

User Attributes

[User attributes](#) are a convenient way to vary shading across multiple objects without changing a shading network. They are especially useful as they can vary across instances, unlike [primitive variables](#).

Setting User Attributes

Here is an example of how to set a color user attribute with an OpScript:

```
gb = GroupBuilder()  
gb:set("value", FloatAttribute({0.0, 1.0, 0.0}, 3))  
gb:set("type", StringAttribute("color"))  
Interface.SetAttr("prmanStatements.attributes.user.myColor", gb:build())
```

While all user attributes can be set up as a GroupAttribute, color attributes are the only type that requires it. Int, float, and string user attributes can be set directly, with their type inferred from the Katana attribute type:

```
Interface.SetAttr("prmanStatements.attributes.user.myFloat", FloatAttribute(3))
```

Using User Attributes

RenderMan ships with several plugins that make use of user attributes. [PxrAttribute](#) or [PxrVary](#) are patterns that read user attributes. The [PxrCryptomatte](#) sample filter can create different outputs based on string user attributes. When using these plugins, the parameter for the attribute should include the prefix `user:` followed by the attribute name.