Chapter 3](#)). This means that the data item definitions in the assigned metamodels are applied to the appropriate SIMPROCESS model components. When the properties of a SIMPROCESS model component (for which data item definitions exist) are edited, metadata based on the data item definitions for that component can be entered ([Chapter 4](#)).

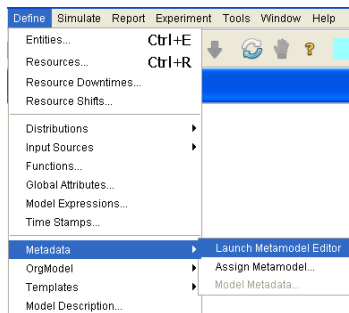
CHAPTER 2

SIMPROCESS Metamodel Editor

This chapter describes the Metamodel Editor and its operation.

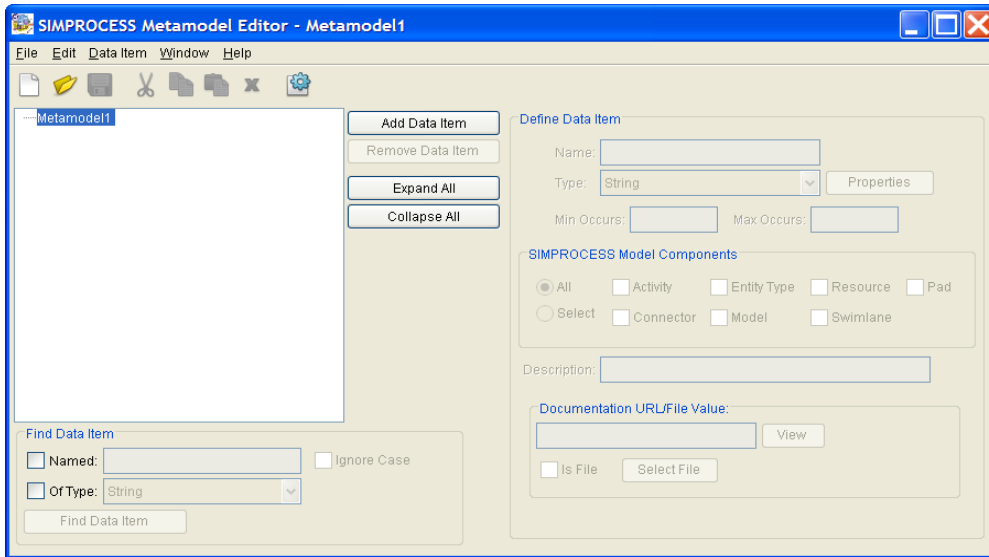
Launching the Metamodel Editor

The SIMPROCESS Metamodel Editor is a separate application installed with SIMPROCESS. It can be launched from within SIMPROCESS by choosing **Launch Metamodel Editor** from the **Define/Metadata** menu.



When selected, the Metamodel Editor opens to an empty metamodel with a default name. If the Metamodel Editor is already open when **Launch Metamodel Editor** is selected, a new

editor window will open with a default name.

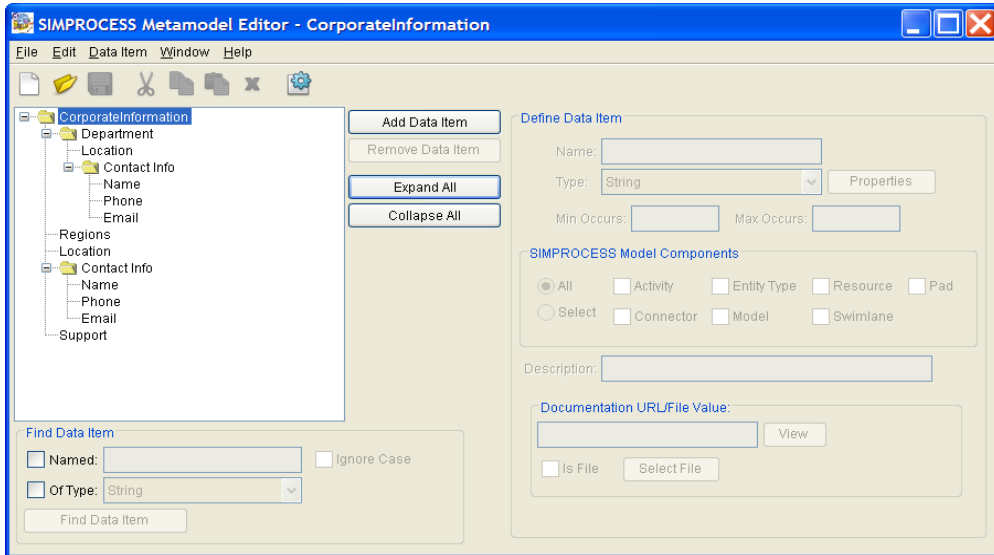


Metamodel Editor Description

The SIMPROCESS Metamodel Editor has two primary sections: the data item tree on the left and the **Define Data Item** section on the right. The buttons in between operate on the data item tree. (See [“Metamodel Editor Buttons” on page 16.](#)) An additional area for finding specific data items is below the data item tree.

Data Item Tree

The data item tree contains a hierarchical view of the data item definitions created for the metamodel. The top node of the tree is the name of the metamodel. All data item definitions are subordinate to the metamodel name.



Define Data Item

The **Define Data Item** section is used to specify the data item definition of the data item selected in the data item tree. A data item definition consists of

- **Name**
- **Type** - String, Real, Integer, Boolean, URL, Single Selection, Multiple Selection, Include, or Group
- **Min Occurs** - Minimum number of occurrences. Must be an Integer greater than or equal to zero. Zero indicates that no instances of this data item are required.
- **Max Occurs** - Maximum number of occurrences. Must be an Integer greater than or equal to zero. Zero indicates that the number of instances of this data item is unbounded.
- **SIMPROCESS Model Components** - Activity, Entity Type, Resource, Pad, Connector, Model, or Swimlane
- **Description**
- **Documentation URL/File Value**

Properties can be set on every **Type** except Group. See [“Creating a Metamodel,” beginning on page 17](#) for more information on these fields.

Find Data Item

Below the data item tree is the **Find Data Item** section. The **Find Data Item** section allows searching for a particular data item by name and type. If **Named** is selected, a data item name to search for can be entered and **Ignore Case** is enabled for selection. Wild cards (*) can be used before and/or after the search text to search for similarly named data items. If **Of Type** is selected, *String*, *Real*, *Integer*, *Boolean*, *URL*, *Single Selection*, *Multiple Selection*, *Include*, or *Group* can be selected. The search will then look for data items with a type matching the selected type. When both **Named** and **Of Type** are selected, the search is a logical AND search. That is, the search will look for data items that match the entered text and the selected type.

Metamodel Editor Operation

Metamodel Editor Menus

File Menu



New opens a new metamodel. Note that each opened metamodel, whether new or previously saved, will open in a new Metamodel Editor window.

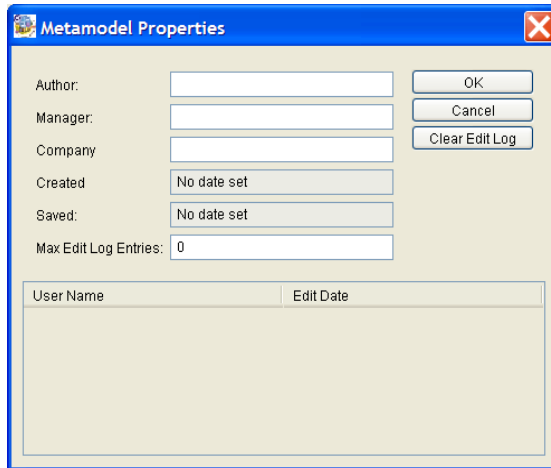
Open opens a previously saved metamodel. Metamodel files have the extension `.mdm` and must be located in the `metamodel` directory located in the SIMPROCESS installation directory. The **Open** menu item only displays metamodels located in that directory and only displays metamodels not already open. If a directory with the same name as the opened metamodel does not exist in the `metamodel` directory, it is created during the open process.

Open Recent contains a list of up to 5 recently opened metamodels for easy access. The last item on the menu is **Clear Menu**, which removes all recently opened metamodels from the list.

Save saves the metamodel. Note that the first time **Save** is used on a new metamodel, **Save As** will be invoked so the default name can be changed to something more meaningful. On the first save, a directory with the same name as the metamodel will be created. This directory is for optional data source or documentation files.

Save As saves the metamodel with a different name and only saves to the `metamodel` directory located in the SIMPROCESS installation directory or subdirectories of the `metamodel` directory. The `.mdm` extension will be appended to the file name if omitted. A directory with the new metamodel name is created in the directory where the metamodel is saved. Contents from the previously named metamodel's directory are copied to the new directory. Using **Save As** is the only way to change the name of a metamodel. The **Save As** dialog allows the creation or deletion of new directories within the `metamodel` directory or within a subdirectory of the `metamodel` directory. If no subdirectory or already saved metamodel is selected, the metamodel will be saved in the `metamodel` directory. If a subdirectory is selected or a metamodel within the subdirectory is selected, the metamodel will be saved in that directory. The **Add Directory** button adds a directory in the same manner. If nothing is selected, the new directory will be added to the `metamodel` directory, otherwise the new directory will be added to the directory that is selected or to the directory of the selected metamodel. The **Remove Directory** button is only enabled when a subdirectory of the `metamodel` directory is selected. Removing a directory will also, after confirmation, delete any directory contents.

Properties opens a dialog that contains four editable fields: **Author**, **Manager**, **Company** and **Max Edit Log Entries**. **Max Edit Log Entries** sets the maximum number of entries that the Edit Log will keep. The value defaults to zero. (Zero indicates that the Edit Log can contain an unlimited number of entries.) The Edit Log contains a list of pairs of **User Name** and **Edit Date**. An Edit Log entry is added every time the metamodel is saved. If **Max Edit Log Entries** contains a number larger than zero, then only the latest entries are kept. The **Created** and **Saved** fields cannot be edited. **Created** contains the date and time that the metamodel was first saved. **Saved** contains the date and time of the latest save. The **Clear Edit Log** button removes all entries from the Edit Log.

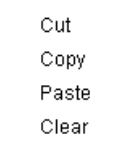


Create HTML Report outputs the metamodel to HTML format to be made accessible via a standard web browser. After selecting a destination directory, a new directory will be created with the same name as the metamodel with any spaces or slashes replaced by underscores and all upper case characters being converted to lower case characters. A tree of the metamodel's data items is provided with links to the properties of each data item. HTML Reports will automatically be created for other metamodels referred to by Include data items, and their properties will link to those reports.

Close closes the metamodel. If the metamodel has been changed since the last save, there will be a prompt to save the metamodel.

Exit close all open metamodels. Each metamodel will prompt for save if necessary. Selecting **Cancel** on any save prompt will stop the exit process.

Edit Menu



Cut removes the selected data items and places them on the clipboard.

Copy copies the selected data items to the clipboard.

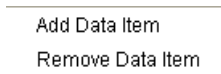
Paste pastes the data item definitions on the clipboard as children of the selected data item defini-

tion. Note that the selected data item definition must either be the top level node (name of the metamodel), a `Group` data item definition, or a data item that has been designated to allow children. (See “[Defining a Data Item](#),” beginning on page 17 for information on data item definitions that allow child data item definitions.)

Clear removes the selected data items and does not place them on the clipboard.

Note that **Cut**, **Copy**, **Paste**, and **Clear** (and their associated tool bar buttons) only apply to data item definitions. To **Cut**, **Copy**, **Paste**, or **Clear** text fields (such as **Data Item Name**), the appropriate keyboard commands must be used. (For example, on Windows systems, Control-C is copy.) Also, multiple data item definitions can be selected, but they must be contiguous. Cutting or copying a data item that has child data items will include all child data items.

Data Item Menu



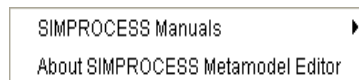
Add Data Item adds a new data item to the selected metamodel, `Group` data item, or data item designated as a parent data item. **Add Data Item** and the **Add Data Item** button are only enabled when the selected data item is the metamodel name or can have child data items.

Remove Data Item removes the selected data items. This and the **Remove Data Item** button function the same as the **Edit/Clear** menu item.

Window Menu

This menu contains a list of the open metamodels. Selecting a metamodel in the list causes that Metamodel Editor to come to the front.

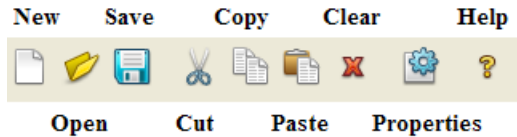
Help Menu



SIMPROCESS Manuals opens the *SIMPROCESS Metamodel Editor User’s Manual* or the *SIMPROCESS User’s Manual*. They will open using the application designated for PDF files by the platform on which SIMPROCESS is installed. On Linux systems, there must be a `view.properties` file as described for the **View** button on the **Documentation** tab of Activity and Process properties dialogs. (See the **Common Activity Input Fields** section of Chapter 2 of the *SIMPROCESS User’s Manual*.)

About **SIMPROCESS Metamodel Editor** displays information about the Metamodel Editor, including the version, the copyright, and technical support information.

Metamodel Editor Toolbar



The toolbar buttons have the same action as the menu item of the same name.

- **New - File/New**
- **Open - File/Open**
- **Save - File/Save**
- **Cut - Edit/Cut**
- **Copy - Edit/Copy**
- **Paste - Edit/Paste**
- **Clear - Edit/Clear**
- **Properties - File/Properties**
- **Help - Help/Metamodel Help**

Metamodel Editor Buttons

Add Data Item adds a new data item as a child of the selected item. This button is only enabled when only one item is selected and the selected item is the metamodel name, a `Group` data item, or a data item that has been designated to allow child data items.

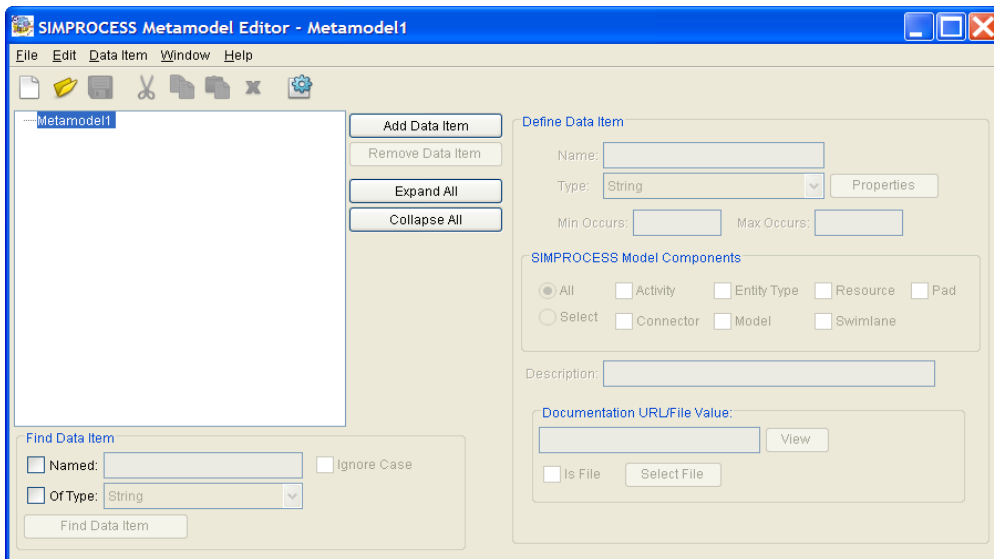
Remove Data Item removes the selected data items. This button and the **Remove Data Item** menu item function the same as the **Edit/Clear** menu item.

Expand All expands the complete data item hierarchy.

Collapse All restores the data item hierarchy to the top level.

Creating a Metamodel

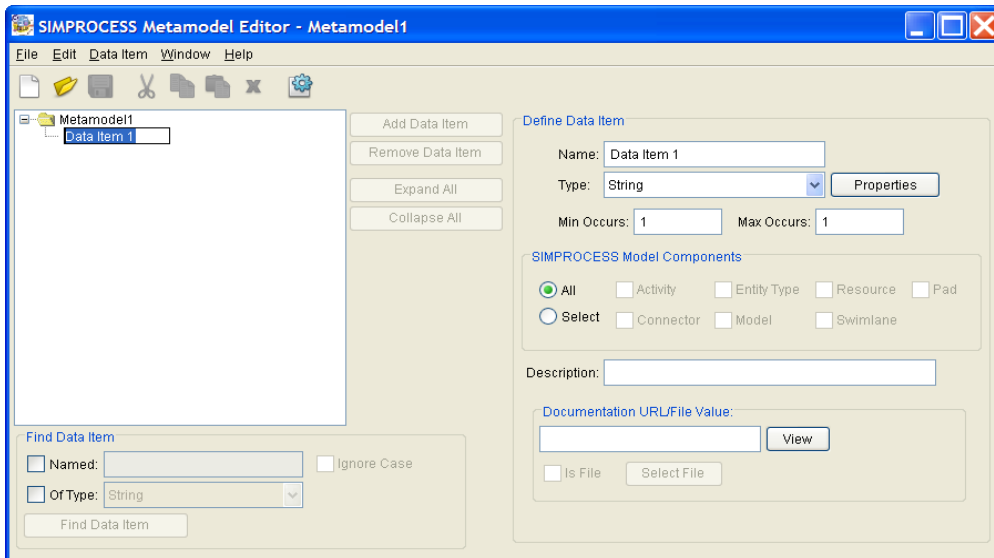
A new metamodel contains only the top node of the tree, the name of which is the default name of the new metamodel.



Since the top node of the tree is selected by default, the **Add Data Item** button and menu item are active. Selecting either of these adds a new data item definition as a child of the top node. The new data item definition will have a default name. Use the **Define Data Item** section to set the properties of the new data item definition.

Defining a Data Item

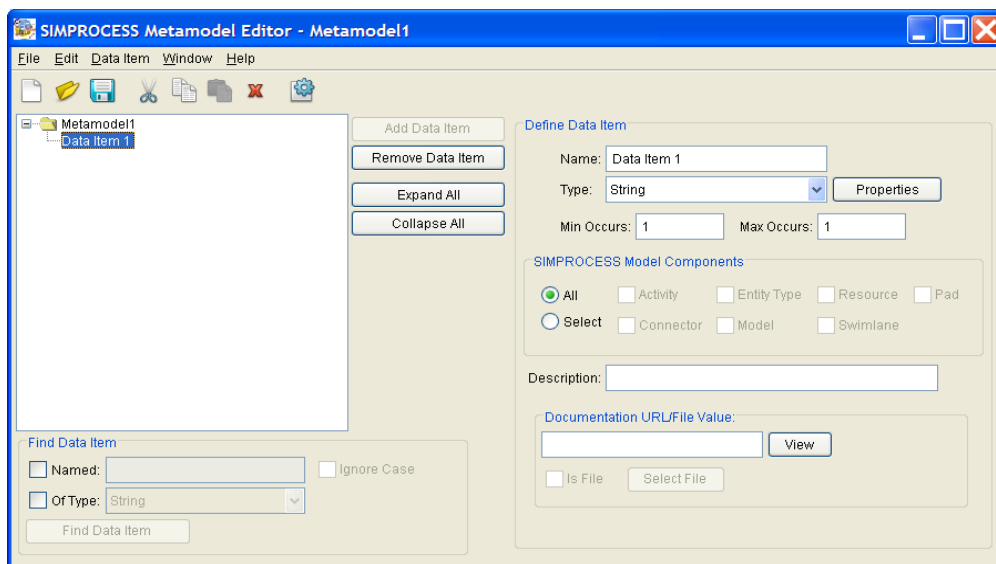
The data item name can be changed by editing the **Name** field. The data item name can consist of any string of characters, numbers, or symbols. However, it is recommended that special characters (especially **&**, **{ }**, and **[]**) be avoided. Data item names can also be edited directly in the data item tree. Once a data item has been selected, simply click the data item again to enable editing. Alternatively, triple clicking will enable editing of the name, and clicking four times will enable editing and select the name of the data item being edited.



While a data item name is being edited directly in the data item tree, several menu items and buttons are disabled. The edit is accepted by pressing the Return/Enter key or by removing the focus from the data item with the mouse or the Tab key. The edit is canceled by pressing the Escape key. Note that the data item name must be unique within the data item definition's hierarchy. That is, the new name must not match the name of the metamodel, the name of the parent data item definition, or the names of any sibling or immediate child data item definitions.

Type defaults to *String*, but can be changed to *Real*, *Integer*, *Boolean*, *URL*, *Single Selection*, *Multiple Selection*, *Include*, or *Group*. A *URL* data item definition indicates that the metadata entered in a SIMPROCESS model for this data item will point to an object external to SIMPROCESS. *Single Selection* indicates that the SIMPROCESS model metadata for this data item will be one item selected from a list of items. *Multiple Selection* indicates that the SIMPROCESS model metadata for this data item will be zero or more items selected from a list of items. The *Include* selection allows another metamodel to be included as a child of this data item when metadata is entered in SIMPROCESS. Selecting *Group* as the type allows the data item definition to have child data item definitions and will cause the **Add Data Item** button and menu item to enable. Also, *Group* disables the **Properties** button next to the **Type** selection. The **Properties** button is used to set optional properties for *String*, *Integer*, *Real*, *Boolean*, and *URL* data items, to define the lists for *Single Selection* and *Multiple Selection* data items, and to set the default metamodel for the *Include* type. The optional properties allow all types except *Include* to function similar to *Group* data items by having child data items. (See “[Data Item Properties](#),” beginning on page 20 for more information on data item properties including how to setup *Single Selection* and *Multiple Selection* data

items.)



Min Occurs sets the minimum number of actual data items that will appear in the SIMPROCESS model and defaults to one. A value of zero indicates that this data item is optional. **Max Occurs** sets the maximum number of actual data items that will appear in the SIMPROCESS model and defaults to one. A value of zero indicates that the maximum number is unbounded. For example, if the **Name** was **Street** and **Min Occurs** and **Max Occurs** were both two, then there would be two data items available to enter **Street** metadata in an associated SIMPROCESS model. **Street [1]** might contain something like “200 Corporate Parkway”, and **Street [2]** might contain something like “Suite 100”. (Note that the indices will be added automatically. See “[Metadata Tree](#)” on page 43.)

SIMPROCESS Model Components sets the SIMPROCESS model components that can use this data item definition for metadata. If **All** is selected (default), then the individual check boxes for **Activity**, **Entity Type**, **Resource**, **Pad**, **Connector**, **Model**, and **Swimlane** will be disabled. These check boxes are enabled when **Select** is clicked. Child data item definitions will have every item under **SIMPROCESS Model Components** disabled since child data item definitions inherit the SIMPROCESS model component selections of the parent data item definition.

Description is for an optional short text description of the data item definition.

Documentation URL/Field Value is optional. Supporting documentation for the data item definition can be entered here. Information entered here will not appear in any SIMPROCESS model where the metamodel is used for metadata entry. If the entry is a URL, the **View** button will launch the preferred web browser. **Is File** must be selected if the entry is a file. The **Is File** option will be

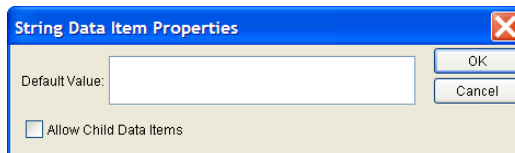
disabled until the metamodel is saved for the first time. This is because supporting documentation files must be located in the directory with the same name as the metamodel, and this directory is not created until the first save. When **Is File** is enabled and selected, the **Select File** button can be used to select a file from the directory of the metamodel. Note that supporting files must be manually copied into the directory named for the metamodel.

Data Item Properties

All **Type** properties are set or changed by clicking the **Properties** button. Default values can be set for all data types. The default values are applied when the metamodel is assigned to a SIMPROCESS model and metadata is entered. (See [“Applying Default Values,” beginning on page 51](#) for information on how default values are applied in a SIMPROCESS model, and see [“Assigning Metamodels,” beginning on page 30](#) for more information on assigning metamodels to a SIMPROCESS model.) **Real** and **Integer** data items also allow constraints to be set on the values that can be entered in a SIMPROCESS model. Every **Type** for which properties can be set except **Include** can be designated to allow child data items. Thus, data items with a **Type** other than **Group** can have child data items. This capability is enabled by selecting the **Allow Child Data Items** checkbox on the properties dialog.

String Data Item Properties

The **String Type** properties dialog has a field for entering the default value.



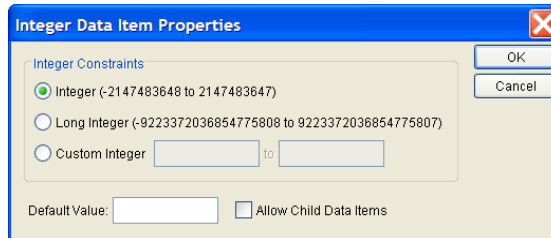
Real Data Item Properties

The **Real Type** properties dialog **Real Constraints** fields allow minimum and maximum values to be entered. The minimum must be less than the maximum. *Infinity* is assumed if the upper value field is empty. If the lower value field is empty, *-Infinity* is assumed. Below **Real Constraints** is the **Default Value** field. Leave this field empty if no default value should be used.



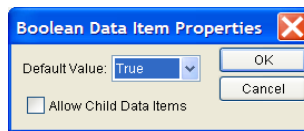
Integer Data Item Properties

The **Integer Type** properties dialog contains three buttons for setting constraints: **Integer**, **Long Integer**, and **Custom Integer**. **Integer** is the default selection. Select **Custom Integer** to set a unique range of allowable values. When selected, the fields to the right will enable allowing minimum and maximum values to be entered. The minimum must be less than the maximum. Below the **Integer Constraints** area is the **Default Value** field. Leave this field empty if no default value should be used.



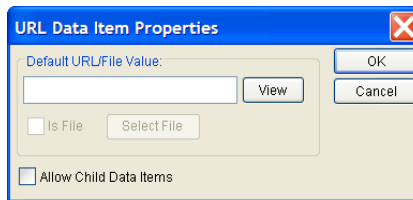
Boolean Data Item Properties

The **Boolean Type** properties dialog contains a single selection box from which **True** or **False** can be selected as the default value.



URL Data Item Properties

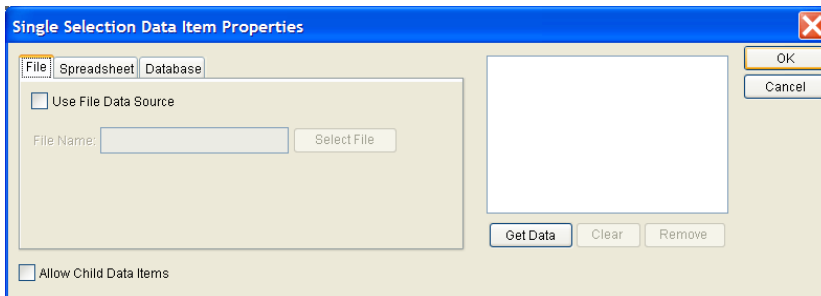
The **URL Type** properties dialog contains a field for the default URL. **Is File** should be selected if the URL represents a file. The **Select File** button enables when **Is File** is selected. **Is File** will be disabled until the model is saved for the first time since supporting files must be located in the directory with the same name as the metamodel.



Single and Multiple Selection Data Item Properties

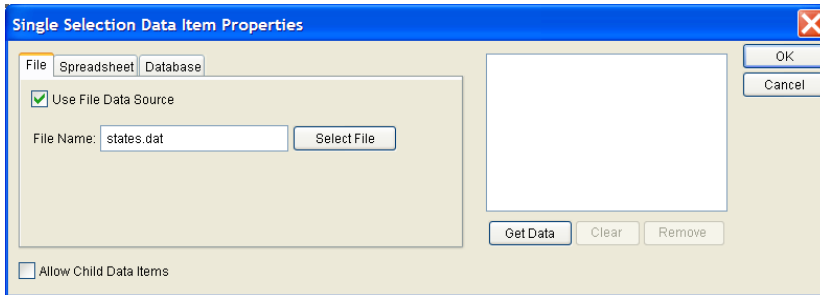
Single Selection and Multiple Selection **Types** offer a list of items from which to choose. In SIMPROCESS, the Single Selection **Type** displays a selection box (like the **Type** selection box) from which only one item can be selected. The Multiple Selection **Type** displays a scrollable list from which items can be selected. Control-click an item to select or deselect it. There is no limit to the number of items that can be on a selection list; however, very large lists could cause performance issues when used in a SIMPROCESS model.

When the **Properties** button next to **Type** is selected, a dialog displays that allows the list to be populated.

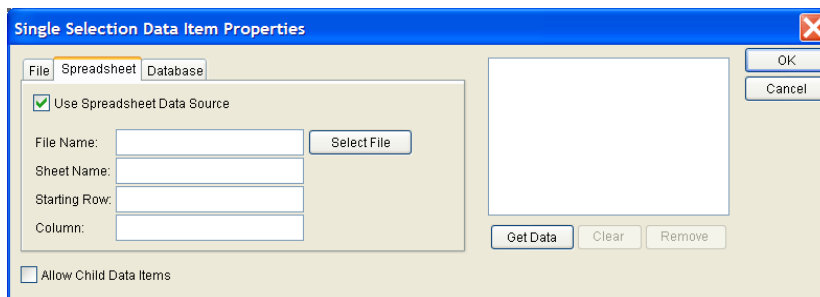


There are three tabs: **File**, **Spreadsheet**, and **Database**. These tabs represent the three types of data sources that can populate the list. One or more types of data source can be used. However, only one list will be created. Thus, if more than one data source is used, the data from all selected sources are combined into one list.

The **File** tab is used to specify an ASCII file as the data source. **Use File Data Source** must be selected to use this type of data source. The **Select File** button can be used to locate the file. Each line of the file will be considered an item of the list. Data source files must be located in the directory with the same name as the metamodel. Thus, the **Select File** button will be disabled until the metamodel is saved for the first time.

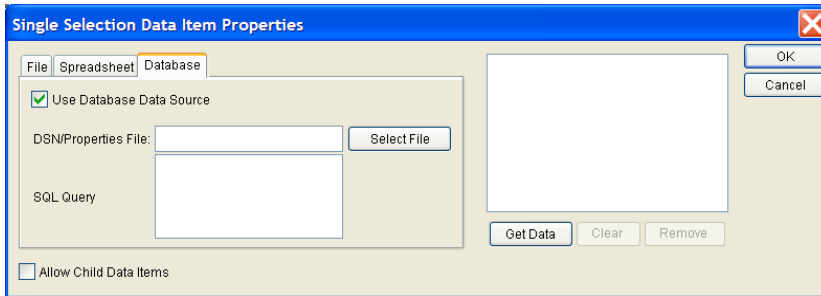


The **Spreadsheet** tab specifies a spreadsheet as a data source. The Spreadsheet can be a Workbook file or an XML Spreadsheet. Once **Use Spreadsheet Data Source** has been selected, four fields are required: **File Name**, **Sheet Name**, **Starting Row**, and **Column**. The **Select File** button can be used to locate the spreadsheet file. As with the **File** tab, the **Select File** button will be disabled until the metamodel is saved for the first time. Enter the name of the worksheet in the **Sheet Name** field. **Starting Row** should have the row number (rows start with 1) that contains the first value. **Column** should have the column number (column A would be column number 1) of the sample data. The data must be in a single column with no empty cells. The data is read beginning at the **Starting Row** and continues until an empty cell is reached.

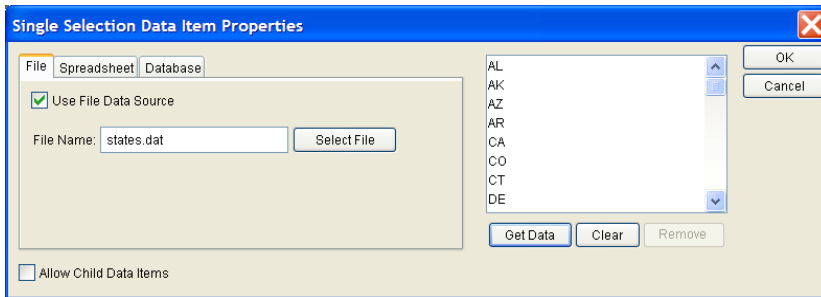


The **Database** tab is used to specify an SQL database as the data source. **Use Database Data Source** must be selected to use this type of data source. Two items are required: **DSN/Properties File** and **SQL Query**. **DSN/Properties File** must contain the database location/connection information. **DSN** stands for Data Source Name and is the **DSN** established in the Windows ODBC Control Panel. (See the **Setting Up Database Using Windows and Open Database Connectivity (ODBC)** section of Chapter 10 of the *SIMPROCESS User's Manual* for more information on using a DSN.) **Properties File** is the name of the properties file that contains the JDBC url and driver information and, if necessary, username and password. A **Properties File** must be used on a non-Windows system. After the metamodel has been saved for the first time, the **Select File** button can be used to locate the properties file. **SQL Query** must contain the query that returns the items to populate the

list. Note that if the query returns multiple columns, only the first column will be read for list data.



The list must be populated in the Metamodel Editor since it cannot be populated from SIMPROCESS. Empty lists will be displayed in SIMPROCESS if the list items have not been retrieved from the data sources. Note that an empty data display area indicates an empty list. The **Get Data** button clears the current list then retrieves list items from the selected data sources.



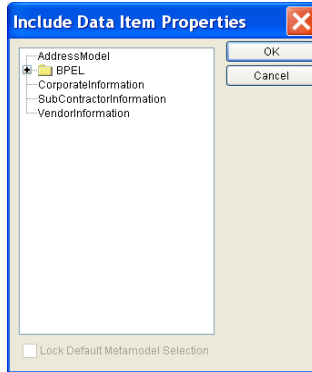
When items exist in the list, the **Clear** button is enabled. **Clear** removes all items from the list. When items in the list are selected, the **Remove** button is enabled. **Remove** removes the selected items from the list. Multiple items can be selected by holding down the Control key (or appropriate non-Windows system equivalent) during selection. If the list is lengthy, **Remove** can be used to eliminate unnecessary items to improve performance.

Any items selected in the list when **OK** is clicked will be considered the default selections for this data item. An error will occur if there is more than one item selected in the list for a **Single Selection** data item. To remove a selection without selecting another item, hold down the Control key (or appropriate non-Windows system equivalent) and click the selected item.

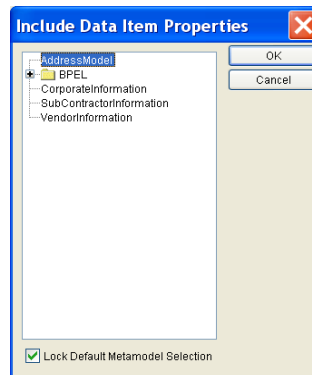
If **Max Occurs** is greater than one for **Single Selection** or **Multiple Selection Types**, the same list will be used for each instance, but different selections can be made.

Include Data Item Properties

The **Include Type** properties dialog displays a list of the metamodels in the metamodel directory.



A single metamodel can be selected from the list. The selected metamodel is the default metamodel for this data item. When a metamodel is selected, the **Lock Default Metamodel Selection** checkbox enables.



If **Lock Default Metamodel Selection** is selected, the selected default metamodel cannot be changed in SIMPROCESS.

Group Data Item Properties

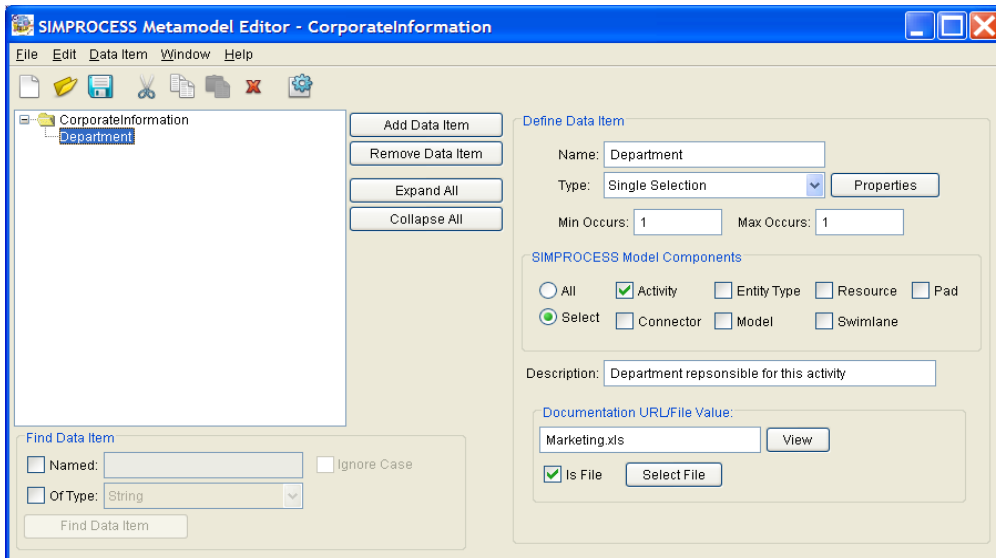
Additional properties cannot be set on **Group** data items. Their purpose is to collect related data items under a single data item.

Moving Data Items

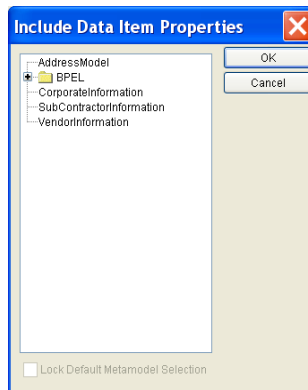
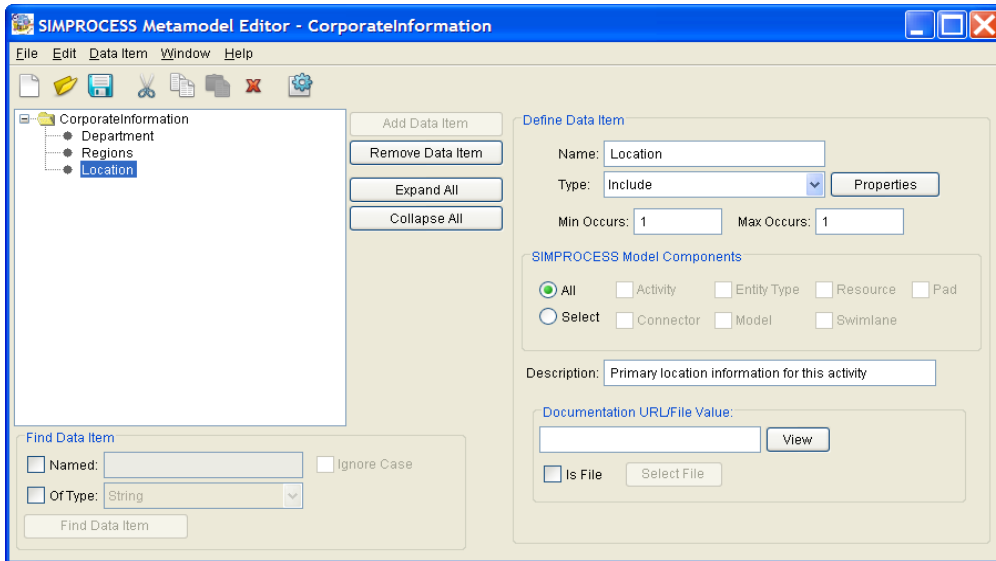
Data items can be rearranged by selecting one or more data items in the data item tree and dragging them to a new location in the data item tree. If more than one data item is selected, the selected data items must be contiguous. Note that before a move can begin the item (or items) to be moved must be selected, then clicked on a second time to actually drag. Data items can be moved within the same level or to a different level. When moving a data item to a different level, only the metamodel name and data items that can have child data items will highlight. When a data item is dropped on a highlighted data item, the moved data item is appended to the end of the data items that are direct descendants of the highlighted data item. Data items can be moved to a specific location by inserting the moved data item between two existing data items. Note that a `Group` data item or any data item with child data items cannot be moved to within its own descendants. A move can be canceled by returning the moved item to its original location or by moving the item off of the Metamodel Editor application. Data items cannot be moved between open metamodels. **Cut** and **Paste** must be used to move a data item from one metamodel to another.

Metamodel Example

In the example below, the default **Name**, **Data Item 1**, was changed to **Department**. **Type** was set to `Single Selection`, and **Min Occurs** and **Max Occurs** were both left at the default of one. **SIMPROCESS Model Components** was set so this data item definition will only apply to SIMPROCESS Activities, and a **Description** was entered. `Marketing.xls` was selected as the documentation file for **Documentation URL/File Values** after the metamodel was saved with the name **CorporateInformation**. In the **Properties** for **Department**, **Allow Child Data Items** was selected.



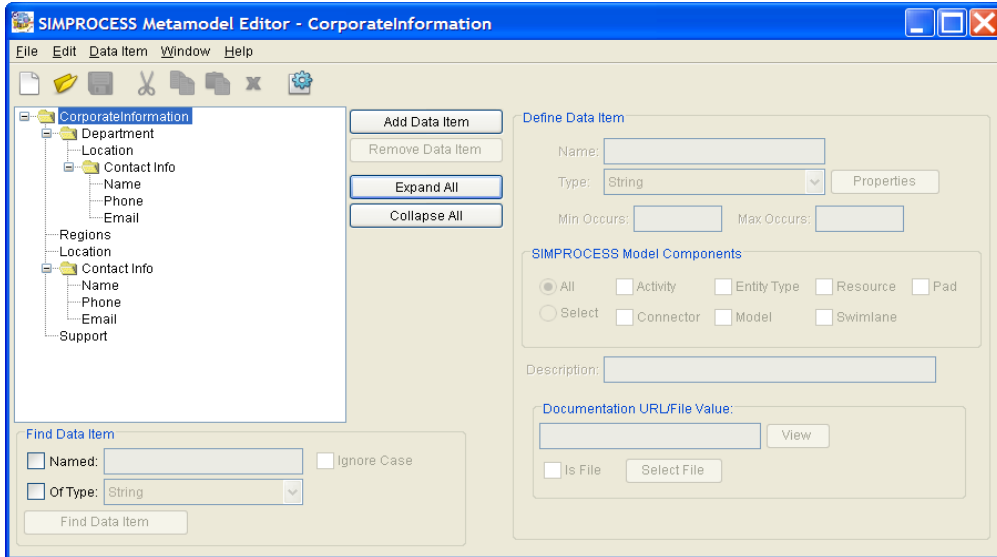
Even though the **Type** of **Department** is not **Group**, **Department** can have child data items so the **Add Data Item** button and menu item are active. To add another data item definition, the top node (**CorporateInformation**) must be selected. In the example below, three other data item definitions were added and their properties set. The data item **Regions** was added and its **Type** set to **Multiple Selection**. The data item definition **Location** could be defined by setting its **Type** to **Group** and adding several child data item definitions (**Street**, **City**, **State**, and **Zip Code**). However, the metamodel **AddressModel** contains the same data item definitions so the **Type** of **Location** was set to **Include** and **AddressModel** was selected in the properties. Although not visible, in **AddressModel** **Street** has **Min Occurs** set to one and **Max Occurs** set to two, and **State** has **Type** set to **Single Selection**.



Another data item definition was added to the top node. This data item definition was named **Contact Info** and its **Type** was set to **Group**. Its **Min Occurs** and **Max Occurs** were both set to zero. Thus, this data item is optional and the number allowed is unbounded. Three child data item definitions were added to **Contact Info**: **Name**, **Phone**, and **Email**. The last data item definition is **Support**, which is an **Include Type** data item. No default metamodel was set.

To complete the metamodel, **Location** and **Contact Info** were copied and pasted on the **Department** data item definition. This allows for separate location and contact information for the **Department**.

The completed **CorporateInformation** metamodel, which is included in the SIMPROCESS installation in the metamodel directory, is shown below.



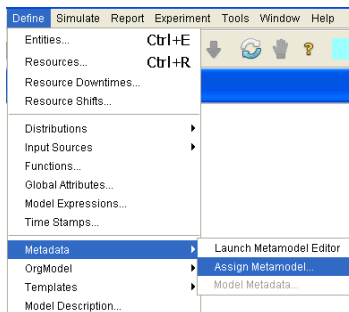
CHAPTER 3

Assigning Metamodels

This chapter describes how to assign one or more metamodels to a SIMPROCESS model.

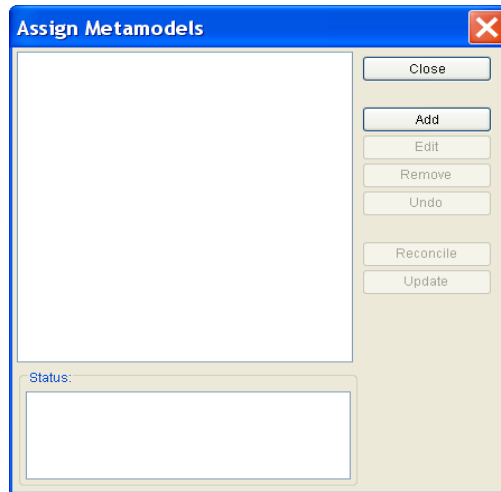
Assign Metamodel... Menu Item

Assigning metamodels to a SIMPROCESS model can only be done from the **Define/Metadata/Assign Metamodel...** menu item. This menu item is only active when a SIMPROCESS model is open.



Selecting **Assign Metamodel...** displays the **Assign Metamodels** dialog, which shows a list of the metamodels assigned to the active SIMPROCESS model. If no metamodels have been

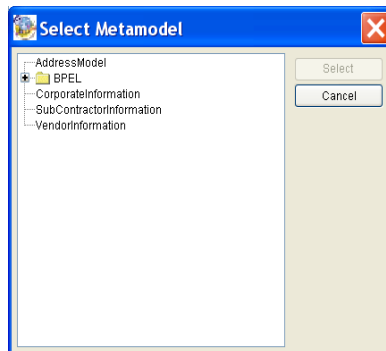
assigned, the list is empty.



Assign Metamodel Operation

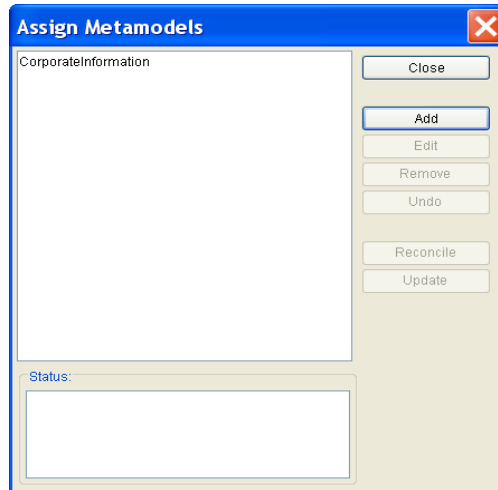
Adding Metamodels

Select **Add** to assign a metamodel to the active SIMPROCESS model. **Add** displays a list of the metamodels in the `metamodel` directory that have not already been assigned to the active SIMPROCESS model. (The `metamodel` directory is in the SIMPROCESS installation directory.)



Assign Metamodel Operation

Choose the desired metamodel then click **Select**, or double click the desired metamodel to assign it to the SIMPROCESS model.



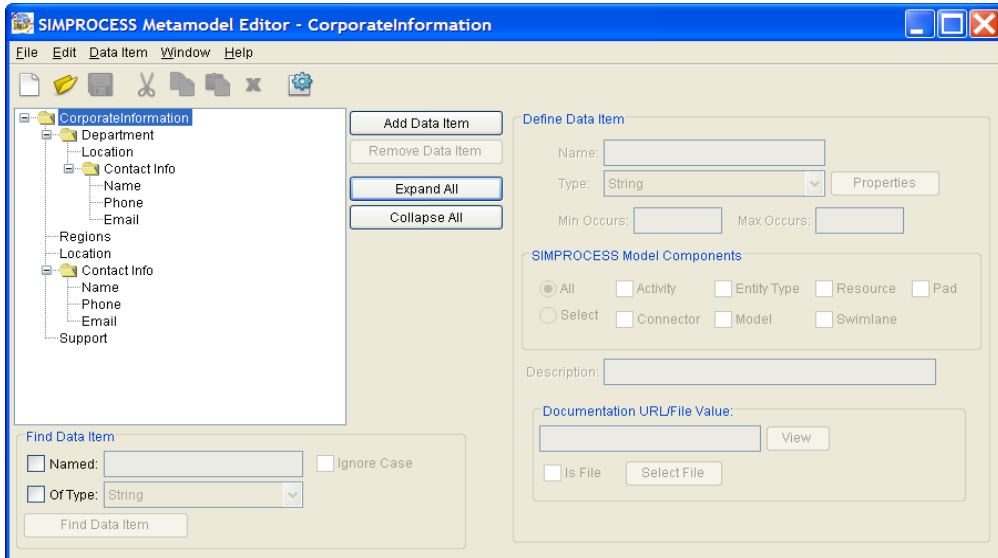
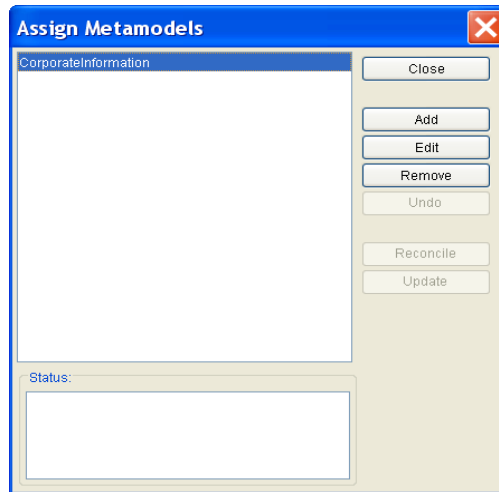
If the selected metamodel has errors, then an error message will appear and the metamodel will not be allowed to be assigned to the SIMPROCESS model. Note that if an assigned metamodel has `Include` data items, the metamodels referenced in the `Include` data items should not be assigned to the SIMPROCESS model unless one or more of those metamodels needs to be used in a stand alone capacity. (See [“Defining a Data Item” on page 17](#) and [“Include Data Item Properties” on page 25](#) for more information on the `Include` data item.)

Editing Metamodels

Selecting an assigned metamodel in the **Assign Metamodels** dialog will cause the **Edit** button to enable. Clicking **Edit** will cause the SIMPROCESS Metamodel Editor to start (or open a new window if already started) with the selected metamodel open in the editor window. Note that if the selected metamodel was already open, that window should become the active window. This is not guaranteed, however, due to differences in system configurations. If the metamodel has been removed from the `metamodel` directory, an editor window will open with a new metamodel that has the same name. (See [“SIMPROCESS Metamodel Editor,” beginning on page 9](#) for information on the SIMPROCESS Metamodel Editor.)

If the **Assign Metamodels** dialog is still open when the metamodel being edited is closed, a check will be made to see if the latest saved date and time of the metamodel matches the saved date and time stored with the metadata. Reconciling the metamodel with the associated metadata may be

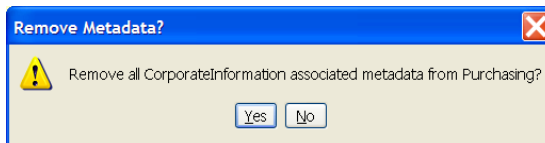
necessary (see “Reconciling Metamodels,” beginning on page 34).



Removing Metamodels

Selecting an assigned metamodel in the **Assign Metamodels** dialog will cause the **Remove** button to

enable. If no metadata associated with the selected metamodel has been added to the SIMPROCESS model, clicking **Remove** will remove the selected metamodel from the list, thus unassigning the metamodel from the SIMPROCESS model. If not already active, the **Undo** button will enable. If metadata has been added to the SIMPROCESS model a dialog will appear asking whether or not to remove all associated metadata. Selecting **Yes** will remove all associated metadata from the SIMPROCESS model. Selecting **No** will leave the metadata. See [“Reconcile/Remove Button” on page 52](#) for the implications of leaving metadata in a SIMPROCESS model that is not associated with a metamodel.



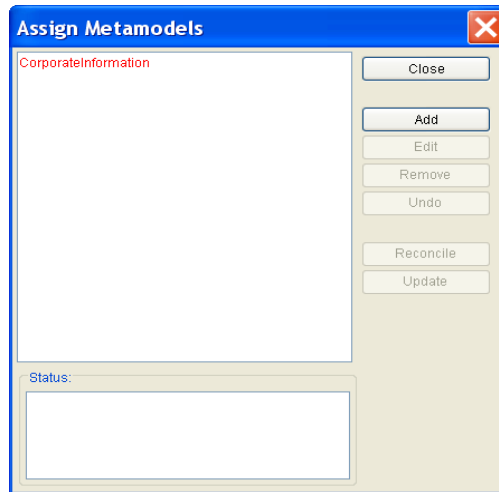
Undo Remove Metamodel

The **Undo** button is active when one or more associated metamodels have been removed. Selecting **Undo** will restore the last metamodel that was removed. Thus, **Undo** restores metamodels in the opposite order from which they were removed. Note that if the associated metadata was removed when metamodel was removed, restoring the metamodel will also restore the associated metadata.

Reconciling Metamodels

Assigned Metamodels

Every time a metamodel is saved, the date and time of the save is stored as part of the metamodel. When a metamodel is associated with a SIMPROCESS model and metadata is entered, the last save date and time of the metamodel is stored with the associated metadata. (See [“Entering Metadata,” beginning on page 39](#) for information on entering metadata.) When **Assign Metamodel...** is selected, a check is performed that compares the last saved date and time of the assigned metamodels with the stored date and time of the metadata in the SIMPROCESS model. If there is a match, the metamodel name displays in the **Assign Metamodels** dialog in black. However, if the dates and times do not match, then the metamodels for which there is not a match will display in red. The metamodel name will also display in red if the assigned metamodel file is no longer in the `metamodel` directory or if the assigned metamodel is no longer valid. (Note that metamodels created or edited with the SIMPROCESS Metamodel Editor should always be valid.)



When a metamodel in red is selected, a message will appear in the **Status** field and the **Reconcile** button will enable.



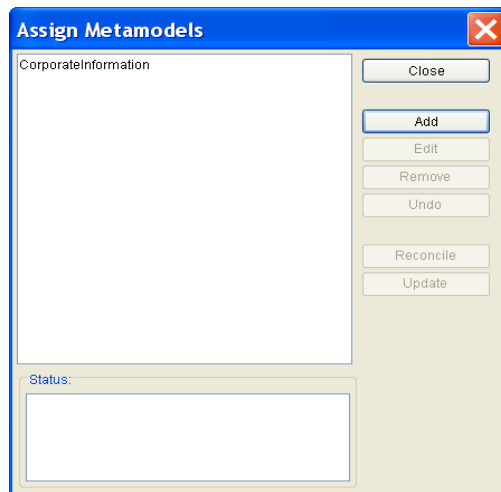
If the assigned metamodel is red because the last saved date and time does not match the stored date and time of metadata in the SIMPROCESS model, clicking **Reconcile** will reconcile all metadata associated with the selected metamodel. The reconciliation process consists of

- Removing metadata for which there is no longer a data item definition in the metamodel.

- Removing metadata in which the type (String, Real, Integer, Boolean, URL, Single Selection, Multiple Selection, Include, or Group) no longer matches the type of the data item definition in the metamodel. If the type of the data item definition has changed from Integer to Real, the metadata is not removed and the type of the metadata is changed to Real.
- Removing metadata for an index value that is larger than the value of **Max Occurs** for the data item definition in the metamodel. For example, if **Max Occurs** was reduced from two to one, the second metadata occurrence would be removed. (See “[Defining a Data Item,](#)” beginning on page 17 for information on **Max Occurs**.)
- Removing metadata for which the assigned **SIMPROCESS Model Component** does not match the **SIMPROCESS Model Component** selections of the data item definition in the metamodel. For example, if there is metadata at an Activity and the data item definition in the metamodel no longer has **All** or **Activity** selected, the metadata will be removed from the Activity. (See “[Defining a Data Item,](#)” beginning on page 17 for information on **SIMPROCESS Model Component** selection.)

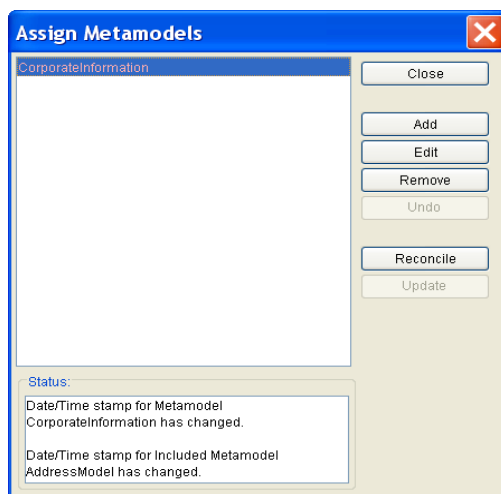
If the assigned metamodel is red because the metamodel file is not valid, **Reconcile** will function the same as **Remove** with the exception that **Undo** will not be available. **Undo** is not available for any **Reconcile** action. (See “[Removing Metamodels,](#)” beginning on page 33 for information on **Remove**.)

Once the reconciliation process is complete (and removal of the assigned metamodel was not necessary) the assigned metamodel name will display in black with no message in the **Status** field.



Included Metamodels

Included metamodels are metamodels added to an existing metamodel by using the `Include` data item type (see “[Defining a Data Item](#)” on page 17). As with assigned metamodels, the last save date and time of an included metamodel is stored with the associated metadata. When **Assign Metamodel...** is selected, not only is a check performed that compares the last saved date and time of the assigned metamodels with the stored date and time of the metadata in the SIMPROCESS model, but the same check is also done for all included metamodels. Thus, the assigned metamodel’s name will display in red if any included metamodels do not pass the check or if any included metamodels are missing or invalid. When the assigned metamodel is selected, multiple status messages may display in the Status area. In the example below, the assigned metamodel **CorporateInformation** and the included metamodel **AddressModel** both have a saved date and time that does not match the stored data and time.



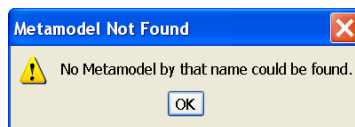
The reconciliation process for included metamodels is the same as the reconciliation process for assigned metamodels. When **Reconcile** is clicked, all of the actions listed above in the **Assigned Metamodels** discussion are performed for included metamodels. Thus, **Reconcile** addresses all inconsistencies at the same time.

Updating Metamodels

If an assigned or included metamodel file has been moved, the assigned metamodel will be in red, and when the assigned metamodel is selected the **Update** button will enable.



Update attempts to find the missing metamodel file. The `metamodel` directory and all subdirectories are searched. If the metamodel file is found the location is updated. Once found a **Reconcile** may be needed (see [“Reconciling Metamodels,” beginning on page 34](#)). If not, no further action is required. If the metamodel file is not located a warning is displayed.



When this happens the file can be manually located, placed back in its original location, and **Update** selected again or **Reconcile** or **Remove** can be selected (both of which will remove the metamodel assignment).

CHAPTER 4

Entering Metadata

This chapter describes how to enter metadata in a SIMPROCESS model.

Prerequisites

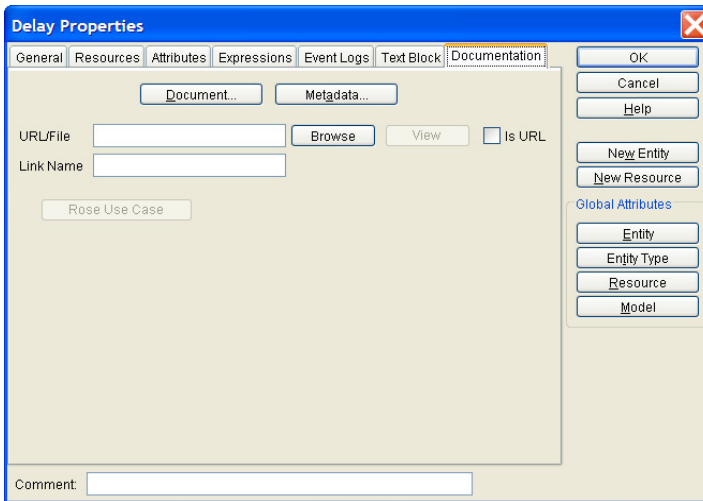
Before metadata can be entered into a SIMPROCESS model, at least one metamodel must have been created ([Chapter 2](#)) and assigned to the SIMPROCESS model ([Chapter 3](#)).

Once a metamodel has been created it can be assigned to multiple SIMPROCESS models, and a SIMPROCESS model can have multiple metamodels assigned to it.

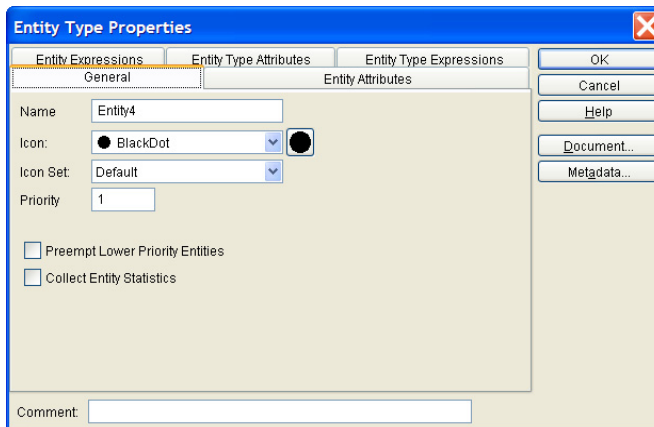
Metadata Entry Point

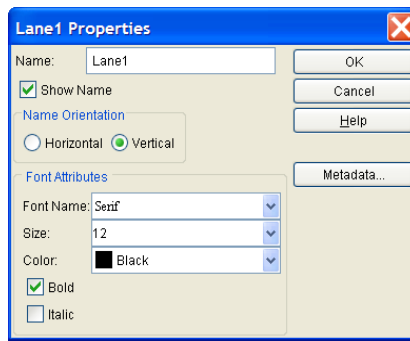
Metadata can be added to the properties of a SIMPROCESS model component (Activity, Entity Type, Resource, Pad, Connector, or Swimlane) by clicking the **Metadata...** button.

For Activities and Processes the **Metadata...** button is located on the **Documentation** tab of every Activity properties dialog. Metadata is added to the Model component by selecting the **Define/Metadata/Model Metadata...** menu item. Metadata can also be added to Activities, Processes, Pads, and Connectors by selecting **Edit Metadata** on the contextual menu.



The **Metadata...** button is located just below the **Document...** button on the properties dialogs of Entity Types, Resources, Pads, and Connectors. For Swimlanes, the **Metadata...** button is located just below the **Help** button. Below are shown the properties dialogs for an Entity Type, a Pad and a Swimlane.



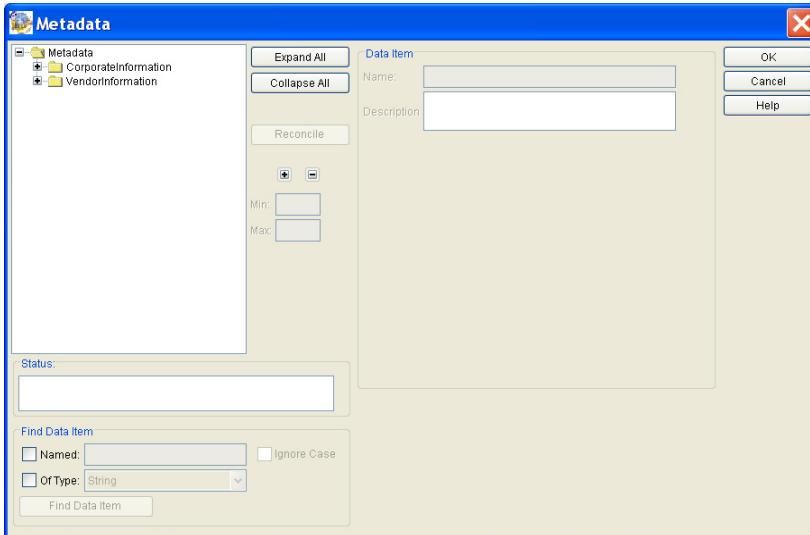


The **Metadata...** button and the **Define/Metadata/Model Metadata** menu item will only be active if there is at least one metamodel assigned to the SIMPROCESS model or metadata already exists for the component being edited.

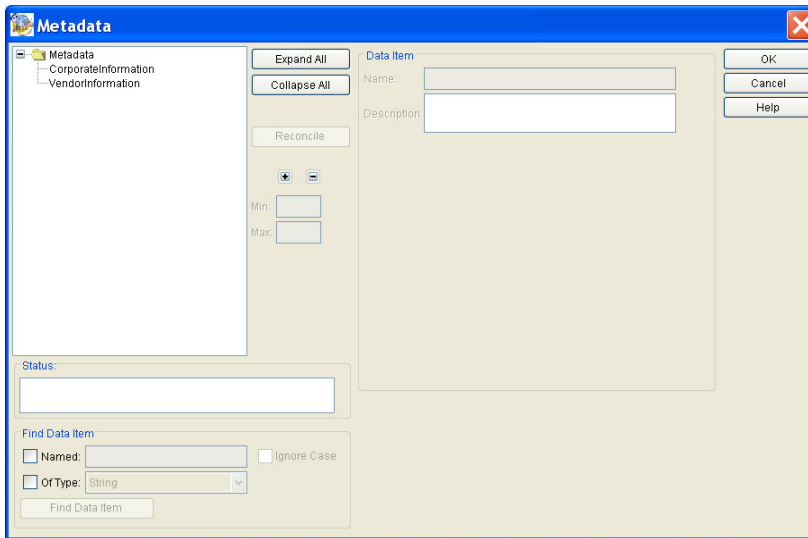
Metadata Dialog

Clicking the **Metadata...** button displays the **Metadata** dialog for entering metadata. The format of the dialog is the same for all SIMPROCESS model components. If the SIMPROCESS model component being edited has data item definitions for this type of component in any of the assigned metamodels, the appropriate data items will be listed in the **Metadata** tree.

Metadata Dialog



If no assigned metamodels have data item definitions that apply to the type of SIMPROCESS model component being edited, the **Metadata** dialog will have a **Metadata** tree that just lists the assigned metamodels and no metadata can be entered.

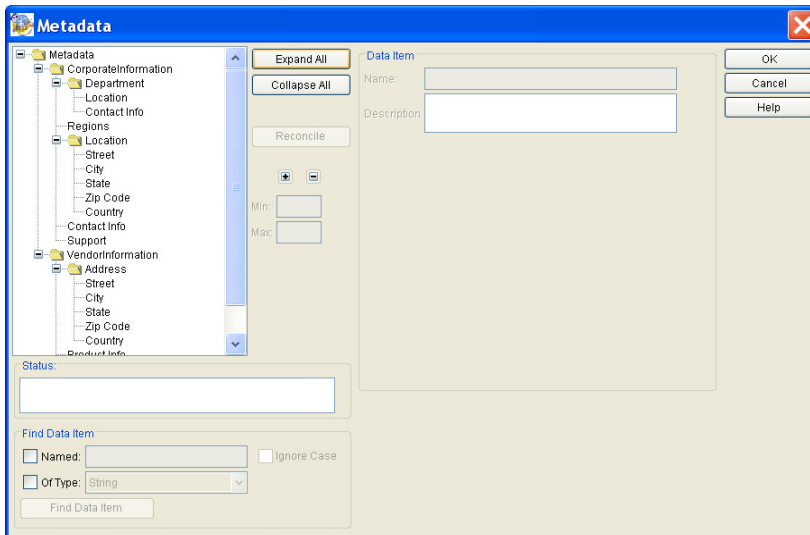


Metadata Dialog Description

The format of the Metadata dialog is similar to the format of the SIMPROCESS Metamodel Editor (see “[Metamodel Editor Description](#)” on page 10). On the left is the metadata tree, and on the right is the **Data Item** section, where the metadata is entered.

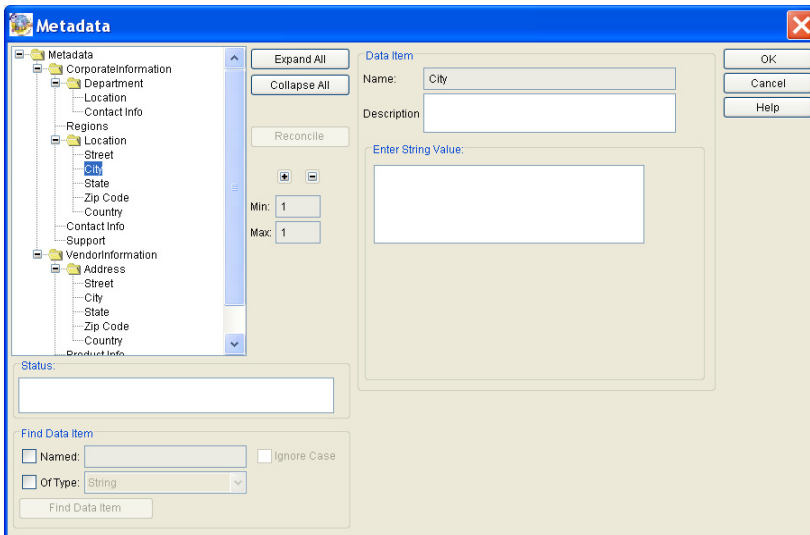
Metadata Tree

The top node of the metadata tree is always **Metadata**, and the top level child nodes are always the names of the assigned metamodels. The remaining items in the metadata tree are data item definitions from the assigned metamodels that apply to the type of SIMPROCESS model component being edited. Below the data item tree is the **Status** field (see “[Reconcile/Remove Button](#)” on page 52) and the **Find Data Item** section. The **Find Data Item** section allows searching for a particular data item by name and type. If **Named** is selected, a data item name to search for can be entered and **Ignore Case** is enabled for selection. Wild cards (*) can be used before and/or after the search text to search for similarly named data items. Indexed data items can be searched by including a space after the data item name and then the desired index number in brackets (for example **Street [2]**). If **Of Type** is selected, String, Real, Integer, Boolean, URL, Single Selection, Multiple Selection, Include, or Group can be selected. The search will then look for data items with a type matching the selected type. When both **Named** and **Of Type** are selected, the search is an “and” search. That is, the search will look for data items that match the entered text and the selected type. The **Expand All** button expands the complete metadata tree, and the **Collapse All** button restores the metadata tree to the top level.



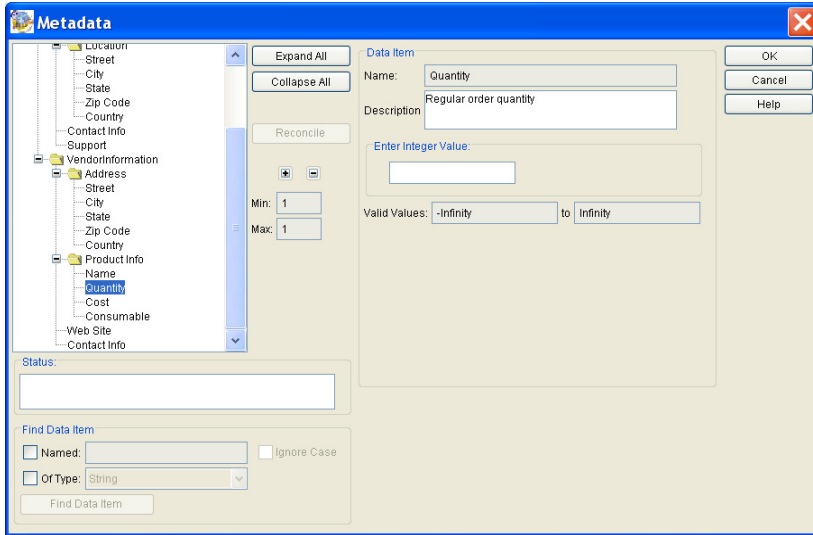
Data Item Section

The **Data Item** section contains the **Name** and **Description** of the data item. If no data item is selected in the metadata tree, or the item selected is a **Group** data item, **metamodel** name, or the top node, nothing else will display in the **Data Item** section. Also, nothing else will display in the **Data Item** section if the **Min Occurs** value of the data item definition in the associated metamodel is zero and an instance of the data item has not been added (see [“Adding and Removing Data Items,” beginning on page 49](#)). For all other items, when selected, a data entry field based on the type of the data item (**String**, **Real**, **Integer**, **Boolean**, **URL**, **Single Selection**, **Multiple Selection**, or **Include**) will display below the **Description**. The data entry field will be empty for **String**, **Real**, **Integer**, **Boolean**, and **URL** types if no default values were specified in the associated metamodel. The **Single Selection** type will display the first item of the possible selections if no default value was set in the metamodel, and the **Multiple Selection** type will show no selections if no default values were set in the metamodel. (See [“Data Item Properties,” beginning on page 20](#) for information on specifying default values.) **Real** and **Integer** will also display the valid range for the field. When a **String** data item is selected the **Enter String Value** field displays. **Include** data items will display a tree of metamodels in the metamodel directory. If a default metamodel was set in the owning metamodel, that metamodel will be selected. In addition, if the default selection was locked (see [“Include Data Item Properties” on page 25](#)), the list of metamodels will be disabled.

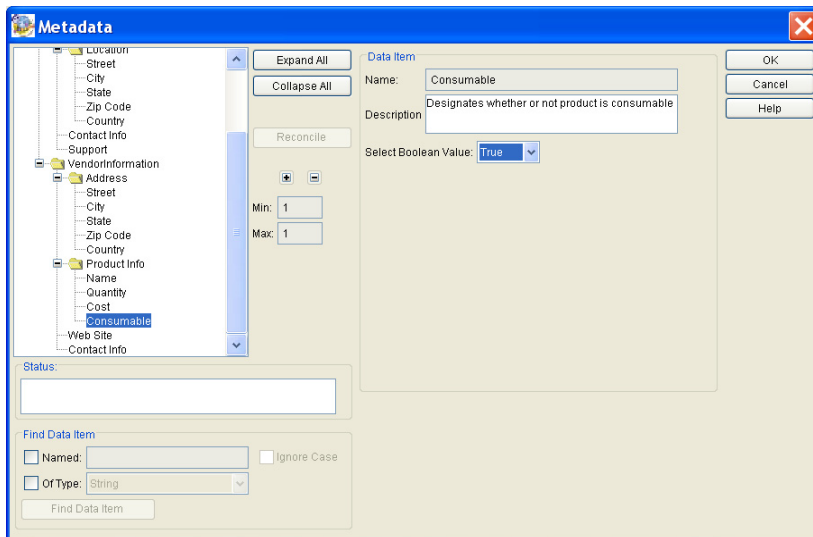


When an **Integer** data item is selected, the **Enter Integer Value** field displays, and when a **Real** data item is selected, the **Enter Real Value** field displays. These fields are similar in appearance but validation does occur to ensure that only **Integer** values are entered in the **Enter Integer Value**

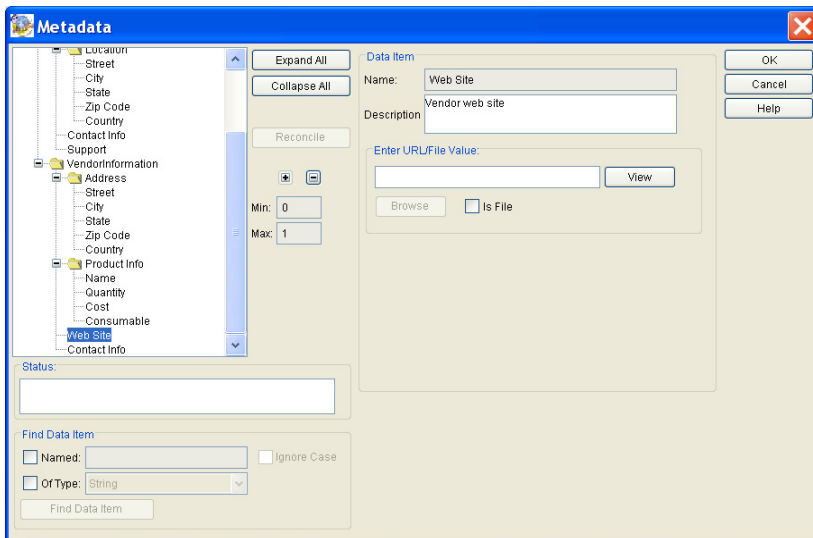
field and only Real values are entered in the **Enter Real Value** field. Also, validation ensures that the entered value is within the **Valid Values** range set in the metamodel.



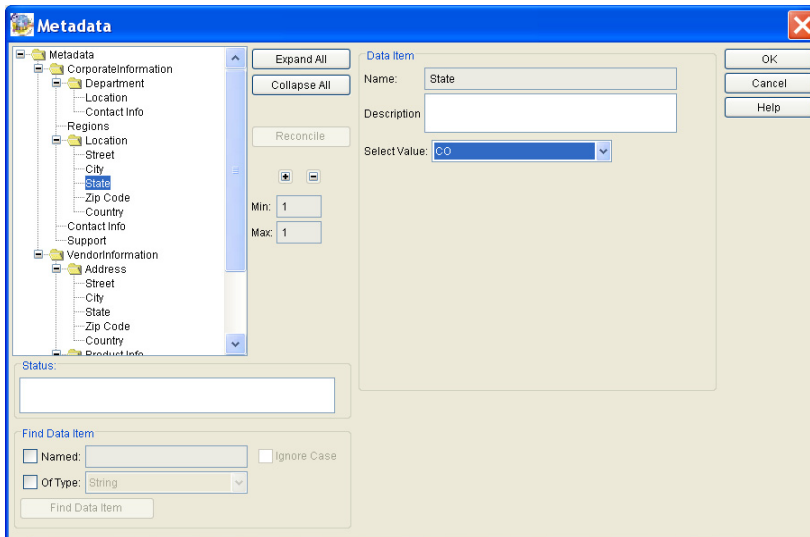
Selecting a Boolean data item causes the **Select Boolean Value** selection box to display. Only True or False can be selected.



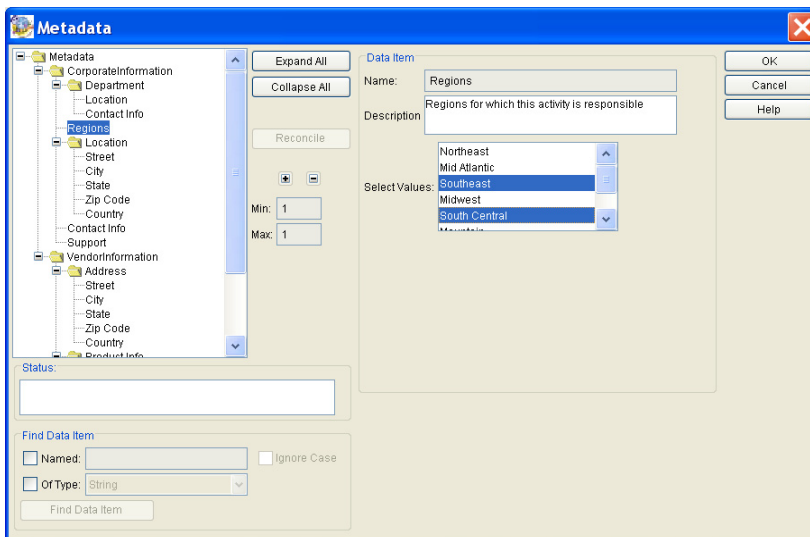
Selecting a URL data item causes the **Enter URL/File Value** panel to display. The **Enter URL/File Value** panel includes a field for entering text, a **Browse** button to select a file, a **View** button to open the URL or file, and a check box (**Is File**) that designates whether or not the text represents a file. The **Browse** button is only active when **Is File** is selected. The **View** button launches the preferred web browser if **Is File** is not selected. If **Is File** is selected, SIMPROCESS assumes the URL is a file and will attempt to open the file with whatever program is designated to open that type of file. For Windows, the **View** button enables when **Is File** is not selected and the text field is not empty, or **Is File** is selected and the text field indicates a path (contains backslashes or begins with . /). The same conditions apply on Linux systems. In addition, on Linux, the `view.properties` file must be in the `SPUser` directory, and the `view.properties` file must contain valid `url.view` and `file.view` properties. Due to other applications being required, the **View** button is not guaranteed to work in every case.



Selecting a `Single Selection` data item displays the **Select Value** selection box. Note that if no selection is made, the first item in the selection box is considered the selected item.

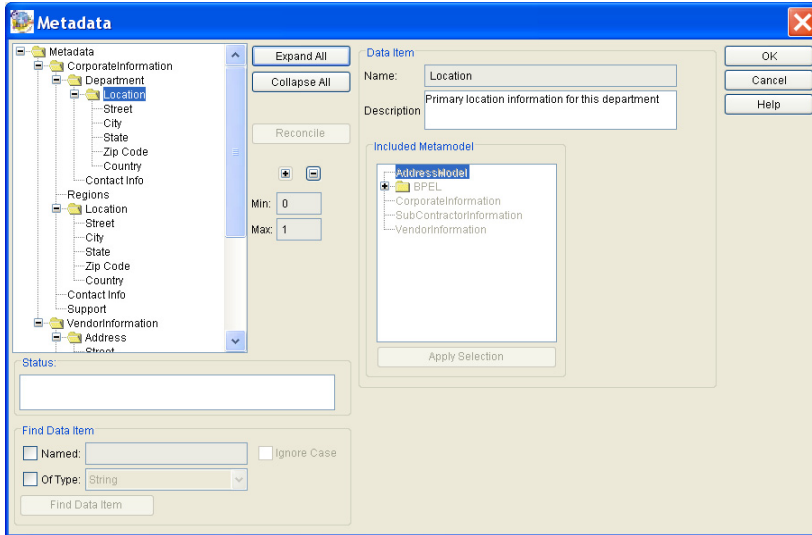


Selecting a Multiple Selection data item displays the **Select Values** list. Multiple selections can be made by holding down the Control key (or non-Windows system equivalent) while making selections.

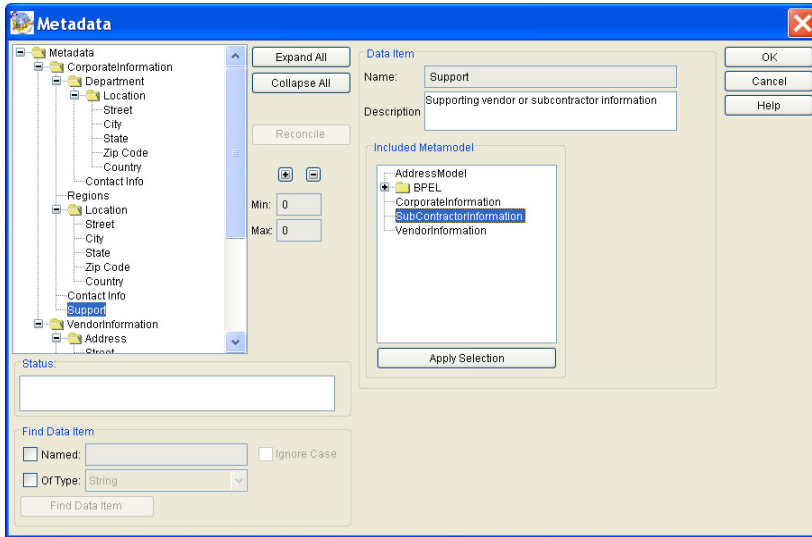


Selecting an Include data item will display a list of the metamodels in the `metamodel` directory. If

the default selection was locked (see “[Include Data Item Properties](#)” on page 25), the list of metamodels will be disabled.



If there was no default selection, or the default selection was not locked, a single metamodel can be selected for inclusion. The selected metamodel will not be included until **Apply Selection** is clicked. If the metamodel selection changes, all data items from the previously selected metamodel will be removed from the data item tree and replaced with data items from the currently selected metamodel.



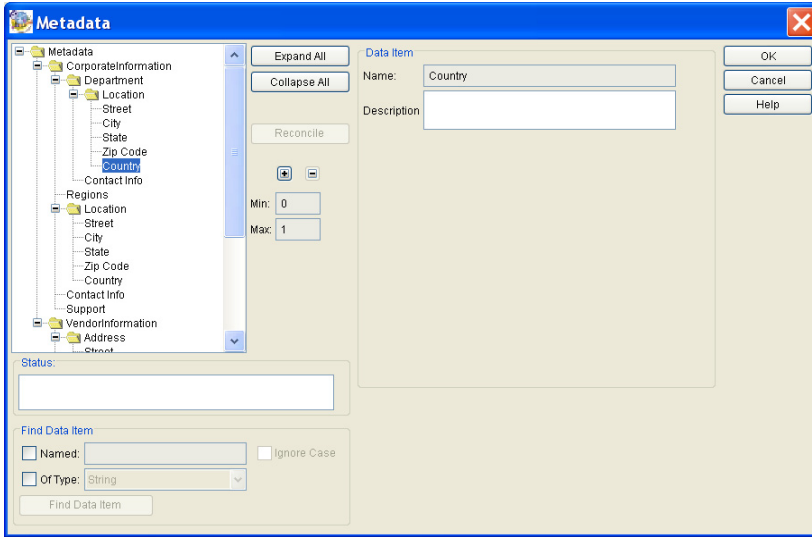
Adding and Removing Data Items

When data items are defined in a metamodel, the minimum and maximum number of occurrences of the data item are set using the **Min Occurs** and **Max Occurs** fields in the Metamodel Editor (see “Defining a Data Item” on page 17). These settings control how many instances of a data item are displayed in the data item tree. When a data item is selected, the **Min Occurs** and **Max Occurs** values are displayed in the **Min** and **Max** fields between the data item tree and the **Data Item** section. The plus button (+) and the minus button (-) above the **Min** field are used to add or remove instances of the selected data item. Those buttons will be enabled or disabled based on the **Min** and **Max** values and the number of instances of the selected data item already in the data item tree.

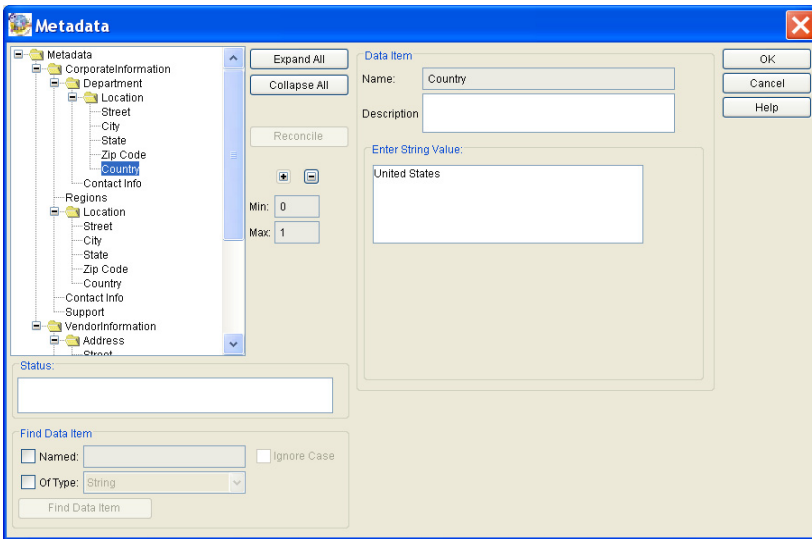
When the **Min** and **Max** values are both greater than zero and are equal, the plus and minus buttons will be disabled since no data items can be added or removed. In the example below, the data item **Regions** has a **Min** and **Max** of one, so both buttons are disabled.



A **Min** value of zero indicates that the data item is optional. In this instance, the data item will appear in the data item tree, but no metadata can be entered until an instance of the data item has been added. Below, the data item **Country** has a **Min** value of zero and no instance of that data item has been added. Thus, there are no data entry fields on the right and the plus button is enabled.



Clicking the plus button adds an instance of **Country**. The only changes are a String data entry field is added on the right with its default value “United States” and the plus button disables. It disables because the **Max** value is one.

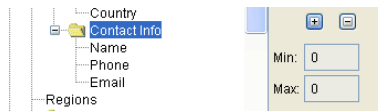


If the data item is an Include data item, Group data item or is one of the other types with child

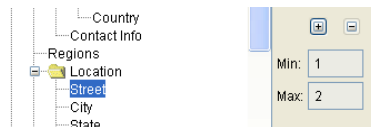
data items, the child data items are not displayed until an instance of that data item is added. Notice below that **Contact Info** has no child data items even though in the associated metamodel there are three child data items defined for **Contact Info**. Also, notice that the **Max** value is zero. This indicates that there is no limit on the number of occurrences of the data item. Thus, the plus button will always be enabled when an occurrence of a data item is selected that has a **Max** value of zero.



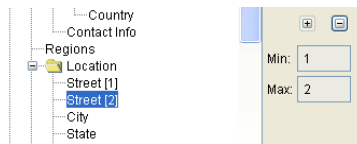
When an instance of **Contact Info** is added, its three child data items are added to the data item tree. Also, the plus and minus buttons are both enabled.



In the example below, since the data item **Street** has a **Min** of one and a **Max** of two, the plus button is enabled. Although not shown, there is a String data entry field on the right since an instance of the **Street** data item already exists (due to the fact that the **Min** value is one).



Clicking the plus button adds another **Street** instance. Notice that the **Street** data items now have indices. When there are more than one of any data item, indices will be appended to the names. Also, since the maximum has been reached, the plus button is disabled, and since there are more data items than the minimum allowed, the minus button is enabled.



Applying Default Values

Metamodels can contain default values for data items. (See [“Data Item Properties,”](#) beginning on

page 20, for information on specifying default values for data items.) The default value (or values) for a data item will display when the data item is selected in the data item tree and no value for the data item has previously been entered or selected. The displayed value can be changed from the default for all data item types except for `Include` when the default selection is locked (see “[Include Data Item Properties](#)” on page 25). When the dialog is closed using the **OK** button, all data items that have not been previously populated will have the default values from the associated metamodel saved in the metadata for the SIMPROCESS component being edited. Note that a data item is only populated with the default value if the data item has no value. This means that if the default value in the assigned metamodel changes, that change will not be automatically reflected in the SIMPROCESS metadata. If the default value for a certain data item is changed in the metamodel and that change should be reflected in instances of that data item in the SIMPROCESS model, then instances of that data item that contain a default value must be edited and their value manually changed to the new default value. Thus, the reconcile process described in “[Reconciling Metamodels](#),” beginning on page 34, will not update default values. Also, the reconcile process described in the next section will not update default values except for default metamodel selections in `Include` data items. Changes in default metamodel selections can be reconciled when either the default selection is locked or an unlocked default selection was not changed during metadata entry.

Reconcile/Remove Button

Below the **Collapse All** button is the **Reconcile/Remove** button. The text of the button starts out with **Reconcile**, but can change to **Remove** depending on the action of the button. As discussed in Chapter 3, it is possible that the underlying metamodel (or metamodels) for the entered metadata has changed or been removed since the metadata was entered. The **Assign Metamodels** dialog offers the option to globally remove metadata that is no longer supported by an assigned or included metamodel (see “[Reconciling Metamodels](#)” on page 34 for a full discussion). The same check that is performed when the **Assign Metamodels** dialog opens is performed when the **Metadata Dialog** opens. Data items that pass the check display in the metadata tree in black. Data items for which action may be required display in red. Situations which cause data items to display in red are

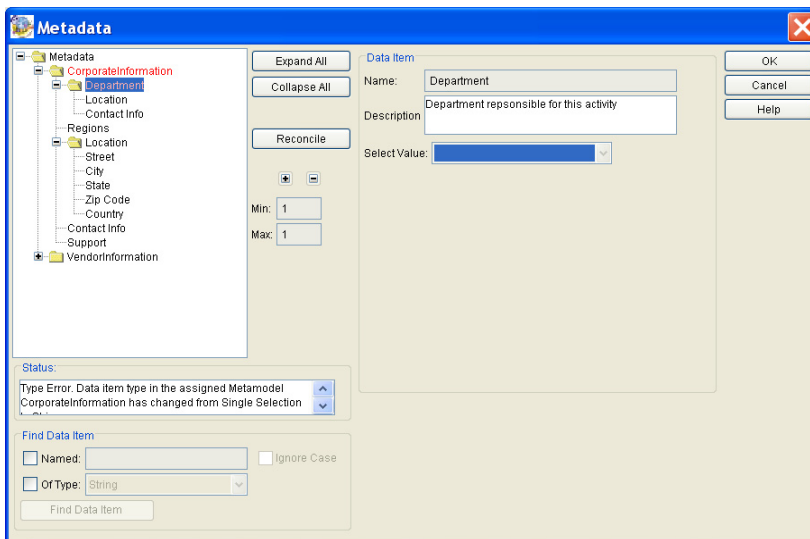
- Supporting metamodel no longer exists in the `metamodel` directory.
- Supporting metamodel is no longer valid.
- Data item definition for the data item no longer exists in the metamodel.
- The `type` (`String`, `Real`, `Integer`, `Boolean`, `URL`, `Single Selection`, `Multiple Selection`, `Include`, or `Group`) of the data item no longer matches the type of the data item definition in the metamodel. Note that if the type has changed from `Integer` to `Real`, the data item type is automatically changed to `Real`.

- The **Max Occurs** of the data item definition in the metamodel has been reduced and the index value of the data item exceeds the new **Max Occurs** value.
- The data item definition in the metamodel no longer applies to the type of SIMPROCESS component (Activity, Entity Type, etc.) being edited.

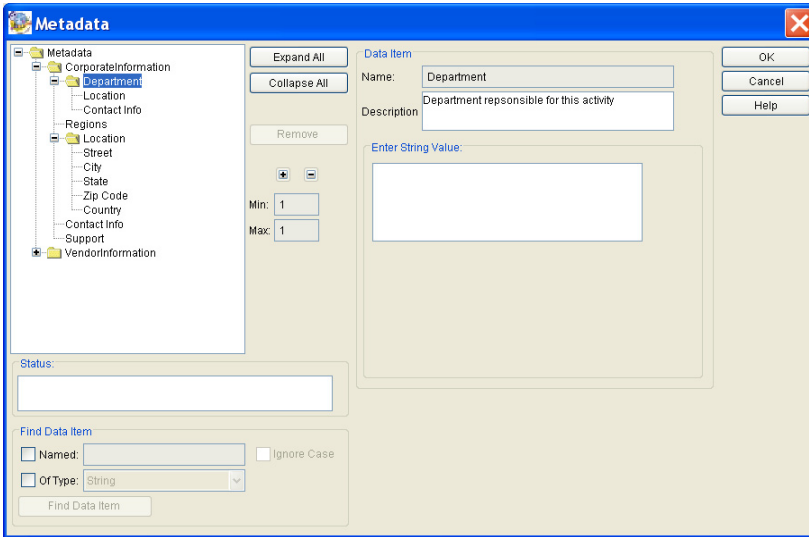
See “[Defining a Data Item](#)” on page 17 for a full discussion of a metamodel data item definition.

Reconcile

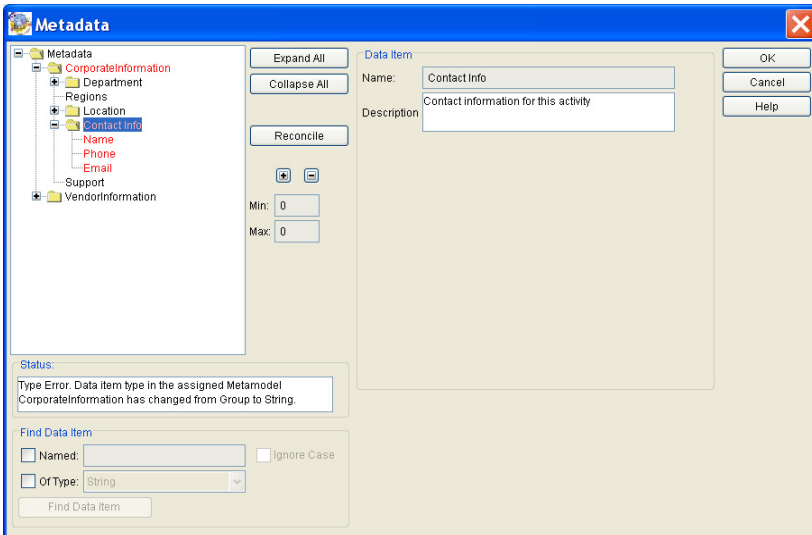
In the example below, the type of the **Department** data item definition has changed from `Single Selection` to `String`, thus the data item displays in red. Note that the immediate parent level node is also displayed in red. When selected, a message displays in the **Status** field, and the **Reconcile** button activates. The value of **Department** cannot be changed because the type has changed.



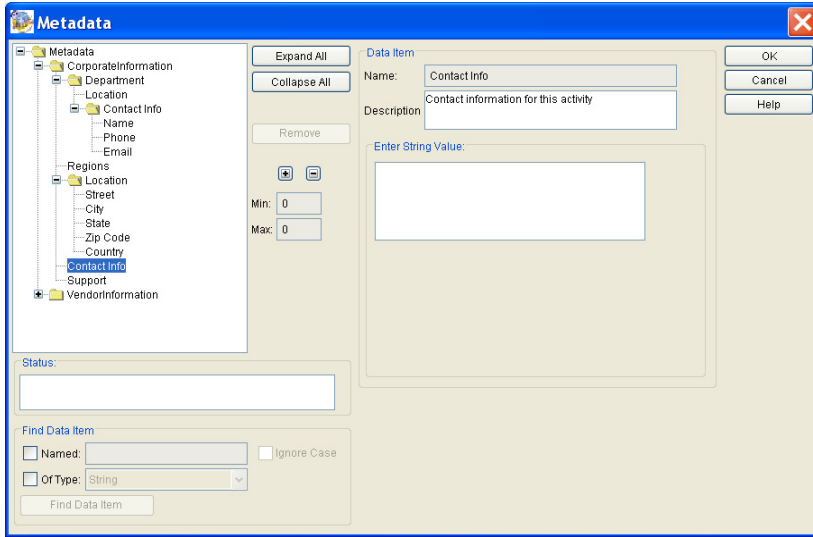
Clicking **Reconcile** removes the `Single Selection` selection box of **Department**, changes the type to `String`, and displays the **Enter String Value** field, which will accept text.



If the type of the data item definition has changed from Group, all child data items will be in red because the child data item definitions no longer exist. Reconciliation in this scenario will offer the option to remove all child data items. This is recommended since the parent data item is no longer a Group data item. In the example below, **Contact Info** was changed from Group to String.

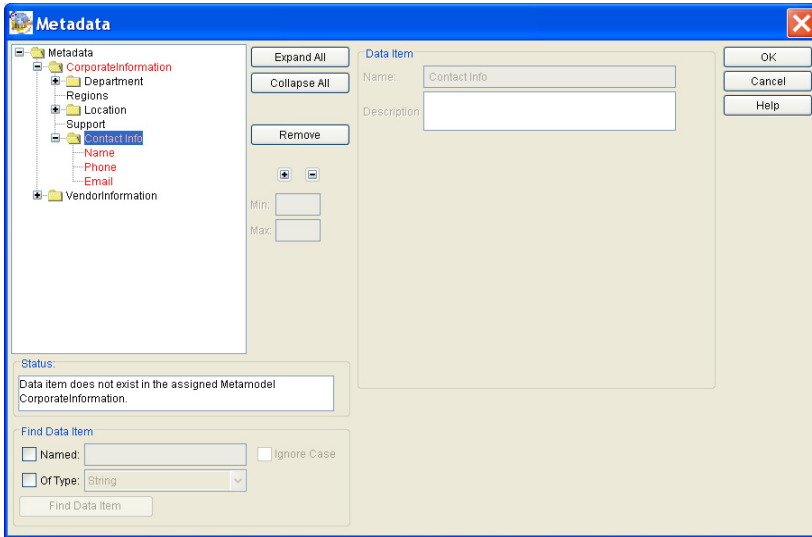


Clicking **Reconcile** will display a dialog confirming the deletion of the child data items. If the child data items are deleted, the following displays.

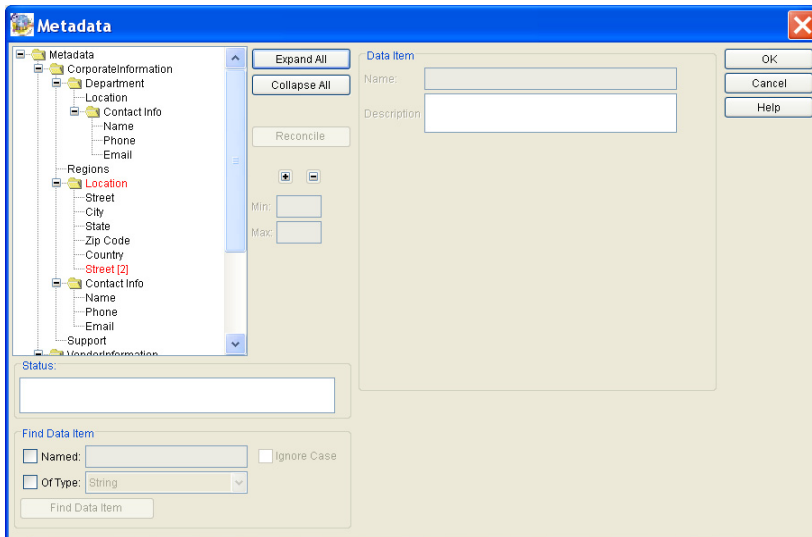


Remove

Some situations only allow the removal of the data item. In the example below the metamodel no longer defines the **Contact Info** data item (and thus its child data item definitions). Selecting **Contact Info** displays a message in the **Status** field and activates the **Reconcile** button with the text of the button changed to **Remove**.



The following example shows that the defined **Max Occurs** for **Street** was reduced to one. Thus, **Street [2]** is no longer supported by the **CorporateInformation** metamodel. Selecting **Street [2]** would cause the **Reconcile** button to enable with the text **Remove**.

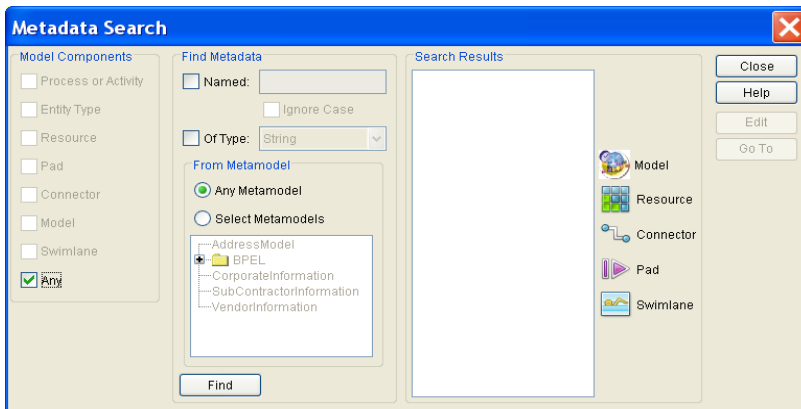


There is no undo for **Reconcile** or **Remove**. The only way to restore data items is to **Cancel** the

Metadata dialog or **Cancel** the properties dialog of the SIMPROCESS component being edited.

Metadata Search

Metadata Search... is on the SIMPROCESS **Edit** menu and, when selected, opens a dialog that searches model components for data items based on search criteria. Note that **Metadata Search...** searches for the names of data items in which metadata has been entered and not for metadata content. The **Metadata Search** dialog has three areas: **Model Components**, **Find Metadata**, and **Search Results**.



Model Components

Model Components has eight check boxes: **Process or Activity**, **Entity Type**, **Resource**, **Pad**, **Connector**, **Model**, **Swimlane**, or **Any** (default). The other items become available for selection when **Any** is deselected.

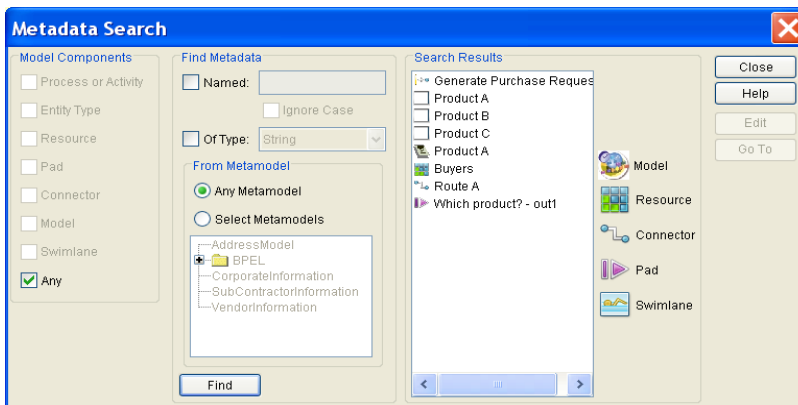
Find Metadata

Metadata Search searches the names of data items in the selected **Model Components** based on the options set in **Find Metadata**. To search for a data item with a specific name, select **Named** and enter the name of the data item to search for in the text field. Wild cards (*) may be used before and/or after the search text. To search for a specific index in a data item name, include a space and the desired index value in brackets, such as **Street [2]**. To search for a data item with a specific type (String, Integer, Real, Boolean, URL, Single Selection, Multiple

Selection, Include, or Group), select **Of Type** and choose the type from the drop down box. If **Named** and **Of Type** are both selected, then an “and” search is done. That is, the text in the **Named** text field and the type selected in **Of Type** must match the name and type of the data item. Select **Ignore Case** to perform a case insensitive search of the name. If **Any Metamodel** is selected, the name of the parent metamodel is ignored, thus, metadata for which the parent metamodel no longer exists will be searched. When **Select Metamodels** is clicked, the list of metamodels in the `metamodel` directory is enabled. If more than one metamodel should be included in the search, use the control key to select multiple metamodels. Once **Find Metadata** has been set, click the **Find** button to perform the search.

Search Results

Search Results displays the model components found. Note that model components containing the matching metadata are displayed in **Search Results**, not the matching metadata. The model components found (Process or Activity, Entity Type, Resource, Pad, Connector, Model, or Swimlane) are identified by an icon to the left of the model component’s name. Process, Activity, or Entity Type items display scaled down versions of their assigned icons. Resource, Pad, Connector, Model, or Swimlane display an icon based on the legend to the right of the list. If **Find** is clicked with **Named** and **Of Type** not selected, then items are found that have metadata of any name or type based on the **From Metamodel** selection. Thus, if **Any Metamodel** is selected, **Find** locates any metadata anywhere in the SIMPROCESS model no matter which metamodel was used to create the metadata. The example below shows a search for any metadata in a model. One Activity, three Processes, one Entity Type, one Resource, one Connector, and one Pad were found with metadata.



Selecting an item will cause the **Edit...** button to activate. The metadata of the item can be edited by clicking the **Edit...** button or by double-clicking an item. The **Go To** button will activate if the item selected is a Process, Activity, Pad, or Swimlane. Selecting **Go To** will take you to the level of the model

where that item is located, and, except for a Swimlane, select that item (highlight in red).

CHAPTER 5

BPEL Metadata

This chapter describes how to enter Business Process Execution Language (BPEL) metadata in a SIMPROCESS model and how to export that BPEL metadata to a BPEL process file. The instructions assume the user understands how to assign metamodels ([Chapter 3](#)), enter metadata ([Chapter 4](#)), add instances of a data item (“[Adding and Removing Data Items](#)” on page 49), and use `Include` data items (“[Data Item Section](#)” on page 44). Also, it is assumed that the user has a working knowledge of XML and BPEL.

Overview

BPEL is an XML-based programming language that is used to describe business processes. The business process could refer to a Web Service interaction between two businesses or between two elements in some business. This chapter assumes that BPEL related metadata will be entered for the purpose of exporting to a BPEL process file. However, in SIMPROCESS, there are no restrictions on how BPEL metadata is entered.

BPEL Metamodels

SIMPROCESS provides a collection of metamodels that can be used to store BPEL related information. The `BPEL` directory under the `metamodel` directory contains 28

metamodels for this purpose. These metamodels were developed based on the [Web Services Business Process Execution Language 2.0](#) specification and must not be changed. The BPEL metamodels can be divided into three categories: Primary, Activity, and Supporting.

Primary BPEL Metamodels

There are two primary BPEL metamodels:

- **BPEL Process**
- **BPEL Activity**

These two are designated as primary because they are the metamodels that should be assigned to a SIMPROCESS model. (See “[Assigning Metamodels](#),” beginning on page 30 for more information on metamodel assignment.) At a minimum, the **BPEL Process** metamodel must be assigned if the intention is to export to a BPEL process file from the metadata. **BPEL Process** contains data items for the attribute and elements of the top level BPEL <process> element and can include any level of BPEL activities. All information necessary for describing a business process using BPEL can be entered using **BPEL Process**. The **BPEL Activity** metamodel should be assigned if SIMPROCESS Activities represent activities in a BPEL process. Thus, BPEL activity metadata can be entered at select SIMPROCESS Activities.

Activity BPEL Metamodels

There are 21 Activity BPEL metamodels. These metamodels represent the 21 BPEL activities.

Metamodel	BPEL Activity Element
Assign	<assign>
Compensate	<compensate>
CompensateScope	<compensateScope>
Empty	<empty>
Exit	<exit>
ExtensionActivity	<extensionActivity>
Flow	<flow>
ForEach	<forEach>
IfActivity	<if>
Invoke	<invoke>
Pick	<pick>
Receive	<receive>

Metamodel	BPEL Activity Element
RepeatUntil	<repeatUntil>
Reply	<reply>
Rethrow	<rethrow>
Scope	<scope>
Sequence	<sequence>
Throw	<exit>
Validate	<validate>
Wait	<wait>
While	<while>

Although these metamodels can be assigned to a SIMPROCESS model, the intent is for these models to only be used for Activity and BPEL Activity data items and not assigned directly to a SIMPROCESS model.

Supporting BPEL Metamodels

There are five metamodels that are considered Supporting metamodels:

- **StandardBPELActivityOptions**
- **ExtensionActivityElement**
- **From-Spec**
- **To-Spec**
- **OpaqueActivity**

The **StandardBPELActivityOptions** metamodel contains data items representing the attributes and elements than any BPEL activity can have. Each Activity metamodel has a Standard BPEL Activity Options data item. When added to the metadata, this data item automatically includes the **StandardBPELActivityOptions** metamodel.

ExtensionActivityElement is used to add elements and attributes to the **ExtensionActivity** metadata. **From-Spec** and **To-Spec** are used in the **Assign** metamodel for the <to> and <from> elements. (See “From and To” on page 73 for more information on the the <from> and <to> elements.) **OpaqueActivity** is used in BPEL Abstract Processes. As with the Activity metamodels, **OpaqueActivity** can be selected at any Activity or BPEL Activity data item. See the BPEL specification for more information on [Abstract Processes](#).

BPEL Metadata

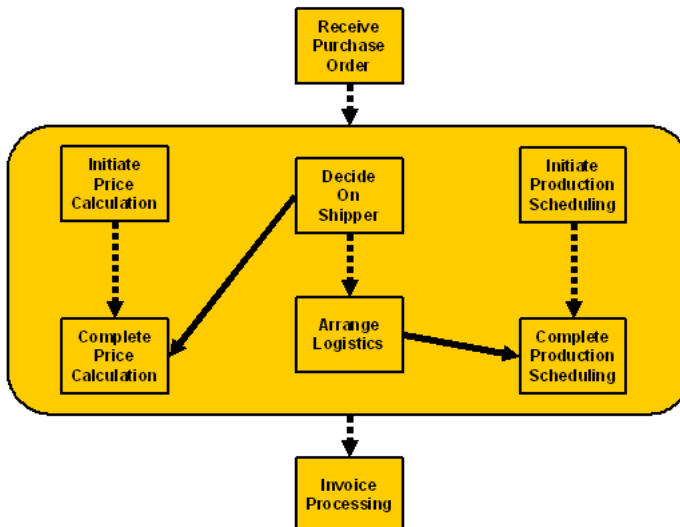
Metadata is considered to be BPEL metadata when BPEL metamodels are used. As with all metadata, SIMPROCESS only validates BPEL metadata to ensure it matches the data item type and any specified constraints on its value. Metadata entered is assumed to be consistent with the BPEL specification.

BPEL Export

Once BPEL metadata has been entered into a SIMPROCESS model, the **File/Export/BPEL Process** menu item can be used to export the BPEL metadata into one or more BPEL process files. Exporting requires BPEL metadata to be entered following a few guidelines, which will be described in this chapter. If export is not intended, then the guidelines can be ignored and BPEL metadata can be entered in any way that SIMPROCESS supports.

BPEL Example

The Web Services Business Process Execution Language 2.0 specification contains a [Purchase Order Example](#).

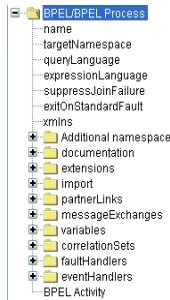


When a purchase order is received from a customer, three parallel paths are initiated: **Pricing**, **Shipping**, and **Scheduling**. There are dependencies between the three paths. The shipping price is needed from **Decide On Shipper** for **Complete Price Calculation** to finish. **Arrange Logistics** must provide the shipping date to **Complete Production Scheduling** before that activity can complete. When the three parallel paths are finished, **Invoice Processing** occurs, and the invoice is sent to the customer.

This example was used to create a SIMPROCESS demonstration model, **BPEL Purchase Order**, which is located in the `models/Demo` directory. This SIMPROCESS model models the purchase order example given in the BPEL specification and will be referenced throughout this chapter.

BPEL Process Metamodel

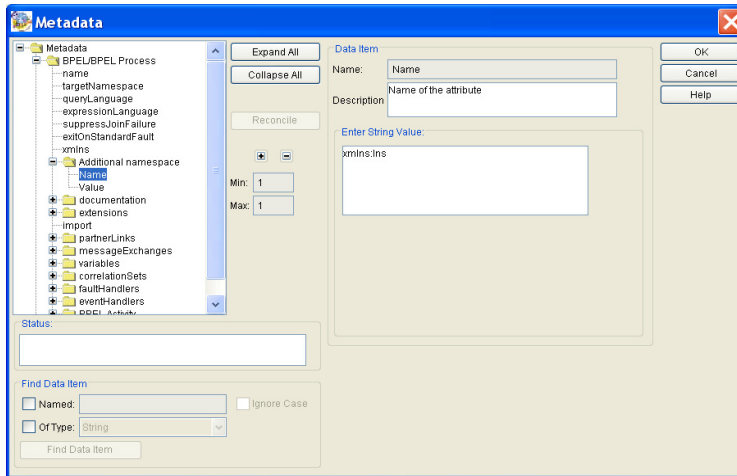
If BPEL metadata is to be exported, the **BPEL Process** metamodel must be assigned to a SIMPROCESS model. The **File/Export/BPEL Process** menu item is not active if **BPEL Process** is not an assigned metamodel. **BPEL Process** represents the `<process>` element with its attributes and all non-BPEL activity elements with their attributes.



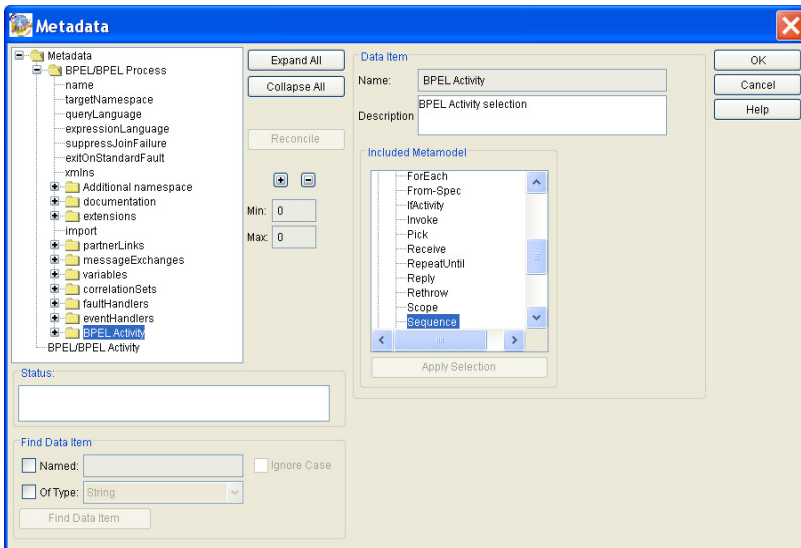
BPEL Process can be populated at any SIMPROCESS Activity; however, for the purposes of BPEL export, it should be populated at the model level (**Define/Metadata/Model Metadata...**) and/or at a hierarchical Process. Populating **BPEL Process** at the model level indicates that at least some of the SIMPROCESS model Processes and Activities represent BPEL activities. Populating **BPEL Process** at a hierarchical Process indicates that at least some of the Processes and Activities within the hierarchical Process represent BPEL activities. Thus, multiple BPEL processes can be represented in one SIMPROCESS model. All BPEL metadata found at level lower than a populated **BPEL Process** is considered to be a part of the BPEL export.

All of the data items except `Additional namespace` and `BPEL Activity` represent the actual BPEL elements or attributes. Instances of `Additional namespace` can be used to include other namespace information. It has two data items: `Name` and `Value`. For example, in

BPEL Purchase Order, an instance of `Additional` namespace was added to include another namespace.



The BPEL `Activity` data item is an `Include` data item. This data item allows the selection of a BPEL activity metamodel that will be the containing element for all BPEL activity metadata. In **BPEL Purchase Order**, the **Sequence** metamodel was included. Thus, all BPEL activity metadata found in the **BPEL Purchase Order** model will be contained within the `<sequence>` element. If an `Activity` metamodel is not included in the BPEL `Activity` data item, then the `<process>` element will be the parent element for the first BPEL activity metadata encountered.



The exported model-level **BPEL Process** metadata from **BPEL Purchase Order** is shown below. Note that `xmlns:lns="http://manufacturing.org/wsd1/purchase"` was created from the `Additional namespace` data item, and the `<sequence>` element was created from the included **Sequence** metamodel at the `BPEL Activity` data item. In order to just display exported BPEL information from a populated **BPEL Process** metamodel, `<sequence>` was collapsed to hide the BPEL activity elements.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <process name="purchaseOrderProcess" targetNamespace="http://example.com/ws-bp/purchase"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:Ins="http://manufacturing.org/wsd/purchase">
  <documentation xml:lang="EN">A simple example of a WS-BPEL process for handling a purchase
  order.</documentation>
- <partnerLinks>
  <partnerLink myRole="purchaseService" name="purchasing" partnerLinkType="Ins:purchasingLT" />
  <partnerLink myRole="invoiceRequester" name="invoicing" partnerLinkType="Ins:invoicingLT"
    partnerRole="InvoiceService" />
  <partnerLink myRole="shippingRequester" name="shipping" partnerLinkType="Ins:shippingLT"
    partnerRole="shippingService" />
  <partnerLink name="scheduling" partnerLinkType="Ins:schedulingLT" partnerRole="schedulingService" />
</partnerLinks>
- <variables>
  <variable messageType="Ins:POMessage" name="PO" />
  <variable messageType="Ins:InvMessage" name="Invoice" />
  <variable messageType="Ins:shippingRequestMessage" name="shippingRequest" />
  <variable messageType="Ins:shippingInfoMessage" name="shippingInfo" />
  <variable messageType="Ins:scheduleMessage" name="shippingSchedule" />
</variables>
- <faultHandlers>
  - <catch faultMessageType="Ins:orderFaultType" faultName="Ins:cannotCompleteOrder"
    faultVariable="POFault">
    <reply faultName="cannotCompleteOrder" operation="sendPurchaseOrder" partnerLink="purchasing"
      portType="Ins:purchaseOrderPT" variable="POFault" />
  </catch>
</faultHandlers>
+ <sequence>
</process>

```

BPEL Activity Metadata

BPEL Activity Metamodel

The BPEL Activity metamodel has two data items: `Export Order` and `Activity`.



`Export Order` is an `Integer` data item, and its use is optional. It is used to order the export of BPEL activity metadata on a particular model layout. If `Export Order` is not used, then the order of exported BPEL activity metadata cannot be guaranteed.

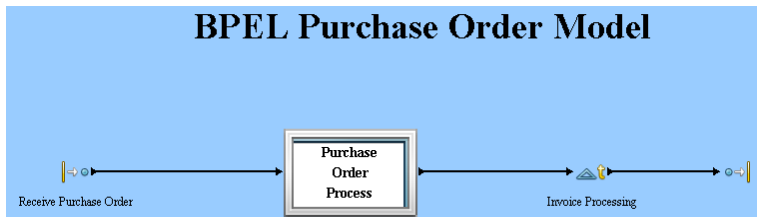
`Activity` is an `Include` data item. When selected, the tree of available metamodels displays. Since SIMPROCESS does not restrict which metamodels can be selected for an `Include` data item, any metamodel can be selected for inclusion. However, the intent of the `Activity` data item is that one of the 21 BPEL activity metamodels will be selected.

Entering BPEL Metadata

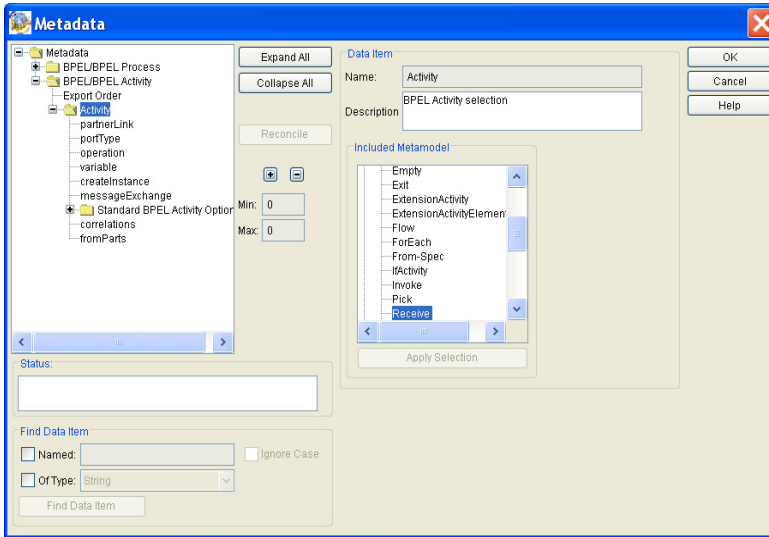
How BPEL activity metadata is entered into a SIMPROCESS model is highly dependent on how well the SIMPROCESS model represents the BPEL process since a SIMPROCESS model will most likely represent more than just Web Service interactions. During the model building phase, knowing that metadata from the SIMPROCESS model will be exported to a BPEL process file can help drive how the model is structured. The image below shows a portion of the BPEL process file for the purchase order example. The `<sequence>` element is a child of the `<process>` element.

```
- <sequence>
+ <receive createInstance="yes" operation="sendPurchaseOrder" partnerLink="purchasing"
  portType="Ins:purchaseOrderPT" variable="PO">
+ <flow>
+ <reply operation="sendPurchaseOrder" partnerLink="purchasing" portType="Ins:purchaseOrderPT"
  variable="Invoice">
</sequence>
```

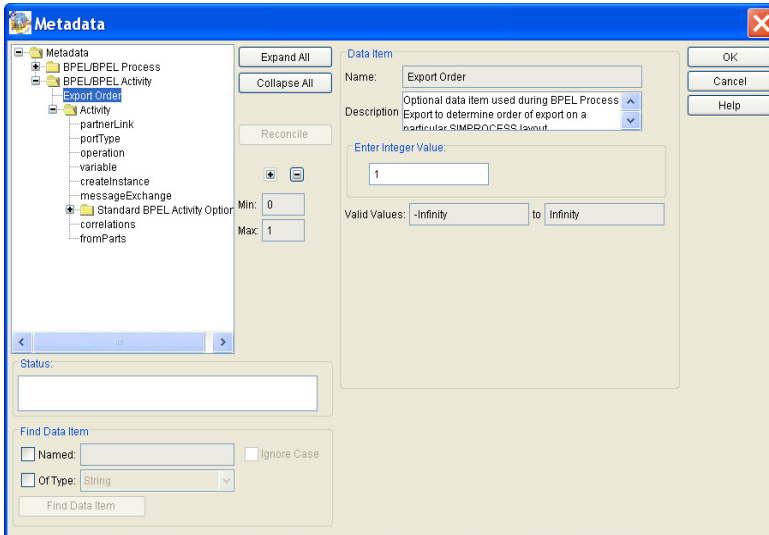
Note that there are three child elements: `<receive>`, `<flow>`, and `<reply>`. This indicates that in the SIMPROCESS model there should be three Activities or Processes at the top level.



In the image above, the Generate Activity represents the `<receive>` element, **Purchase Order Process** represents the `<flow>` element, and the Delay Activity represents the `<reply>` element. Editing the metadata of the Generate, **Receive Purchase Order**, shows the following.



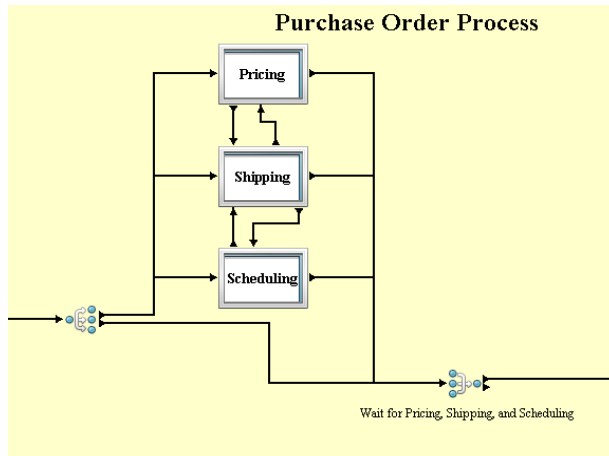
The **Receive** metamodel, which represents the BPEL <receive> activity element, was selected as the metamodel to include. The `Export Order` was set to 1 since this BPEL element is the first in the <sequence>.



`Export Order` was set to 2 for the **Purchase Order Process** metadata since it represents the <flow> element, and `Export Order` was set to 3 for the **Invoice Processing Delay Activity**

metadata since it represents the <reply> element.

Descending into the **Purchase Order Process** shows three Processes: **Pricing**, **Shipping**, and **Scheduling**.

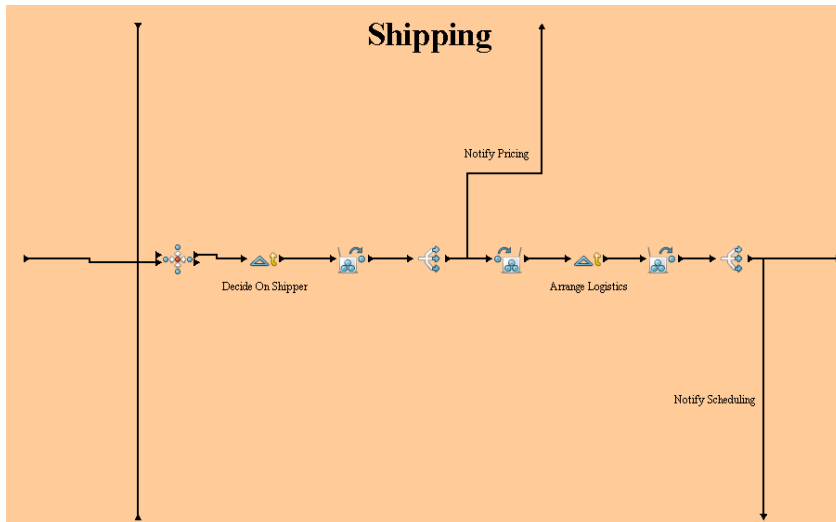


These three Processes equate to the three <sequence> elements within the **Purchase Order Process** <flow> element. Note that a Split Activity is used before the Processes since the events within each Process occur concurrently.

```
- <flow>
  <documentation>A parallel flow to handle shipping, invoicing and scheduling</documentation>
- <links>
  <link name="ship-to-invoice" />
  <link name="ship-to-scheduling" />
</links>
+ <sequence>
+ <sequence>
+ <sequence>
</flow>
```

The **BPEL Activity** metadata of each Process includes the **Sequence** metamodel for the Activity data item. **Shipping** has an Export Order of 1, **Pricing** has an Export Order of 2, and **Scheduling** has an Export Order of 3.

Descending into the **Shipping** Process shows the following.



The BPEL XML for Shipping is shown below.

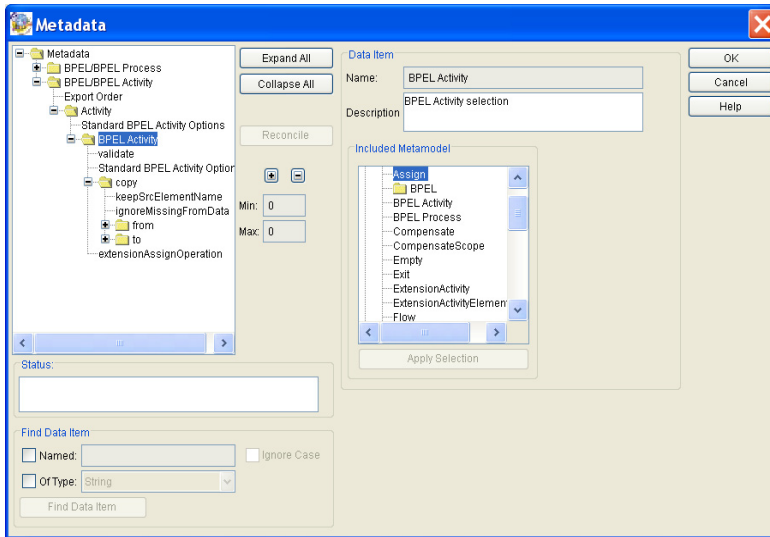
```

- <sequence>
- <assign>
  - <copy>
    <from>$PO.customerInfo</from>
    <to>$shippingRequest.customerInfo</to>
  </copy>
</assign>
- <invoke partnerLink="shipping" portType="Ins:shippingPT" operation="requestShipping"
  inputValue="shippingRequest" outputVariable="shippingInfo">
  <documentation>Decide On Shipper</documentation>
  - <sources>
    <source linkName="ship-to-invoice" />
  </sources>
</invoke>
- <receive partnerLink="shipping" portType="Ins:shippingCallbackPT" operation="sendSchedule"
  variable="shippingSchedule">
  <documentation>Arrange Logistics</documentation>
  - <sources>
    <source linkName="ship-to-scheduling" />
  </sources>
</receive>
</sequence>

```

There are three BPEL activity elements within the <sequence> element: <assign>, <invoke>, and <receive>. The BPEL metadata for the <invoke> activity is found in the **Decide On Shipper** Delay Activity, and the BPEL metadata for the <receive> activity is found in the **Arrange Logistics** Delay Activity. Again, Export Order was used to ensure that the <invoke> metadata was exported before the <receive> metadata. The BPEL metadata for the

<assign> activity is in the **BPEL Activity** metadata for **Shipping**.

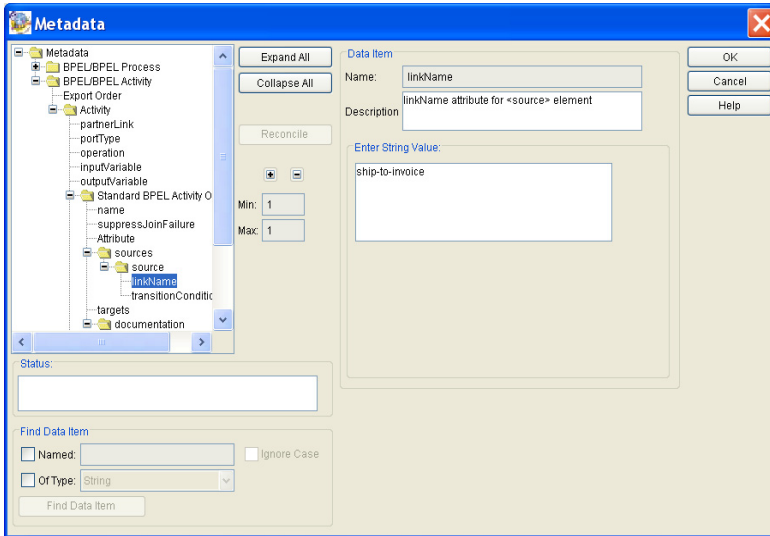


The **Activity** data item for **BPEL/BPEL Activity** has the **Sequence** metamodel included. The **Sequence** metamodel has a **BPEL Activity** data item (selected above). This is an Include data item with the **Assign** metamodel selected. Thus, during the BPEL export the **Assign** metamodel’s metadata will be exported before the BPEL metadata in any Activities or Processes within the hierarchical Process (in this case **Shipping**).

Standard BPEL Activity Options

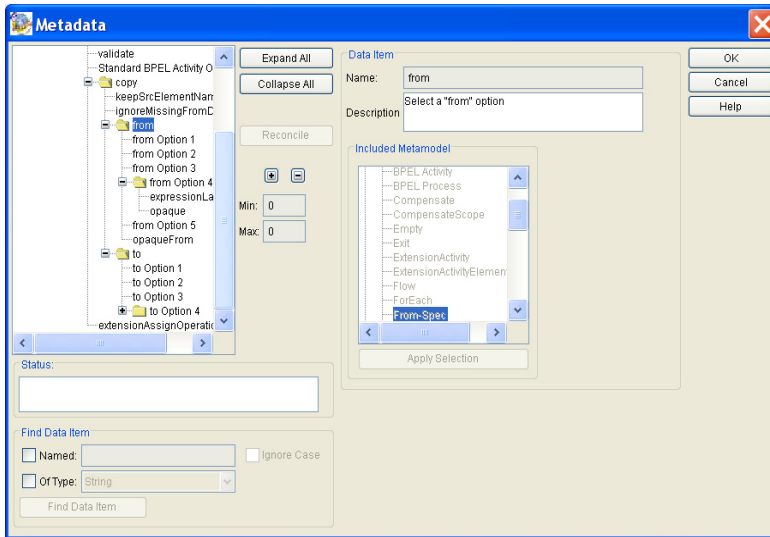
Each BPEL activity metamodel contains a data item named **Standard BPEL Activity Options**. This is an Include data item that is preset to include the **StandardBPELActivityOptions** metamodel. Every BPEL activity can have certain attributes and elements. The common attributes are name and `suppressJoinFailure`, and the common elements are <targets> and <sources>. See the [BPEL specification](#) for a discussion of these. Also included in the **StandardBPELActivityOptions** metamodel is a data item for additional attributes (**Attribute**) and a data item for the <documentation> element.

The metadata for the **Decide On Shipper Delay Activity** in the **Shipping** Process contains metadata for <sources> and <documentation>. Shown below is the `linkName` attribute for the <source> element.



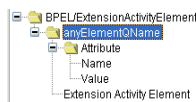
From and To

The BPEL <assign> activity can contain a <copy> element. The <copy> can have a <from> and/or a <to> element. The **Assign** metamodel has a **copy** data item that has subordinate **from** and **to** data items. The **from** data item automatically includes the **From-Spec** metamodel, and the **to** data item automatically includes the **To-Spec** metamodel. The **From-Spec** metamodel has five options for defining a <from> element and an option for an <opaqueFrom> element. The **To-Spec** metamodel has four options for defining a <to> element. Only one option for each should be populated in the metadata.

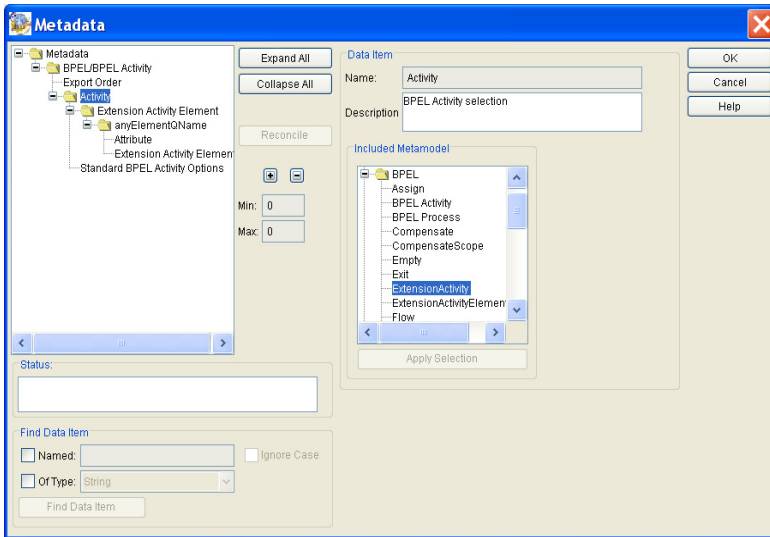


Extension Activity

The `<extensionActivity>` element allows the inclusion of activities that are not defined by the BPEL specification. These can be included in the metadata by using the **ExtensionActivity** metamodel. This metamodel contains two data items, `ExtensionActivityElement` and `Standard BPEL Activity Options`. `ExtensionActivityElement` is an Include data item that automatically includes the **ExtensionActivityElement** metamodel. There is one data item, `anyElementQName`, that has two child data items, `Attribute` and `ExtensionActivityElement`. The `anyElementQName` data item is a `String` data item. The value should be the tag name of the activity being defined. The `Attribute` data item is used to define the attributes for the element specified in `anyElementQName`. Multiple instances of `Attribute` can be added. `Name` is for the name of the attribute, and `Value` is for the attribute value. The `ExtensionActivityElement` data item allows the addition of child elements. It adds the same data item structure.



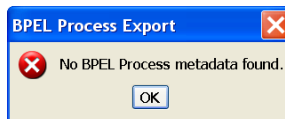
Selecting the **ExtensionActivity** metamodel at an Activity or BPEL Activity data item displays the following.



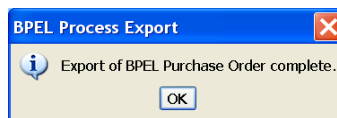
BPEL Export

Export Procedure

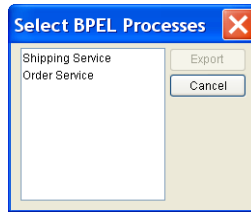
Selecting **File/Export/BPEL Process** starts the BPEL export procedure. The procedure first checks to determine that metadata has been entered for the BPEL Process metamodel in at least one location. If none is found, an error message will appear, and the export will stop.



If only one set of **BPEL Process** metadata is found, the export continues with no user interaction. A completion notice appears when the export finishes.



When there is more than one set of **BPEL Process** metadata, a dialog will appear that lists the locations where the metadata was found. If **BPEL Process** metadata is at the model level, then the name of the SIMPROCESS model appears in the dialog. If the metadata entry occurred at a hierarchical Process, the name of the Process appears in the dialog. In the example below **BPEL Process** was populated at two hierarchical Processes, **Shipping Service** and **Order Service**.



Selecting an item for export causes the **Export** button to enable. Multiple items can be selected. When **Export** is clicked, each selected item will be exported in turn. A completion dialog will appear for each. The exported files are created in the SIMPROCESS model's directory. Each file name is the name of the selected item with `.xml` appended.

If errors occur during export, a notification dialog will appear, and the export will stop. Note that if multiple exports were selected, an error on one export will not stop the other selected exports. When errors occur, the `simprocess.err` file in the `SPSYSTEM` directory may contain helpful information. Contact the SIMPROCESS Help Desk if export errors persist.



Export Considerations

The BPEL export file created from exporting the BPEL metadata in the **BPEL Purchase Order** model, matches exactly the purchase order example in the BPEL specification. This is because the **BPEL Purchase Order** model was constructed specifically for this example. The BPEL export from other SIMPROCESS models may not be complete. How complete the export is will depend upon the structure of the SIMPROCESS model and how the BPEL metadata was entered. Also, SIMPROCESS does not perform any type of validation on the BPEL metadata. It simply exports the metadata it finds. The user must ensure that the exported BPEL file is consistent with the BPEL specification.