

# Using the configuration editor

## 1. Introduction

The OpenMI standard provides the capability for linking different computational models, mostly used for water and environment modeling. After this standard was introduced, the need arose for a user-friendly application to link OpenMI-compliant models. The Configuration Editor performs this task.

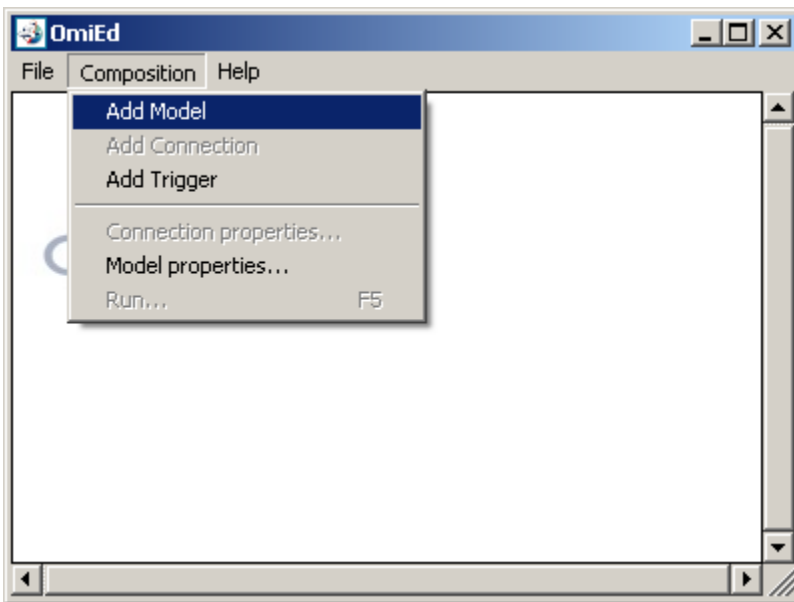
Note that the Configuration Editor in previous releases was called OmiED. You may still find this name in figures and documentation files.

You can download the configuration editor [here](#)

## 2. Working with the Configuration Editor

### 2.1 Adding models to a composition

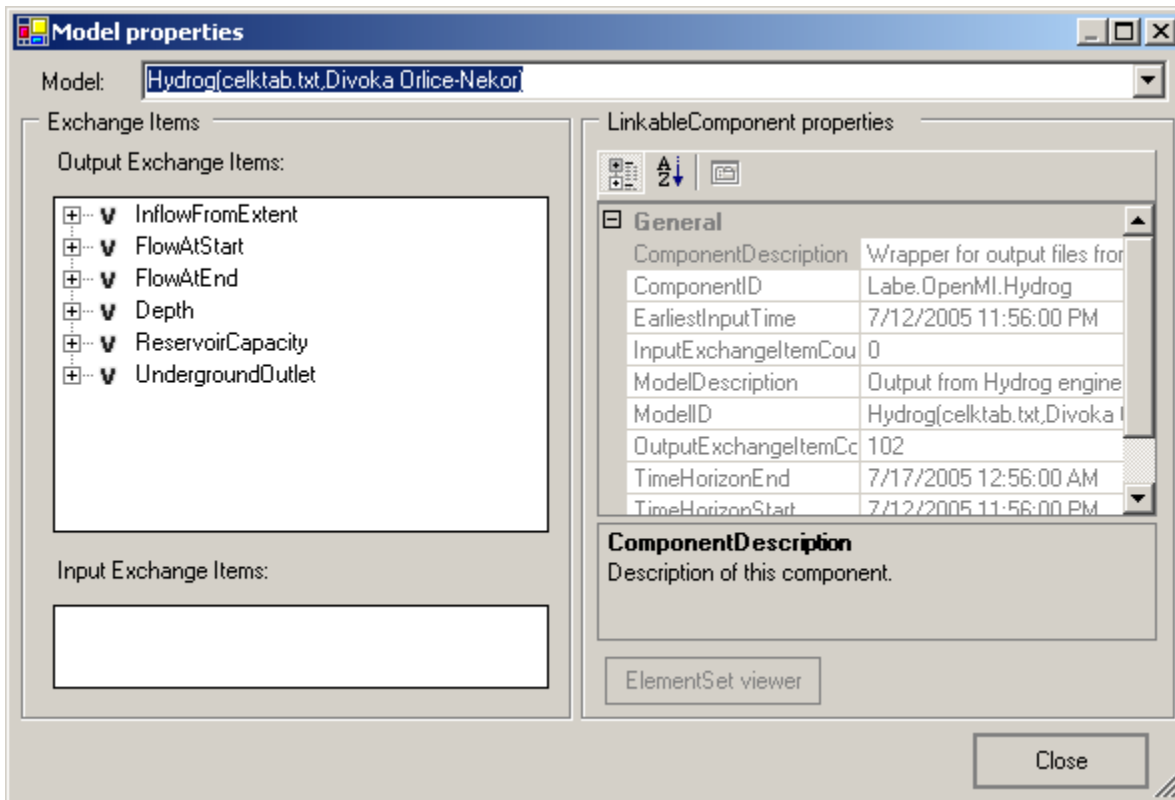
The first step in linking models together is to add the models to a *composition*. From the menu bar, select Composition > Add Model. In the Open dialog, choose the OMI file corresponding to the first model and click on OK. A rectangle representing the model appears in the composition area. (Note that the same operation can be performed using a pop-up menu, which is opened by right-clicking.)



Repeat this procedure for each model that you want to include in the composition. The only limitation is that two models cannot have the same ModelIID.

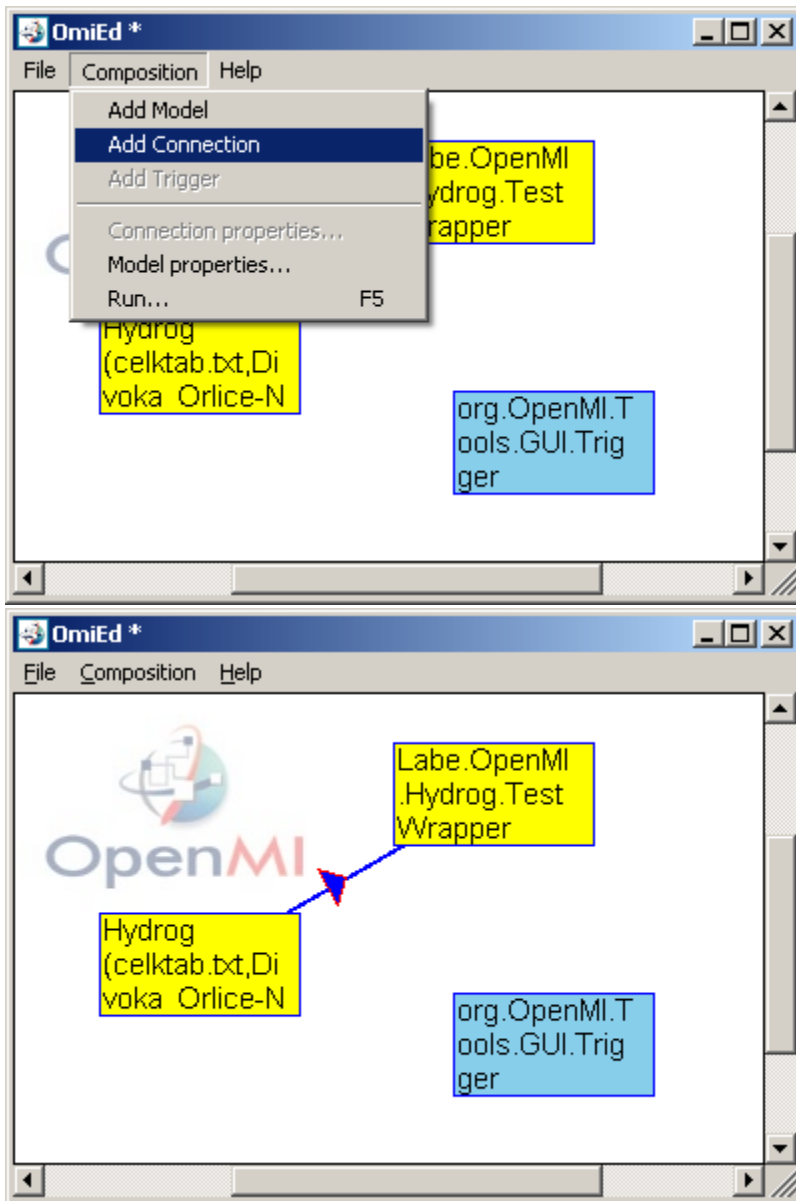
You should also add a Trigger if you want to run a simulation of the composition. Select Composition > Add Trigger. The Trigger is a special model used to start the simulation; see *Running a simulation* for more detail.

To show the properties of the model, select Composition > Model properties and choose the required model. Alternatively, right-click on the model rectangle and select Model properties.



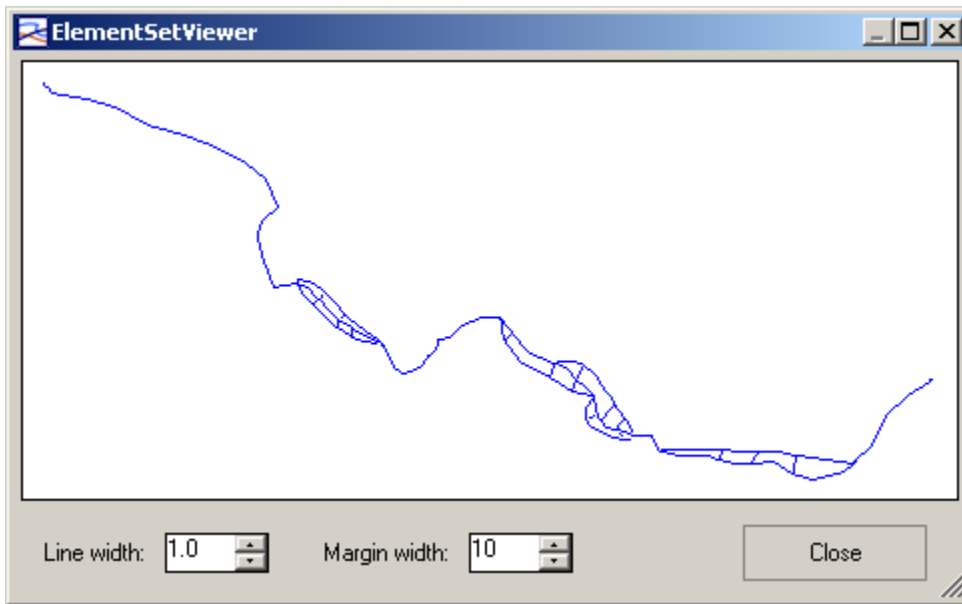
## 2.2 Linking models

To link two models, you must first create a connection between them. A connection collects the links between two models where all its links have the same direction as the connection itself. From the menu select Composition > Add Connection or on the pop-up menu click Add Connection. Click on the source model's rectangle and then on the target model's rectangle. The source model is the model providing data, the target model is the model accepting data. A line now connects the two models, with a small triangle indicating the direction of the link.



If you select the Use Dimension filter, only input exchange items are shown that have the same dimension as the quantity checked in the output exchange items. The Use ElementType filter is similar but only input exchange items are shown that have either an ID-based element set or an element set of the same type (e.g. line, polygon, poly-line) and with the same number of elements as the checked output element set.

The ElementSet Viewer is a simple tool used to show spatial properties of one or two ElementSets. Check the Output and/or Input ElementSet and click on the corresponding button.



## 2.3 Running a simulation

Before running a simulation, make sure that the Trigger is part of your composition and is linked to one model. The Trigger is a special model used to fire the simulation; its only action is to invoke `GetValues()` on the linked model at the specified time.

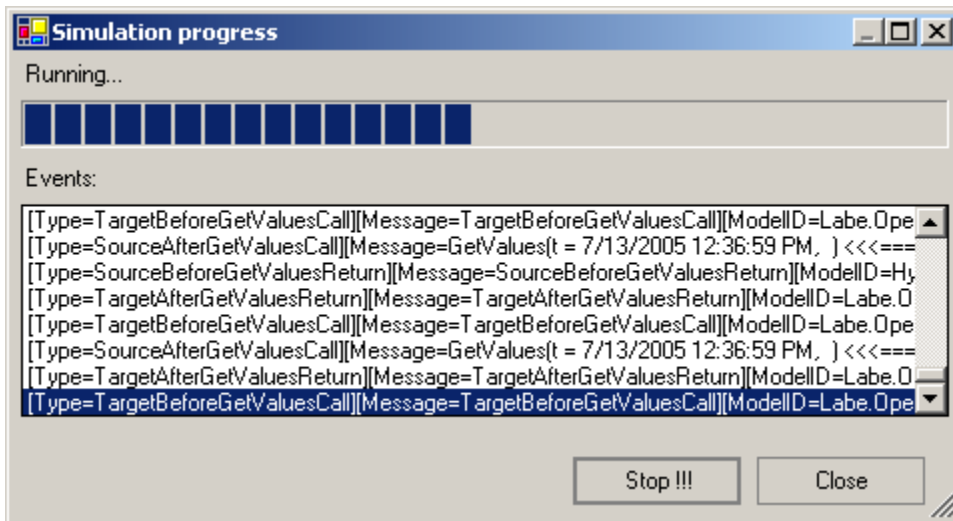
From the menu, select Composition > Run to show the Run Properties dialog.

Here you can set the types of event you want to list during the computation. These events can be stored in a log file (check *Log to file* and type a filename or click on the right-hand button to browse for a file) and/or shown in a list-box (check *Show events in listbox*). Remember that the handling of events can cause large performance overheads for models performing fast timesteps showing events in a list-box will make the simulation particularly slow.

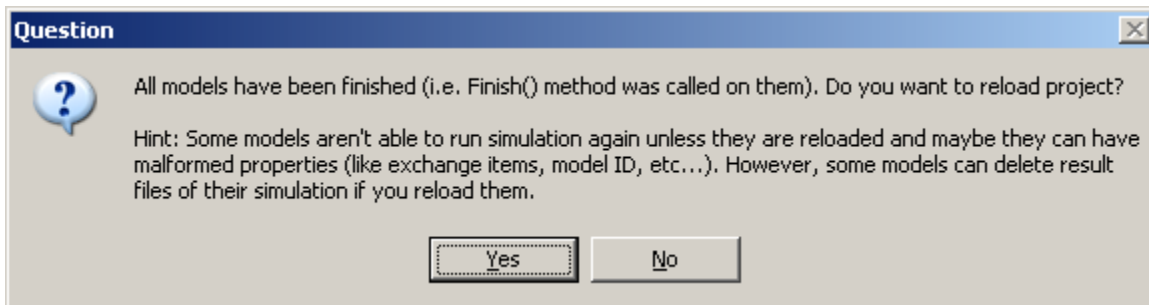
The *Invoke trigger at* entry defines the time when the Trigger invokes `GetValues()` on the linked model. It can be set to the latest overlapping time of all the models in the simulation by clicking the *Latest overlapping* button.

The *Don't use separate thread* check-box determines whether the simulation is run in the same thread as the user interface (UI). If the simulation runs in a separate thread, the UI responds immediately and the user can even stop it. However, some components wrapping - for example a model using COM single-thread apartments - can cause problems, because the model (in this case the COM object) is initialized in the UI thread and not in the simulation thread. If the simulation runs in the same thread as the user interface, and the timestep of a model takes a long time to execute, the UI can stop responding and the user isn't able to stop the simulation.

To start the simulation, click RUN.



After the simulation has finished, a message box appears asking you whether you want to reload the project. The OpenMI standard doesn't include functionality to reinitialize models after a simulation has finished, and the model may not be able to run the simulation again and may have malformed properties (exchange items, model ID etc.) Reloading just works around this behavior, and has the same effect as saving the project, restarting OmiEd and opening the project again. However, some models may delete the result files from their simulation if you reload them.



If you decide not to reload the composition, you can do it later using File > Reload.