Subdivision Surfaces in Katana

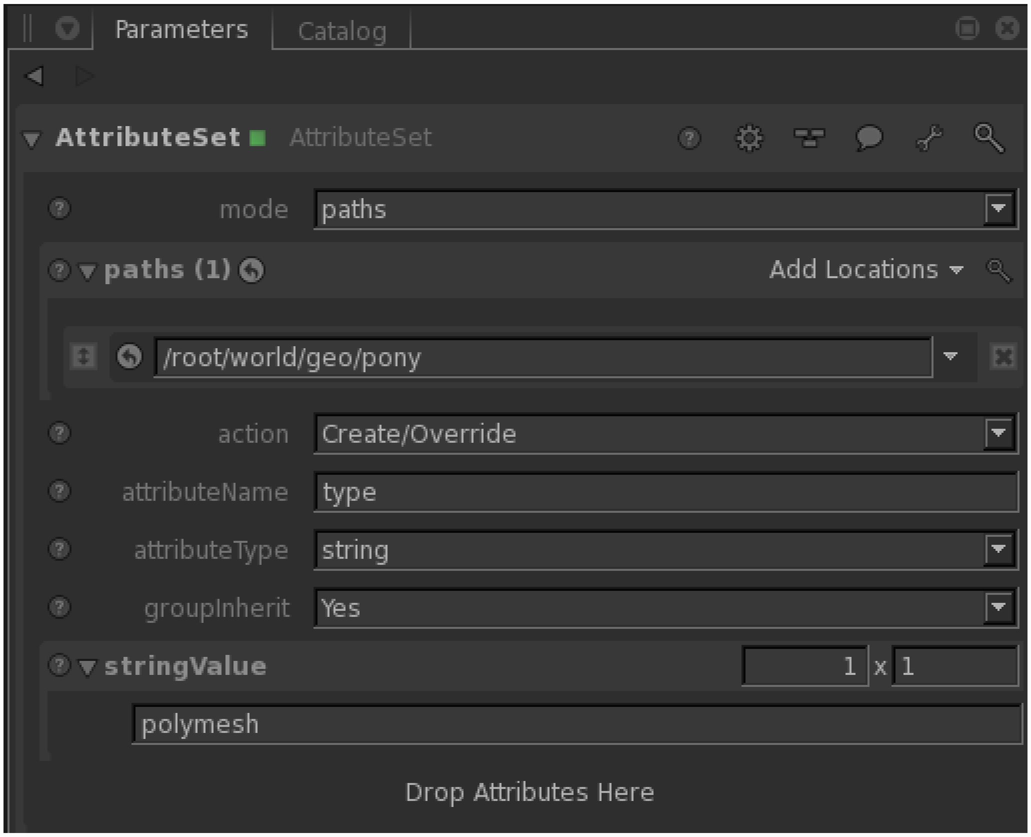
Subdivision surfaces provide you with a great way to use the simplicity of polygonal-style modeling, together with the smoothness of a mathematically defined surface. They are used heavily when modeling organic or curved surfaces. The RenderMan documentation for Subdivision Surfaces will give you a great primer on them.

In Katana, you will most likely bring in a subdivision surface via Alembic or some other geometry format that supports them. You as an artist are not likely to have to do anything special to utilize them in Katana.

Any texture coordinates or other primitive variables that you need for look development must be added upstream in your pipeline. They will be available when you import your geometry into Katana.

Speed Trick: If your subdivision surface is off-screen or the pixel coverage is very small, it may make sense to convert the subdivision surface to a polygonal surface. This will save memory in your render, will shorten the time to seeing the first pixel, and depending on your scene complexity, will reduce render times.

To change your geometry from a subdivision surface to a polygonal surface, use an AttributeSet node to set the "type" of the geometry to polymesh. Below is a screen snapshot of what you will need to do.



mode: paths

paths (1): /root/world/geo/pony

action: Create/Override

attributeName: type

attributeType: string

groupInherit: Yes

stringValue: 1 x 1

polymesh

Normals and faceting

When rendering surfaces in Katana, you may need to discard normals. The basic workflow for rendering smooth-looking surfaces is below:

- Export normals for poly meshes with smoothed normals that you wish to preserve when they are *not* going to being subdivided. This is a common "fake" for low-poly surfaces that need to look smooth. No normals export is needed when you only require hard edges/faceting.
- Going to subdivide the mesh? Do not export normals for these poly meshes, let PRMan calculate them new when they are subdivided. This also saves space for the exported file since you don't need the normals.

The presence/absence of normals may affect tessellation. Otherwise, remove unwanted faceting by deleting normals in Katana if they were exported despite your intent to subdivide.

To Duplicate the Geometry Settings from Maya

It may be useful to match what you're seeing in Maya since it may be your primary modeling package. Below are the settings necessary to achieve the same look if you asset was not exported with the data.

1. Create an attribute Script node
    a. In the CEL section, select the mesh(es) where you wish to apply the attributes. In this case we've added the subdmeshes from Maya

i. Custom CEL statements
   1. /root/world/geo//*{@type == "polymesh"}
   2. /root/world/geo//*{@type == "subdmesh"}

ii. In the script section add the following:
   1. DelAttr('geometry.vertex.N')
   2. SetAttr('type', ['subdmesh'])
   3. SetAttr('geometry.facevaryinginterpolateboundary', [3])
   4. SetAttr('geometry.facevaryingpropagatecorners', [0])
   5. SetAttr('geometry.interpolateBoundary', [1])
   6. 'geometry.vertex.N' (*Note: This deletes the existing normals off the mesh as renderman will create its own for a SubD mesh. This is explained in the above section.)*
   7. 'type' changes the mesh to a subdivisional surface

2. We set several attributes above to match the Maya settings. Below we list the other options should you need to experiment.
   a. Options for geometry.facevaryinterpolateboundary
      i. 0(old style)
      ii. 1(new style)
      iii. 2(new style, no corners)
      iv. 3(new style, smooth internal only) *This is the current Maya default*

   b. Options for 'geometry.facevaryingpropogatecorners'
      i. 0(off) *This is Maya default*
      ii. 1(on)

   c. Options for 'geometry.interpolateBoundary'
      i. 0(no interpolation)
      ii. 1(edge crease, corner crease) *This is the Maya default*
      iii. 2(edge crease only)

For Developers

If you are implementing support in a custom Katana plugin for subdivision surfaces, you will be interested in how RenderMan for Katana translates geometry attributes to RenderMan's subdivision format. See the developer documentation here.

We support the following attributes, which are converted to RenderMan's internal format. You can also examine the code for RenderMan for Katana to see the details of how each are handled.

- geometry.edits
- geometry.edits.tags
- geometry.edits.numArgs
- geometry.edits.intArgs
- geometry.edits.floatArgs
- geometry.edits.stringArgs

- global.prmanStatements.subdivisionMesh.scheme
- global.prmanStatements.subdivisionMesh.attributeSubsets "shading"|"geometrydefinition"|"geometrymodification"

- geometry.faces
- geometry.faces.visible
- geometry.faces.material
- geometry.faces.prmanStatements
- geometry.faces.lightList

- geometry.holePolyIndices

- geometry.creaseLengths
- geometry.creaseIndices
- geometry.creaseSharpness
- geometry.creaseSharpnessLengths

- geometry.cornerIndices
- geometry.cornerSharpness

- geometry.facevaryinginterpolateboundary
- geometry.facevaryingpropagatecorners
- geometry.interpolateBoundary