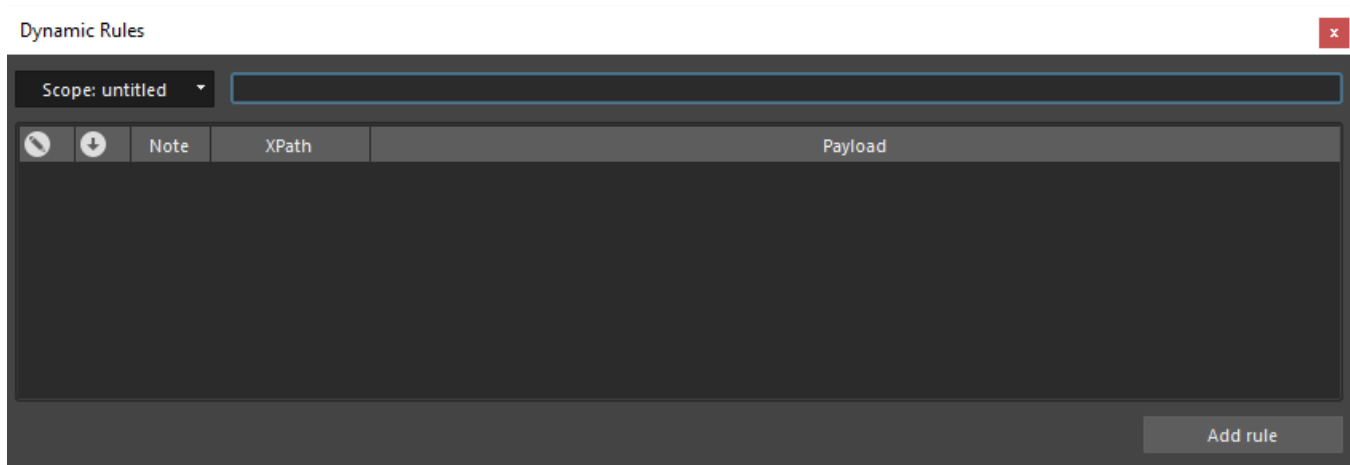


Dynamic Rules

The *Dynamic Rules* interface allows you to dynamically bind shaders to geometry and modify shading parameters by setting up a series of rules. The Dynamic Rules window is accessed via the *RenderMan* menu:

RenderMan > Dynamic Rules



Creating Rules

A rule maps objects to a payload. The objects can be geometry, sets of geometry, or shader parameters, and the payload is a chunk of data that can be inserted into a RIB stream for the object. For example, a rule could map a specific shader (the payload) to all the geometry in your scene that has a name beginning with "buzz". The payload gets applied to objects that match the [XPath](#) expression in the rule.

A new rule can be created using the *Add Rule* button at the bottom of the window, or by right-clicking the empty table and selecting *Insert Rule* or *Import Rule*.

Each rule has a set of properties in the table:

- **Phase:** Determines when the rule is executed. Double-click the Phase setting to change its value. A rule is executed in either the Injection or Edit phase. An injection rule acts first, and deposits one or more Ri Requests into the Ri stream. For example, an injection rule can be used to dynamically bind a shader to some geometry. An edit rule acts last and edits the parameters of an individual Ri request. For example, this type of rule can be used to dynamically change shader parameters.
- **Flow:** Determines whether additional rules should be evaluated. Double-click the Flow setting to change its value. When a rule's [XPath](#) expression finds a match, the rule is said to succeed. Upon finding a match, the associated payload is obtained and stored on a stack. At this point, execution will either terminate or continue as governed by the flow of the rule. When the rule set is exhausted or a matching rule terminates the rule search, the process moves on to the injection phase, where the stack of payloads is delivered to the renderer. Those payloads encountered last will appear first in the Ri stream; this allows higher-precedence overrides to compose with lower-level defaults, following standard Ri behavior.
- **Note:** You can add helpful information about the rule in this property. For example, information about which objects the XPath expression matches.
- **XPath:** Objects that match the XPath expression will get the payload applied to them. You can validate the XPath expression by right-clicking this field and select **Validate**.
- **Payload:** Determines the data that is inserted into the RIB stream for the objects that match the XPath expression. For injection rules, the data can be a RiSurface call for a shader or some other RIB that is specified in a RIB Box. You can display a payload selection tool for injection rules by either double-clicking the payload field or right-clicking the field and selecting **Insert Payload**. This will display a selection tool.

You can choose the type of payload by clicking on the arrow next to the text box in the selection tool.

To add a payload for an edit rule, you can right-click the payload field and select **Add Value Payload**.

Managing Rules

Once created, an entire rule can be selected by clicking on the rule number in the table (the vertical header in the table). Multiple rules can be selected using Shift+click or Ctrl+click. Once rules are selected, you can cut, copy, paste, and duplicate the rules by right-clicking the vertical header in the table and selecting one of the menu items. You can also drag and drop rules to change their order.

Filtering Rules

You can control the rules that are displayed in the Dynamic Rules table by changing the selected scope with the *Scope* button at the top of the table. The text box to the right of the Scope button allows you to filter the rules. You can type a regular expression in the filter text box, and only the rules that have a *Note*, *XPath*, or *Payload* value that match the expression will be displayed in the table.