Overview of connection rules

Connection rules handle the automatic parameter value propagation for connected layout components. These connection rules were previously known as “machine kits”, due to the use of iLogic rules in their implementation in the 2011 release. The following workflows are considered with connection rules:

1) When a system asset containing connection rules is republished as a user asset, the connection rules will continue to work without any additional user intervention.
2) When creating new assets, it’s possible to have them interact – in terms of parameter propagation – with other similar assets by assigning “connector class properties”. These properties control which parameters should be referenced to achieve the desired connection behavior. More on this below.
3) Advanced users may define custom connector classes. These connector classes identify the types of connection-time parameter propagation that is supposed to happen, and can specify filters, based on other parameters, to control how and when the parameter propagation is to occur. This aspect of functionality is beyond the scope of this document.

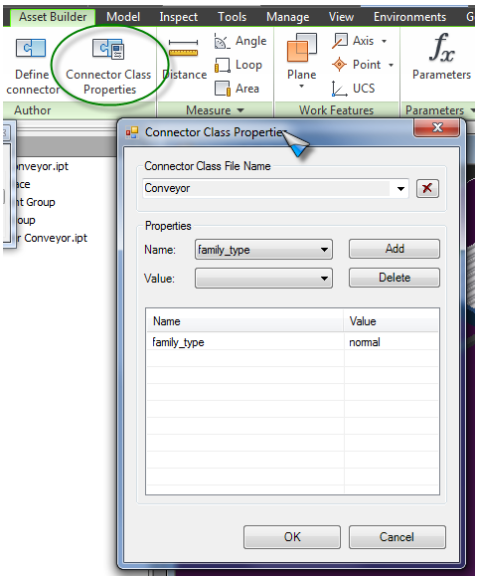
The following sections provide additional detail for these improvements.

Republish Support

This improvement is largely transparent. When you republish an asset that contains connection information (such as a conveyor section), since this information is stored directly within the Inventor model file, the newly-published asset will also contain it. So, when instances of this asset are placed into a Factory Layout, the connection logic specified by the connector class will be used automatically.

Defining Connector Class Properties

In the Asset Builder, we have added a new command, labeled “Connector Class Properties”. This command brings up the corresponding Connector Class Properties dialog:



Using this dialog, you choose the connector class to be used for the asset, and also define the connector class property values that are to be used for this particular asset. These “connector class properties” are defined in a separate “connector class” file. A set of system connector classes is provided (and used by the system assets), and additional custom connector classes can be defined.

To choose the connector class, you choose the desired connector class from the “Connector Class File Name” dropdown. This will be automatically populated with the connector class files found in the set of available asset collections.

To assign connector class properties, you choose the desired property from the “Name” dropdown, the desired value for the property from the “Value” dropdown, and click “Add” to add the mapping to the list box. To remove a mapped value, you can select the list item to remove, and click the “Delete” button.

In order for an asset to behave properly with respect to the connection, a value must be provided for all connector class properties. In the example above, only a single connector class property is required.

Creating Custom Connector Classes

For entirely new asset types, which require connection behavior beyond the out-of-the-box connector classes, it’s possible to define custom connector classes. These can then be used to control parameter propagation behavior among assets assigned to the class.

A “connector class” is defined by an XML file, using the extension “.connectorClass”. The elements of the connector class define the parameter mapping, and any conditional checks that are used to control the conditions in which the particular mappings are made.

The structure of a connector class is quite simple. We’ll use the following simple connector class file in the descriptions that follow.

```
<connector>
    <ConveyorWidth value_source="parameter">Width</ConveyorWidth>
    <ConveyorHeight family_type="normal"
                    value_source="parameter">Height</ConveyorHeight>
    <ConveyorHeight family_type="inclined" connector_name="Connector1"
                    value_source="Parameter">UpperHeight</ConveyorHeight>
    <ConveyorHeight family_type="inclined" connector_name="Connector2"
                    value_source="Parameter">LowerHeight</ConveyorHeight>
    <ConveyorHeight family_type="spiral" connector_name="UpperConnector"
                    value_source="Parameter">UpperHeight</ConveyorHeight>
    <ConveyorHeight family_type="spiral" connector_name="LowerConnector"
                    value_source="Parameter">LowerHeight</ConveyorHeight>
</connector>
```

Connector class files always use a root element named “connector.” This contains a series of child elements that identify the parameters that are to be propagated between assets when they are connected – and belong to the same connector class.

The child elements define “mapping groups,” and can have any name desired, and are used according to the following rules:

1) Each mapping group, which represents a certain parameter, should have a unique element name. In the example above, “ConveyorWidth” and “ConveyorHeight” represent the 2 mapping groups.
2) In a given asset, once a match is found for a parameter meeting the matching rules described below, testing of the current mapping group ends, and the next mapping group is processed. Therefore, mapping group elements should be specified in decreasing order of specificity – most specific to least specific.

The value of these elements usually identifies the parameter name to be mapped (e.g., “Width” above). Both assets must provide a parameter matching this name, or no mapping will be performed.

Attributes on these elements provide additional information to control the mapping operation. These attributes are described in the following table.

Attribute	Attribute Values	Notes
value_source	Parameter (default)	This specifies that the value for the mapping is to be provided by the identified parameter in the source asset.
	Constant	This specifies that the value to be assigned will be a constant value, provided in the **value** attribute.
value		This attribute provides the value to be used for the parameter when **value_source** is set to **Constant.**
connector_name		This attribute is used to restrict the matching to a specific connector (by name). If the connector being used for the connection in the asset does not match this name, the mapping will be ignored.

In addition to the above reserved attributes, additional attributes can be specified, which define matching conditions. The attribute names (e.g., “family_type” in the above example) identify “connector class properties”, and the values for these properties must be specified within the asset (via the Connector Class Properties dialog) in order to facilitate matching. If the asset does not provide a matching connector class property, the mapping will be ignored.