

OpenMI Editor

Comparisons between versions 1 and 2

Table of Contents

OpenMI Editor.....	1
Comparisons between versions 1 and 2.....	1
Overview.....	1
Model Add and Remove.....	2
Model Trigger.....	2
Model Properties.....	2
Connection Add and Remove.....	3
Link Editing.....	3
Creating a decorated output exchange item.....	4
Editing arguments on a decorator	6
Runs.....	7
OPR File.....	9

Overview

The “OpenMI Association” is responsible for the standard. However, it also commits to providing an open source editor that is easily obtainable. It is not intended that this editor be “complex” or “advanced”, its primary goals are

- to provide a simple demonstration of OpenMI functionality
- allow a basic means of connection and running OpenMI components interactively
- minimize complexity of maintenance e.g. make bug fixing as simple as possible by anyone

The association assumes, and indeed actively supports, the development of 3rd party editors that provide advanced or more specialized functionality.

However, whenever a new version of the standard is released (or proposed) then this editor is the simplest way of demonstrating and providing examples of any new features. This document shows the proposed changes to the editor required to support changes made between versions 1 and 2 of the OpenMI standard.

The figures in this document are taken from screen shots of the editor under development, the full open source under development is available

- <https://openmi.svn.sourceforge.net/svnroot/openmi/trunk/Oatc/src/csharp/Gui>

and the latest version of this document can be obtained from

- <https://openmi.svn.sourceforge.net/svnroot/openmi/trunk/doc/EditorComparisonV1V2.odt>

and as a PDF

- <https://openmi.svn.sourceforge.net/svnroot/openmi/trunk/doc/EditorComparisonV1V2.pdf>

This updated document will eventually become a subsection of the new UI Help located in

- <https://openmi.svn.sourceforge.net/svnroot/openmi/trunk/Oatc/src/csharp/Gui/Help>

HEALTH WARNING

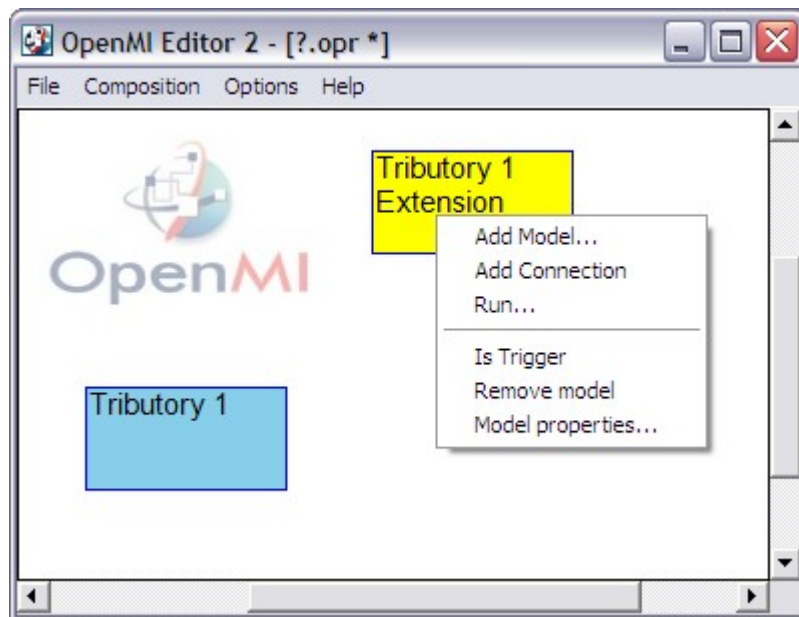
The UI is under development, figures and design will change!

Model Add and Remove

The mechanism for adding and removing models from *.omi files is unchanged between versions.

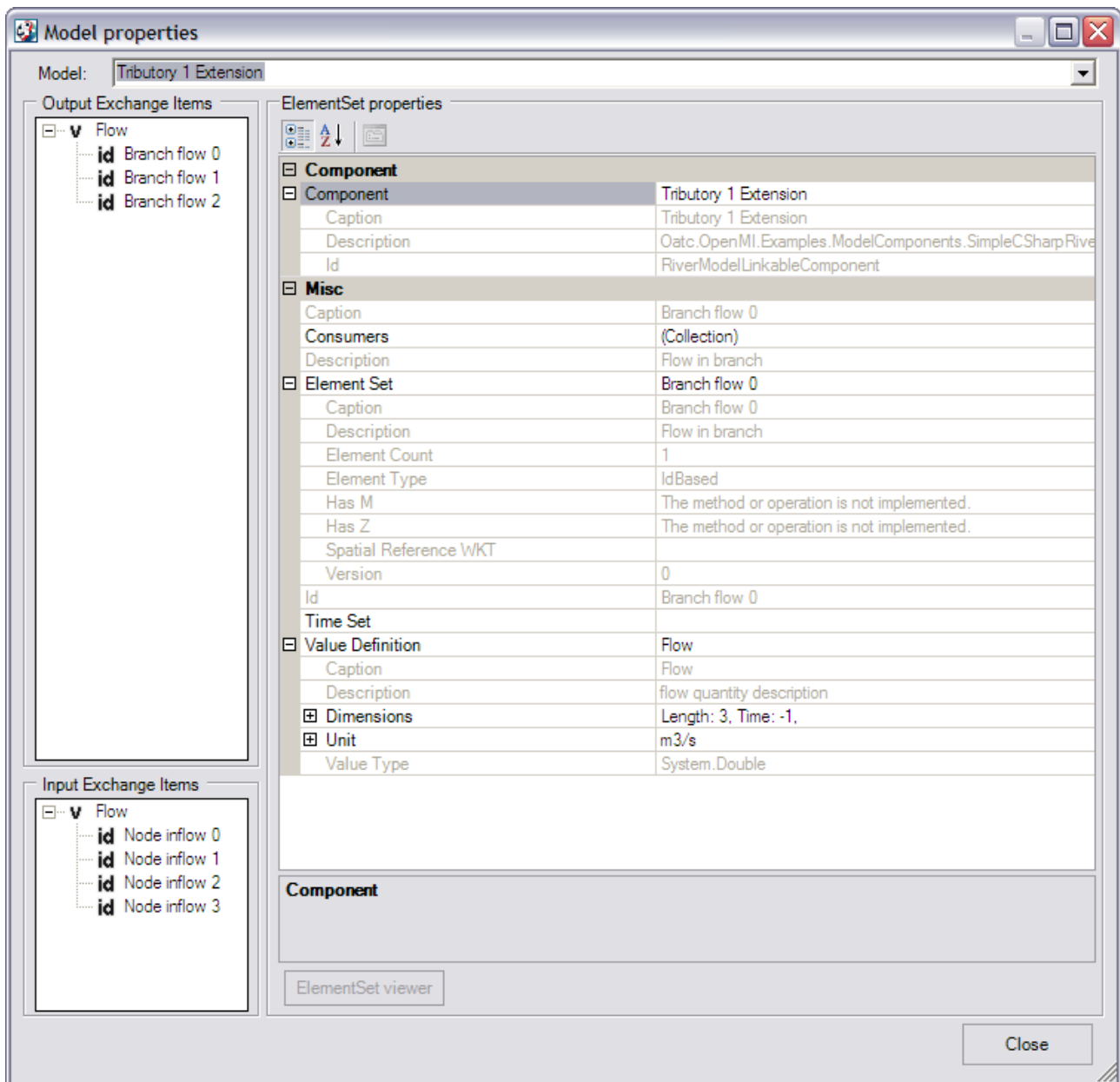
Model Trigger

- The trigger box in version 1 no longer exists, version 2 allows the trigger to be applied directly to a model.
- The model that is also the trigger for the composition appears as a blue box. Only one trigger is allowed in an composition so all other added models will appear as yellow.
- The first added model will have a trigger added to it automatically.
- To move a trigger to another model
 - Select the model with the cursor
 - Right click mouse to obtain 'context menu'
 - Select “Is Trigger”
 - Model should change colour from yellow to blue



Model Properties

The 'context menu' for the model, see previous figure, also has the option 'Model Properties ...'. Choosing this will produce a new menu which shows information about the OpenMI implementation for that model.



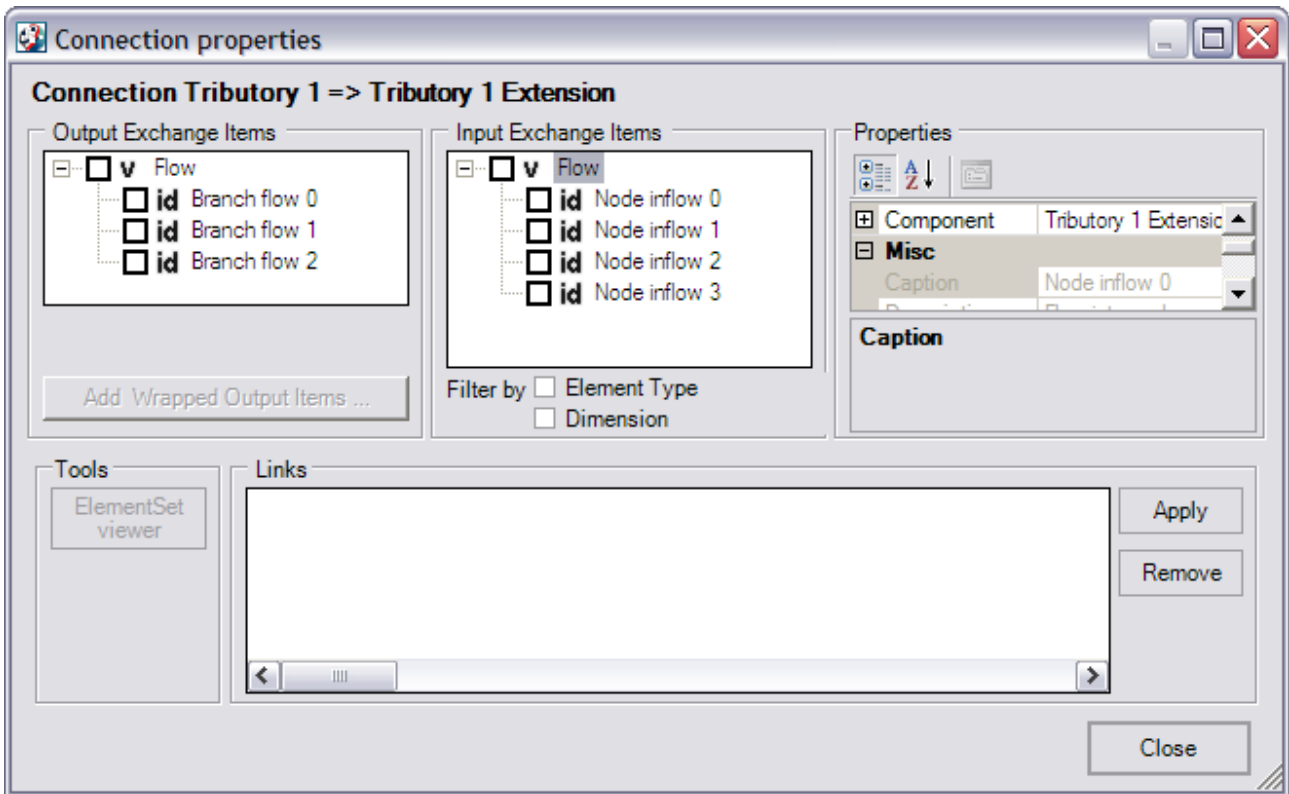
This dialog is the same between version 1 and 2 and operates in the same way. However, the information displayed in the right hand side property pane is changed to version 2 values. See help for a description of these values.

Connection Add and Remove

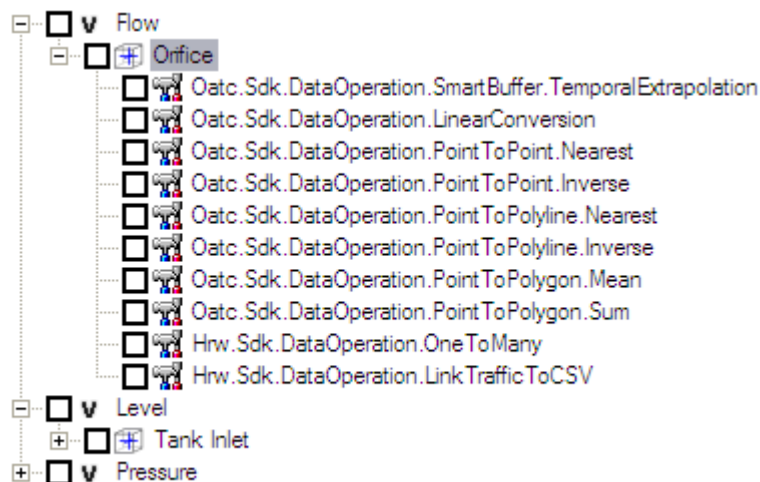
The mechanism for adding and removing models from connections is unchanged between versions.

Link Editing

The link editor dialog has changed



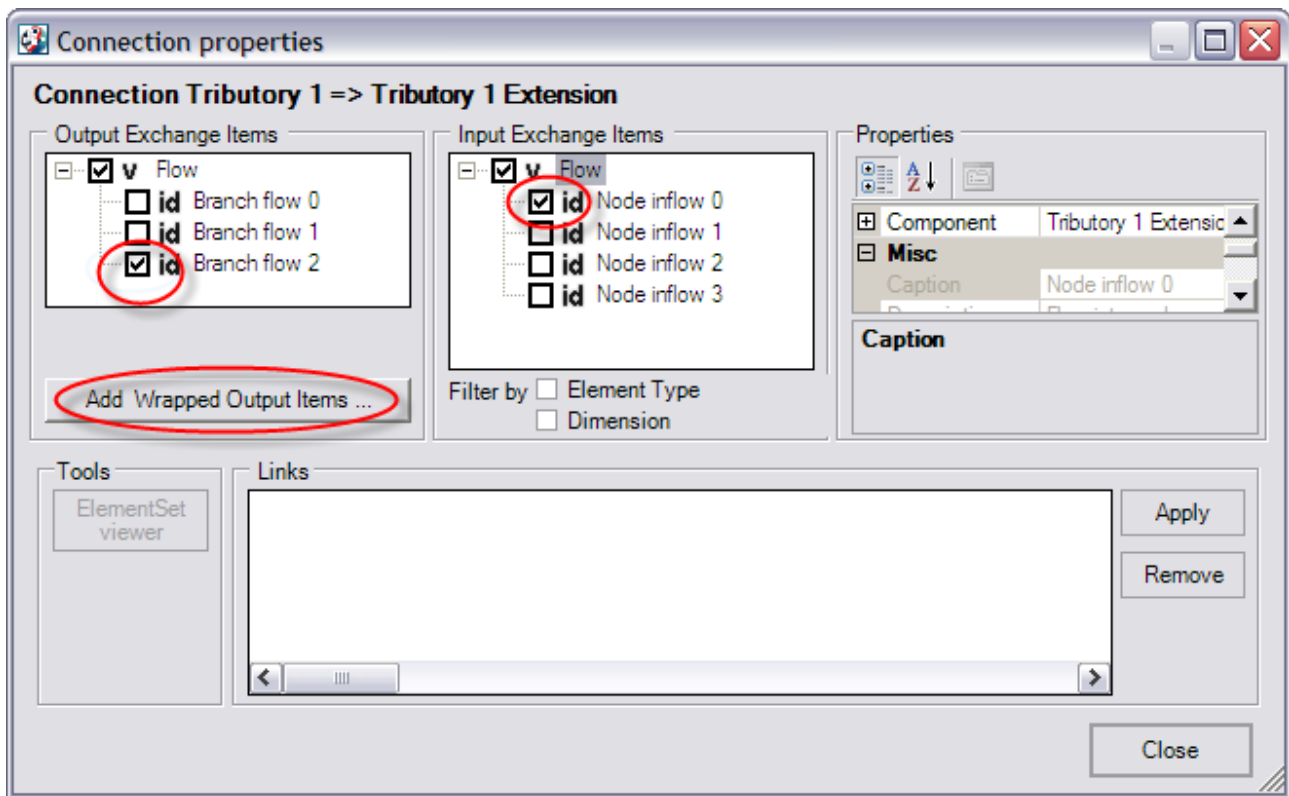
The most significant change is in the “Output Exchange Items” panel. In version 1 it looked like this ...



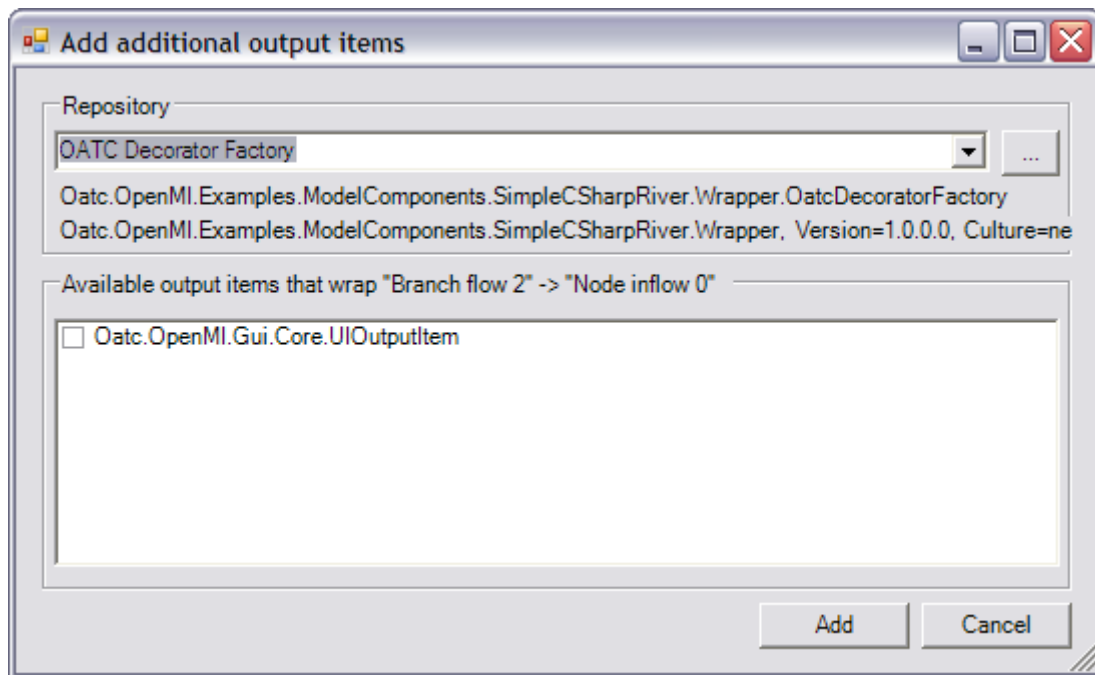
So we can see that there is only levels in version 2, the third level 'Data Operations' have been removed. In version 2 we build data operations by decorating output exchange items. This new approach is described below.

Creating a decorated output exchange item

First select both an output and input exchange item pair. When both items have been selected the button “Add Wrapped Output Items” will be enabled.

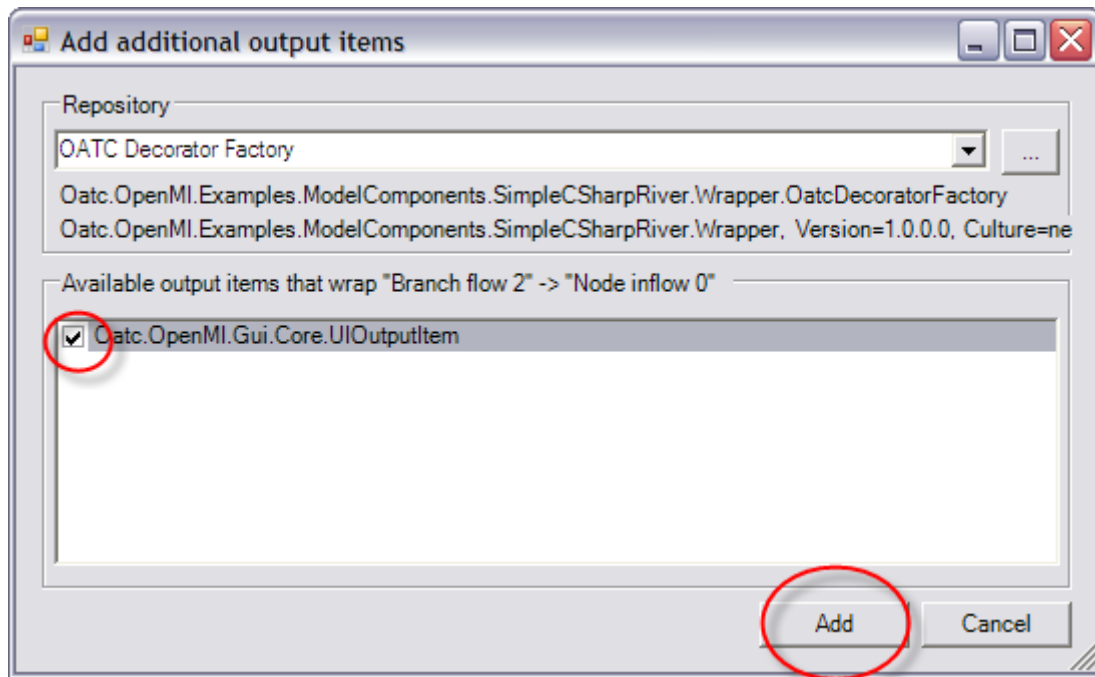


Click on that button, a new dialog appears ...

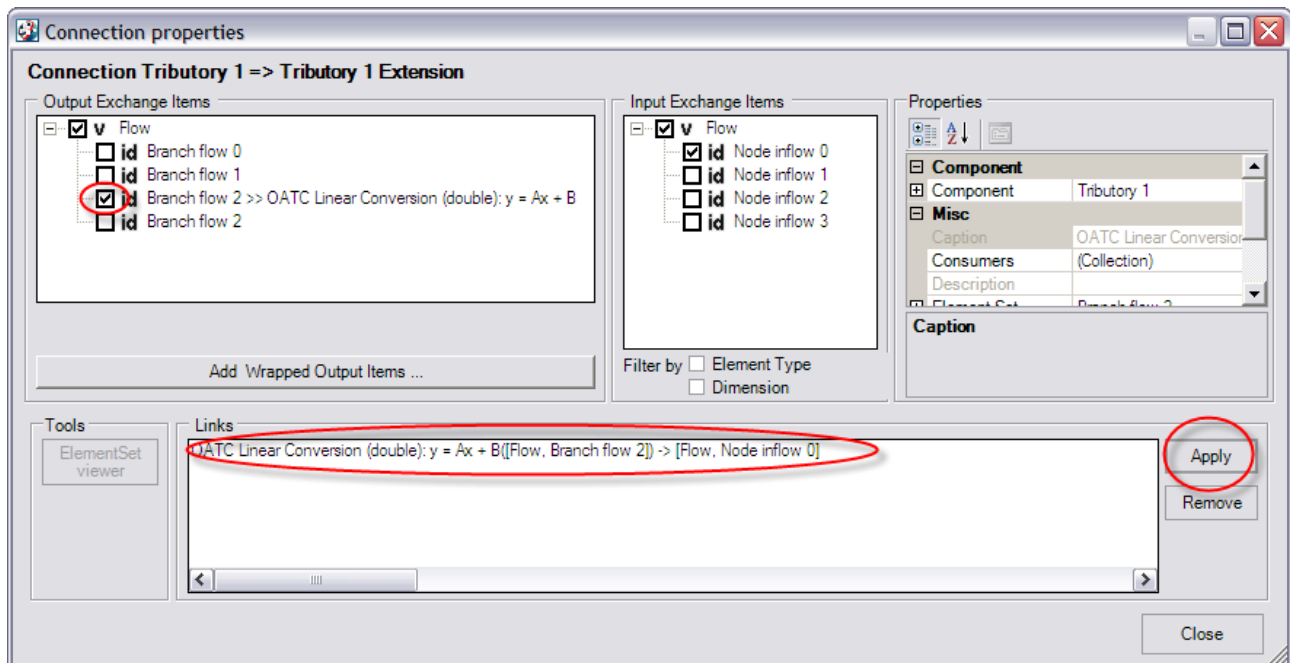


Version 2 allows decorated exchange items to be created from factories. These factories can be found from the source component or from 3rd party code libraries. By default, the factory from the source component is loaded. This can be seen in the above figure in the “Repository” panel. In the example shown above, the factory is called “OATC Decorator Factory” and the decorators available in that factory are listed in the lower panel, in this simple example, there is only one decorator “Oatc.OpenMI.Gui.Core.UIOutputItem”. If you wish to create a decorator from a 3rd party library, click on [...] button and browse for a *.dll files that implement the OpenMI version 2 [IexchangeItemDecoratorFactory](#) interface.

Select a decorator and Add ...



Which returns to the link editor



A new output exchange item will have been added. If you select this (along with the already selected input exchange item) you can click on apply which then adds the new link to the Links Panel.

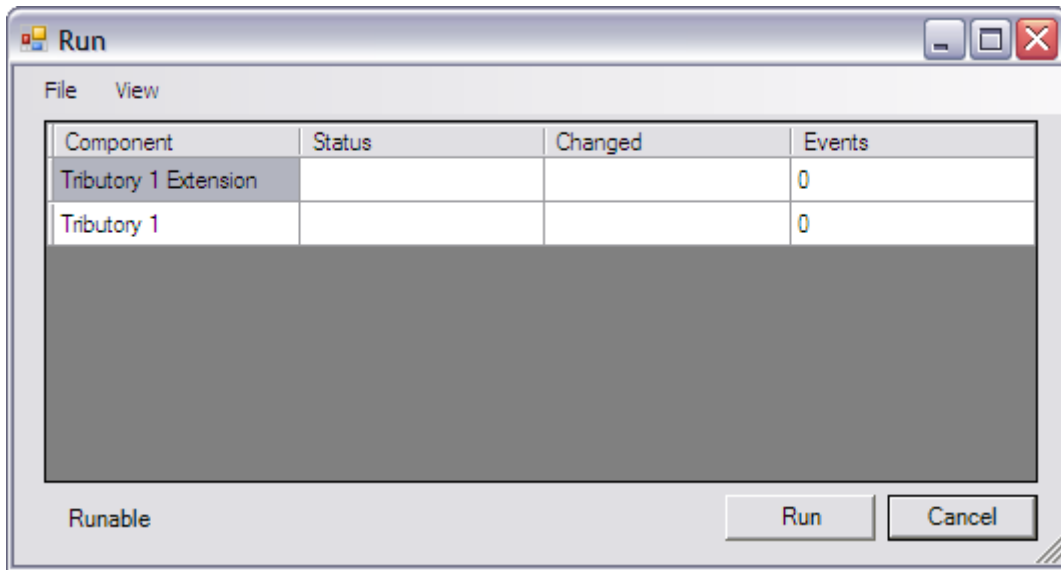
This process can be repeated as many times as necessary to create the required linkages.

Editing arguments on a decorator

TODO

Runs

The UI for running models has changed between versions 1 and 2. The new run dialog looks like

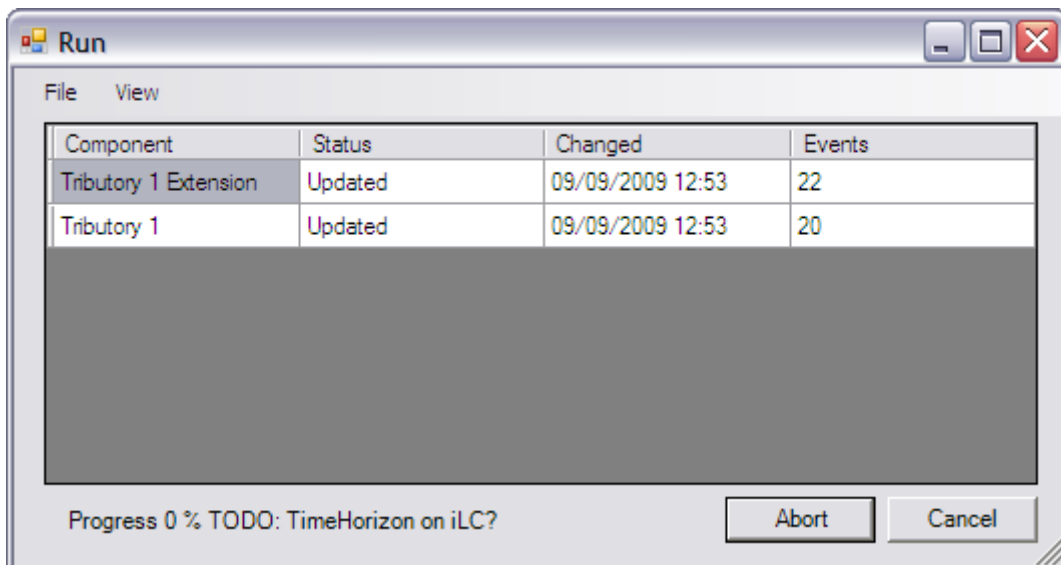


The table will have a row for every model in the composition.

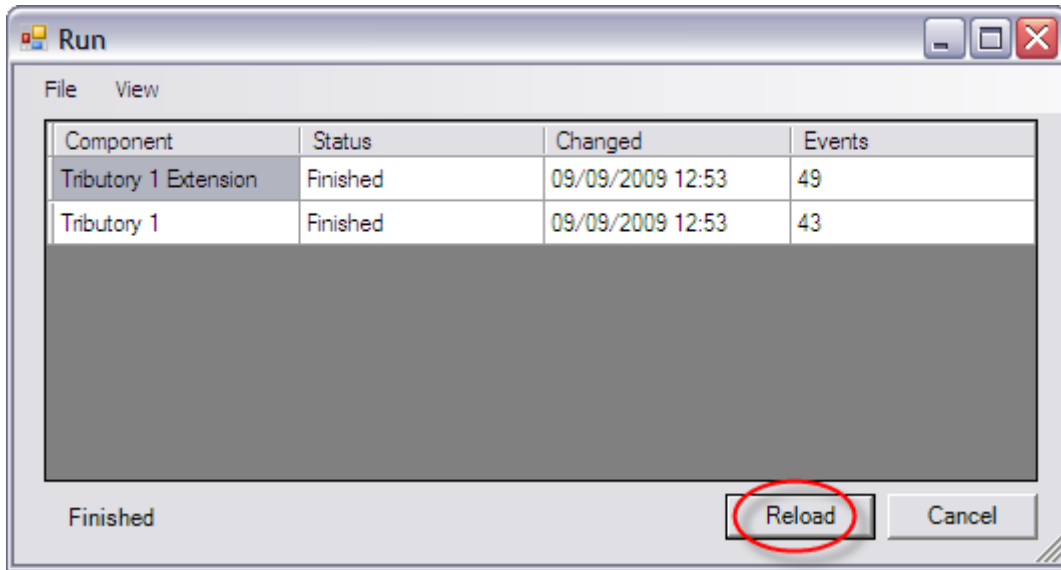
The columns in the table

- Component
 - The caption of the component
- Status
 - The state that the component is in e.g. initialised, waitingForData, Finished etc
- Changed
 - The time when the status last changed
- Events
 - The number of Status/Exchange events that have been intercepted

During the run ...



and on completion ...

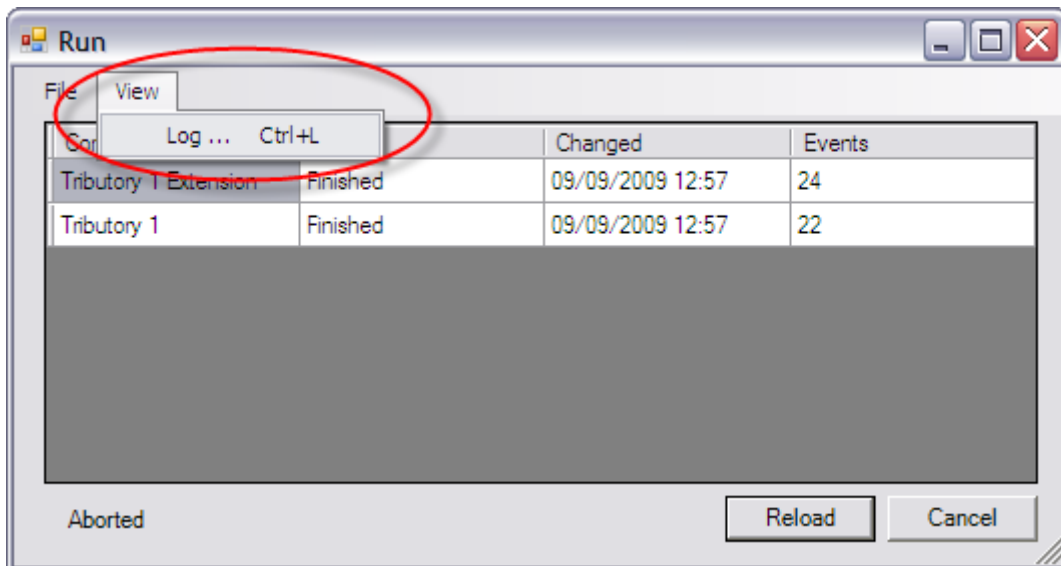


Notice that the highlighted button in the last figure changes its operation between Run/Abort and Reload.

It is likely that addition changes to this dialog will be made to

- enable disabling of events (for efficiency reasons)
- pause/step/restart for debugging

After a run, you can view the log via the view menu



which will display something similar to the following

DateTime	Component	Details
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: Initialized => WaitingForData
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: WaitingForData => Updating
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 0, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 1, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 2, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: Updating => Updated
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: Updated => WaitingForData
09/09/2009 12:57	0: Tributary 1 Extension	Target: Node inflow 0, 53286.6934259259 = 0
09/09/2009 12:57	1: Tributary 1	Component Status: Initialized => WaitingForData
09/09/2009 12:57	1: Tributary 1	Component Status: WaitingForData => Updating
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 0, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 1, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 2, Cache[1] = (53286.6934259259 .. 53286.6934259259)
09/09/2009 12:57	1: Tributary 1	Component Status: Updating => Updated
09/09/2009 12:57	1: Tributary 1	Component Status: Updated => WaitingForData
09/09/2009 12:57	1: Tributary 1	Component Status: WaitingForData => Updating
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 0, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 1, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 2, Cache[2] = (53286.6934259259 .. 53287.6934259259)
09/09/2009 12:57	1: Tributary 1	Component Status: Updating => Updated
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 2, Cache[2] = (53286.6934259259 .. 53287.6934259259)
09/09/2009 12:57	0: Tributary 1 Extension	Target: Node inflow 0, 53286.6934259259 = 3.6
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: WaitingForData => Updating
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 0, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 1, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Source: Branch flow 2, Cache[0] = (...)
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: Updating => Updated
09/09/2009 12:57	0: Tributary 1 Extension	Component Status: Updated => WaitingForData
09/09/2009 12:57	0: Tributary 1 Extension	Target: Node inflow 0, 53287.6934259259 = 3.6
09/09/2009 12:57	1: Tributary 1	Component Status: Updated => WaitingForData
09/09/2009 12:57	1: Tributary 1	Component Status: WaitingForData => Updating
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 0, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 1, Cache[0] = (...)
09/09/2009 12:57	1: Tributary 1	Source: Branch flow 2, Cache[2] = (53287.6934259259 .. 53288.6934259259)

Currently you cannot save this log, but you can cut and paste via the clipboard.

OPR File

The format has changed between versions. The UI will provide a version 1 opr import function and some means of migrating version 1 models to version 2. TODO