

Compositor / Effects registry

Currently FG will load effects from \$FGData/Effects by looking for the file.

To enhance this for Compositor it is proposed to extend this by implementing an Effects registry; this works by having an XML file that contains the Effects that are available in each pipeline.

So in \$FGData/Compositor there will be

- Global \$FGData/Compositor/effects.xml
- Pipeline specific \$FGData/Compositor/<pipeline>/effects.xml

The actual files will be contained within Compositor/Effects with the shaders being in Compositor/Effects/Shaders (again to keep things nicely organised)

Both of these files will be in the usual property list format and simply provide a mapping between files on disk and a well-known name.

So when a model uses the directives `<inherits-from>Effects/model-interior</inherits-from>` the first step will be to check the effects registry for the entire key, in this case **Effects/model-interior**. If there is nothing registered with this key the current system of looking in various places on the disk for the effect key with **.eff** appended will be used.

To maintain backwards compatibility the `<inherits-from>` tag will work as before, however there will be a new tag `<uses>` that operates only on shaders that are defined in the registry; and another new tag `<fallback>` that defines the action when the shader isn't available

```
<effect>
  <name>fuselage</name>
  <uses>Effects/shadow</uses>
  <fallback>disabled</fallback>
  <object-name>pitch-ratio-panel</object-name>
</effect>
```

Sample effects.xml

```
<PropertyList>
  <effect>
    <name>Effects/shadow</name>
    <description>Render shadows</description>
    <group>Lighting</group>
    <can-disable>true</can-disable>
    <definition>Effects/shadow.eff</definition>
  </effect>

  <effect>
    <name>Effects/pointlight</name>
    <description>Emissive point light</description>
    <group>Lighting</group>
    <can-disable>true</can-disable>
    <definition>Effects/pointlight.eff</definition>
  </effect>
</PropertyList>
```

We can then have a dialog in the rendering that allows these effects to be selected based on the XML definitions (group, description)

Rendering pipeline configuration

Lighting

- Render shadows
- Emissive point light

Close