

# Understanding Collections

Collections are a context sensitive shader and attribute assignment strategy, the context being defined by a *Render Pass*. When *3Delight for Maya* sends an object for rendering, it goes through the following steps to decide what shader to use (same decision path is followed for attributes):

1. If the rendered pass has a collection specified in its *Collection to Render* attribute (in the [Render Sets](#) section of the *Scene Elements* group of rendering attributes) and the collection has a shader specified in "Override Assignments", then use it.
2. If the *Render Pass* has a collection specified in "Render Set" and the object has a shader specified in "Collection Assignments" for the same collection, use it.
3. If the object has a shader specified in "Object Assignments", use it.
4. Use *Maya's* Hypershade network as the shader.

Collections are a powerful tool in multi-pass rendering. Say that you have a specular and a diffuse pass, in the specular pass you want to use a Phong shader and in the diffuse pass you want to perform some "ambient occlusion". You can achieve this by following these steps:

1. Create two collections, one 'specular' and one 'diffuse'. Collections can be created by clicking on the "texture square" near the "Collections" line in the assignment panel (first line in the panel, left hand side).
2. Create two *Render Passes*: one choosing the 'specular' collections and the other one choosing the 'diffuse' collection in the Render Sets section (see the [Render Sets](#) section of the *Scene Elements* group of rendering attributes).
3. Using the assignment panel, assign a specular shader (such as 'metal') to your object in the 'specular' collection and attach some ambient occlusion shader in the 'diffuse' collection.

Rendering the first pass will give you the specular highlights and rendering the second pass gives you the ambient occlusion. This working scheme frees you from writing big shaders with built-in context sensitive code (which often gets messy) and encourages you to write re-usable, simpler shaders.