# MEL Attributes

## The Anatomy of a RenderMan attribute

All RenderMan attributes are prefixed with rman____ (that's two underscores), and all RenderMan attributes follow this naming convention:

    rman__setting-type__namespace_setting-name

But luckily, you don't need to remember that. There are ways to query the name of a setting which will be described in the next section.

Setting types include:

- toropt
  - Translator options. These settings are used in the process of translating a maya scene. They apply across an entire rendering job.
- torattr
  - Translator attributes. These settings are also used in the process of translating a maya scene. They can vary between passes, but they don't have to. For example, it's possible to render multiple passes for which only some have motionBlur enabled.
- riopt
  - RenderMan Options are settings that affect the rendering of an entire image. The names of some riopts and riattrs may require a namespace.
- riattr
  - RenderMan Attributes are settings that are part of the graphics state, and unlike Options can be associated with individual primitives.
- param
  - Params can refer to either shader parameters or command parameters. In the case of shading parameters a simpler naming convention is followed: rman__param-name. Command parameters follow the same convention as other setting types.

## How to figure out the name of an attribute

There are a couple ways you can find out the name of an attribute. The easiest is to select the node you want to add an attribute to, then open the RenderMan Attributes window (Attributes->RenderMan->Manage Attributes...) The window lists all the attributes that you can add to the selected node. These are listed as labels, because they're more readable that way. When you click on one of the labels, the corresponding name of the attribute appears in the description field at the bottom of the window. You can jot it down for future use.

Another way to find out the name of an attribute is to look in the DeclarationTable in RenderMan_for_Maya.ini. It lists all the known attributes, but their names aren't quite in the format you need. Here's are a couple example declarations:

```
declare riattr {float shade_relativepixelvariance} {
    label "Relative Pixel Variance"
    range {0 1 0.01}
    subtype slider
    description "Values from 0.1 to 1 allow for faster
            renders, while a value of .01 is higher quality and slower."
}

declare riattr {int trace:maxspeculardepth} {
    label "Max Specular Depth"
    subtype slider
    range {0 10 1}
    description "Number of bounces for reflections and
            refractions.  Value of 1 or 2 is sufficient unless you
            need multi bounce effects."
}
```

The piece of this declaration that you can use to find out the corresonding attribute name in maya is the name of the setting -- the portion after the data type. In these two examples that's "RelativePixelVariance" and "trace:maxspeculardepth". Note the latter has what's called a namespace, and the former doesn't. That's nothing you need to be particularly conscious of, just know that it takes part in the attribute name. You can pass the name of the setting to a MEL script called:

    global proc string rmanGetAttrName( string $declname )

For example:

```
rmanGetAttrName "RelativePixelVariance";
// Result: rman__riattr___shade_relativepixelvariance //
rmanGetAttrName "trace:maxspeculardepth";
// Result: rman__riattr__trace_maxspeculardepth //
```

## Which attributes might you want to add to a node?

To get a list of the attributes you might want to add to a node, try this command:

```
global proc string[] rmanGetOptionalAttrs( string $node )
```

Example:

```
rmanGetOptionalAttrs nurbsSphereShape1;
// Result: rman__riattr___ShadingRate rman__riattr___SmoothShade
rman__riattr___MatteObject rman__riattr___DoubleSided
rman__riattr___ReverseOrientation rman__riattr___MotionFactor
rman__riattr__cull_backfacing rman__riattr__cull_hidden
rman__riattr__derivatives_centered rman__riattr__derivatives_extrapolate
rman__riattr__dice_rasterorient rman__riattr__grouping_membership
rman__riattr__identifier_name rman__riattr__identifier_objectid
rman__riattr__sides_backfacetolerance rman__riattr__sides_doubleshaded
rman__riattr__stitch_enable rman__riattr__stitch_newgroup
rman__riattr__trace_bias rman__riattr__trace_displacements
rman__riattr__trace_maxdiffusedepth rman__riattr__trace_maxspeculardepth
rman__riattr__trace_samplemotion rman__riattr__visibility_camera
rman__riattr__visibility_specular rman__riattr__visibility_diffuse
rman__riattr__visibility_transmission rman__riattr__visibility_midpoint//
```

## How to add an attribute to a node

Now that you know how to figure out the name of an attribute, the recommended way of adding it to a node is with a MEL script called "rmanAddAttr". Here's its definition:

```
global proc rmanAddAttr( string $node, string $attr, string $val)
```

It takes three arguments, the node name, the attribute name, and the default value. RFM will figure out the appropriate type of attribute to add to the maya node, and will convert the value string to the expected type. The value can be an empty string if you want RFM to use its own default for the setting.

Some examples:

```
rmanAddAttr nurbsSphereShape1 `rmanGetAttrName "trace:maxspeculardepth"` "4";
```

## How to add bunches of attributes to nodes

The selection sensitive menu entries which appear in the Attribute Editor under the Attributes->RenderMan menu usually add more than one attribute. And some of them, like "Add Subsurface Scattering", even do more complex things like creating a new network of pass nodes. It might be handy to invoke these entries via MEL rather than from the menu. Here's how you can do that.

Note, it's also possible for you to add menu entries to the Attributes->RenderMan menu. These are defined toward the end of RenderMan_for_Maya.ini.

The command definition is:

```
global proc rmanExecAEMenuCmd( string $node, string $menuItemLabel)
```

## How to delete attributes

There's nothing special you need to know about deleting attributes. It's done in the same way as deleteing other maya attributes, with the deleteAttr command.

Example:

```
deleteAttr nurbsSphereShape1.rman__riattr___RelativePixelVariance
```

## How to set attribute values

If you know the name of the node and attribute, you can use maya's setAttr command. RFM also provides a command that might make setting attributes a little easier in the respect that you don't need to know the data type of the attribute, and the value can be supplied as a string. Feel free to use whichever you prefer. This is its definition:

```
global proc rmanSetAttr(string $node, string $attr, string $val)
```