**Model Conversion Directives**

Version: PreRelesase

Last Revised: March 29, 2016

Author: Patrick Wingo

In order to augment System Dynamics models during the conversion process, a simple command syntax has been implemented. The modification directives are intended to be implemented on a per-model component basis, with the actual commands being placed in a field that has no effect on the model execution within its native environment. The upshot to this approach is that an originating model can be modified to make the appropriate changes during the OME conversion process while still being able to function normally within its native environment.

Presently, conversion tools only exist for models originating from Simile. Each Simile model component has a “comments” field which is reserved for free text to be used for any further annotation required by the author. It is within this field that conversion directives can be defined and are extracted from during a model's conversion from Simile's native format to OME's intermediate XML format.

To denote the beginning of OME conversion directives within a free text field, the string (excluding the quotes) “--!!OME:” is used; the ending of the block (again excluding the quotes) is denoted by the string “!!--”. Each directive within the directive block is prefixed with an at ('@') symbol. Some directives require an argument; in this case, a colon (':') separates the command from the argument.

Here are a pair of examples of OME conversion directive blocks:

**Omit an object:**

*--!!OME:@omit!!--*

**Mark an object as init only and change the update expression to read time (the object will now record the start time):**

*--!!OME:@initOnly@expression:time()!!--*

The following are the directives that will be processed by the conversion tool (text enclosed by < and > denote arguments):

**execgroup:<group>** – Assign a model component to a specific execution group with the number provided in place of **<group>**. An execution group waits for all instances of a lower-number execution group to complete before its own contents are executed.

**expectsSpatial** – Indicates that a submodel is expected to have its number of initial instances set by a SDP at runtime.

**expression:<newExpr>** - Override an object’s expression with the one supplied in the **<newExpr>** argument. This is useful for patching holes from the **omit** expression and adding Spatial Data Provider commands to the converted model.

**influenceFrom:<objName>** - Create a new influence from the model component specified by **<objName>** to the current component. Can be used to patch holes in the model created by the **omit** command.

**influenceTo:<objName>** - Create a new influence from the current model component, to the component identified by **<objName>**. Can be used to patch holes in the model created by the **omit** command.

**initOnly** - Explicitly marks an object to only be evaluated once at the beginning of a simulation. This can be useful for optimization reasons.

**move:<mdlName>** - Move the model component to the model referred to by the name supplied by the **<mdlName>** argument. This will break any influences to/from the object. Influences can be restored using the **influenceTo:** and **influenceFrom:** directives.

**omit** - Removes the object from the model and any associated influences. If **omit** is applied to a submodel, all objects contained within a submodel are removed as well. Removing a submodel can be useful particularly when a spatial coverage is replacing association-linked models used for determining neighbors.