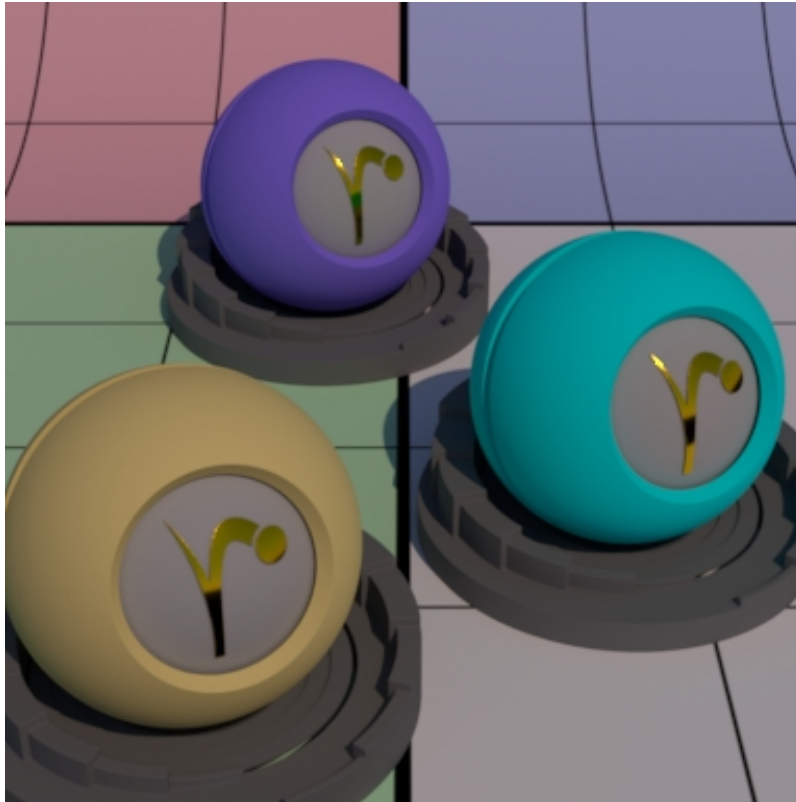


# PxrAttribute

PxrAttribute allows the user to read attributes attached to (stored) on a node. Such an example would be to add a color attribute to a set of objects to be read by a material later. In this way, a material can change its result based on the object being rendered instead of a different material. Below there are color attributes attached to the sphere's of the shader ball. A single [PxrSurface](#) material renders with a different diffuse color as specific by each shape's defined attribute. Examples on usage are below.



There may be a performance penalty for using this node in many places in your scene. Efficiency is key to avoid too many evaluations of user attributes if not necessary.

## Input Parameters

### Variable name

This field takes a string that identifies the attribute. The string should include the namespace for the attribute and the attribute name separated by a colon. For example, trace:maxdiffusedepth or user:Ball.

### Variable Type

This specifies the type of variable to read and must match what was specified above on the other nodes.

- Integer
- Float
- Float 2
- Color
- Point
- Vector
- Normal

## Output Parameters

### resultF

A float result.

### resultRGB

The color result.

## Example Usage

DCC applications may use a different mechanism for applying a user attribute. Below are two examples for applying a color user attribute named "Ball" to a shape:

### Maya:

Add a Pre Shape MEL attribute to the shape using the Attributes > RenderMan menu

```
RiAttribute "user" "color Ball" 1 0.2 0.65
```

### Katana:

The below is an OpScript example of the same attribute in Katana

```
gb = GroupBuilder()
gb:set("value", FloatAttribute({1.0, 0.2, 0.65}, 3))
gb:set("type", StringAttribute("color"))
Interface.SetAttr("prmanStatements.attributes.user.Ball", gb:build())
```

### Houdini:

See [Using PxrMaterialID](#) as an example of how to add user attribute in Houdini.