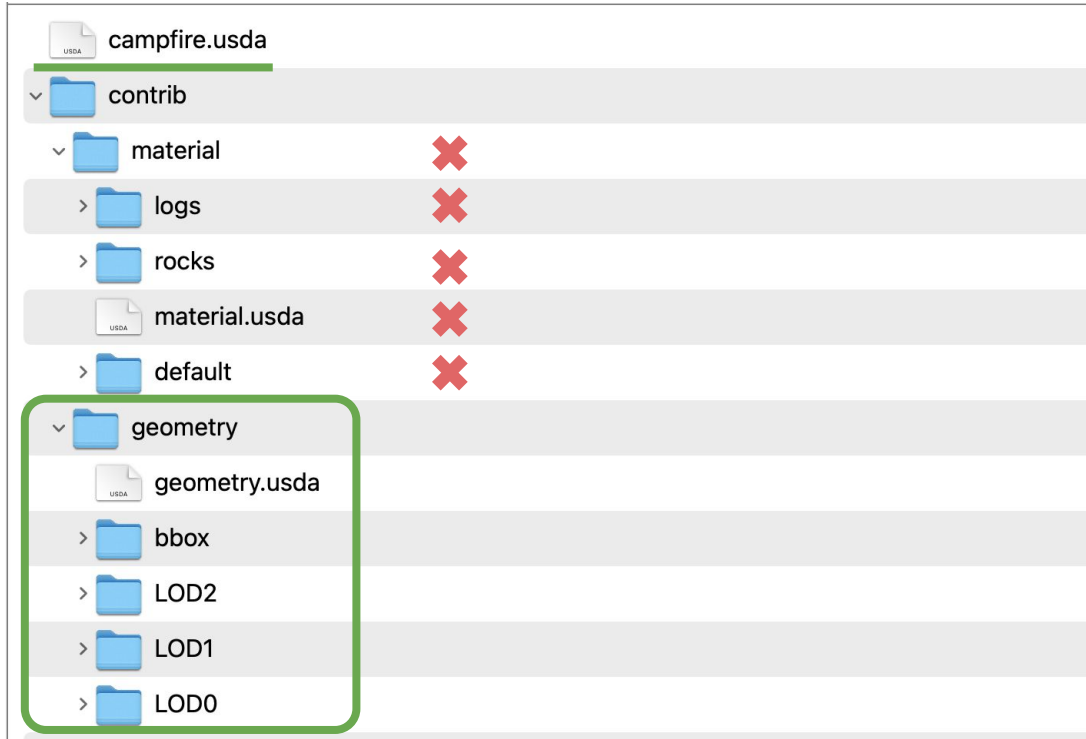


CIS 7000 USD Workflows

1. Check-in / Check-out Workflow
2. (Hypothetical) artist workflow
3. Additional Resources

1. Check-in / Check-out Workflow

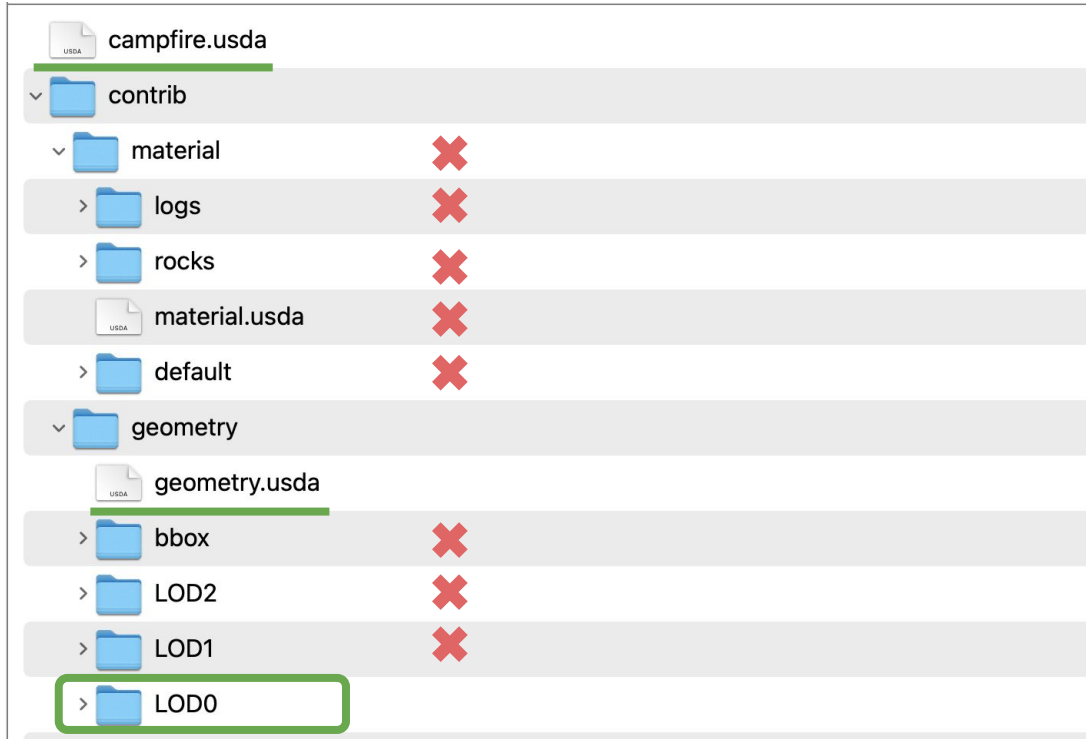
“I want all geometry”



■ = required

✗ = can be marked
read-only or not returned

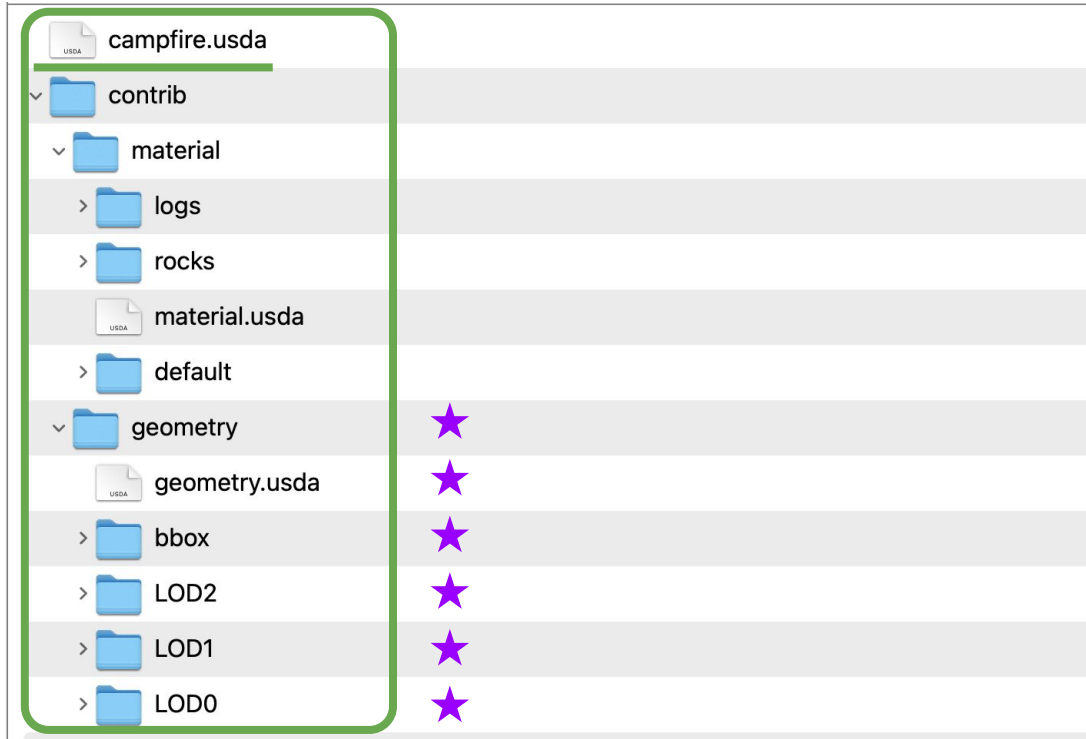
“I want LOD0 only”



■ = required

✘ = can be marked
read-only or not returned

“I want all material”

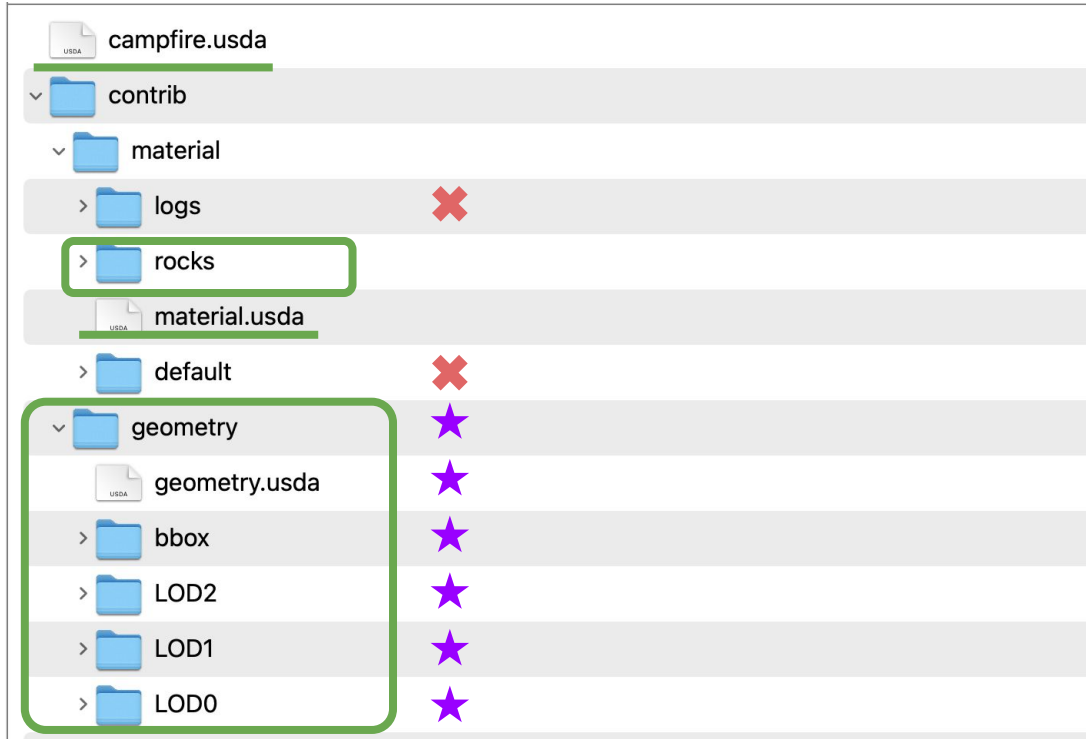


■ = required

✘ = can be marked
read-only or not returned

★ = can be read-only

“I want rock material only”



■ = required

✗ = can be marked
read-only or not returned

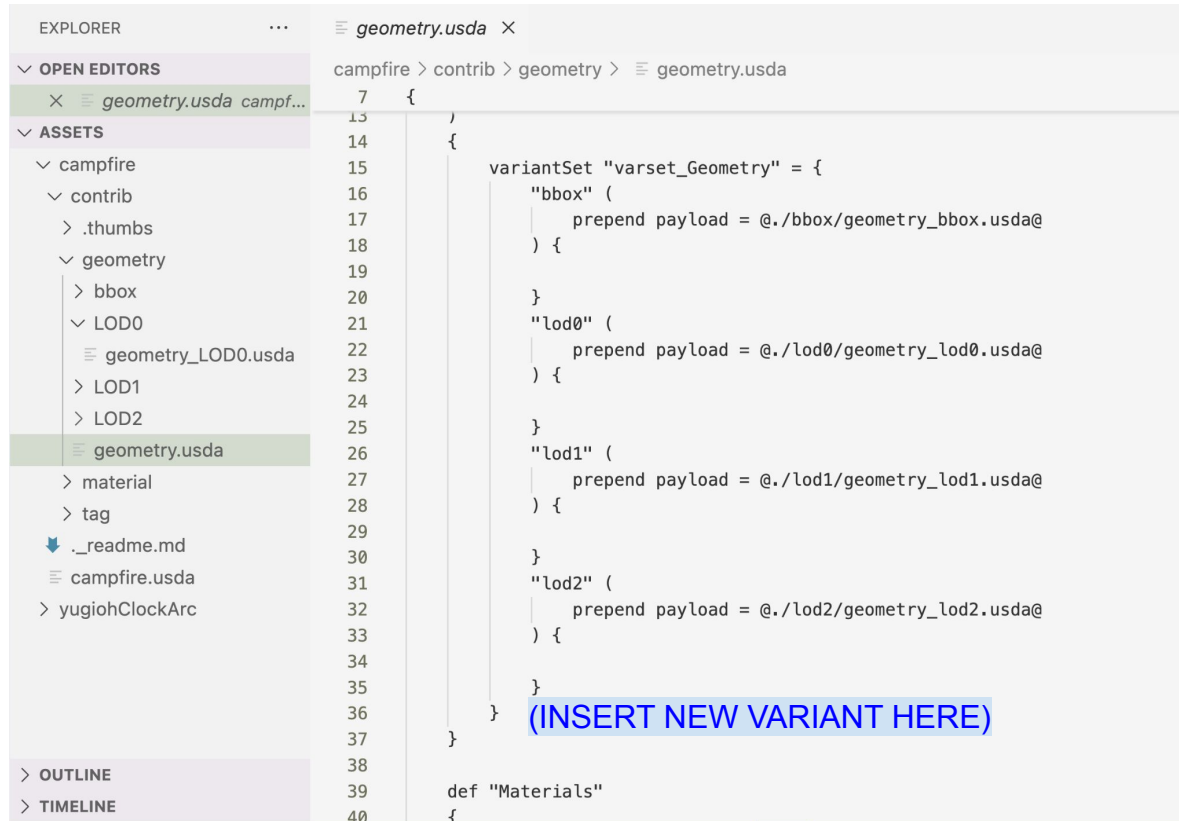
★ = can be read-only

2. (Hypothetical) artist workflow

“I am a geometry artist”

3. I add my new geometry variant to the variant set

at ./{ASSET_NAME}/contrib/geometry/geometry.usda



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders for 'campfire', 'contrib', and 'geometry'. The 'geometry' folder is expanded, showing sub-folders like 'bbox', 'LOD0', 'LOD1', 'LOD2', and 'material'. The 'geometry.usda' file is selected and open in the editor. The code in the editor is a variant set definition for 'varset_Geometry'. It includes variants for 'bbox', 'lod0', 'lod1', and 'lod2', each with a 'prepend payload' pointing to a corresponding USD file. A blue highlight is placed over the text '(INSERT NEW VARIANT HERE)' at the end of the variant set definition.

```
7 {
13 }
14 {
15     variantSet "varset_Geometry" = {
16         "bbox" (
17             prepend payload = @./bbox/geometry_bbox.usda@
18         ) {
19         }
20     }
21     "lod0" (
22         prepend payload = @./lod0/geometry_lod0.usda@
23     ) {
24     }
25     "lod1" (
26         prepend payload = @./lod1/geometry_lod1.usda@
27     ) {
28     }
29     "lod2" (
30         prepend payload = @./lod2/geometry_lod2.usda@
31     ) {
32     }
33     (INSERT NEW VARIANT HERE)
34 }
35 }
36 }
37 }
38 }
39 def "Materials"
40 {
```

“I am a material artist”

1. I go and see what materials this geometry needs

at ./{ASSET_NAME}/contrib/geometry/{VARIANT}/geometry_{VARIANT}.usda

```
campfire > contrib > geometry > geometry.usda
7 {
37 }
38
39 def "Materials"
40 {
41   def "MaterialClasses"
42   {
43     class "class_Default" (
44       prepend apiSchemas = ["MaterialBindingAPI"]
45     )
46     {
47     }
48     class "class_Rocks" (
49       prepend apiSchemas = ["MaterialBindingAPI"]
50     )
51     {
52     }
53     class "class_Logs" (
54       prepend apiSchemas = ["MaterialBindingAPI"]
55     )
56     {
57     }
58   }
59 }
60 }
61
62
```

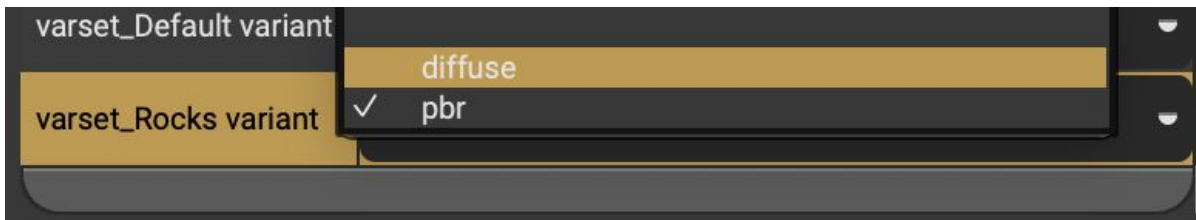
1. Rocks

2. Logs

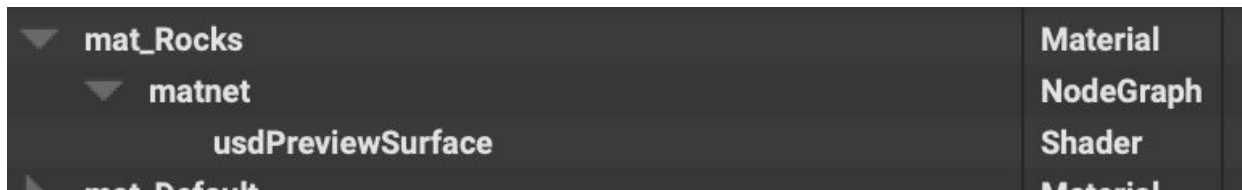
2. I create my material

at ./{ASSET_NAME}/contrib/material{VARIANT}/material_{VARIANT}.usda

Each material has variants

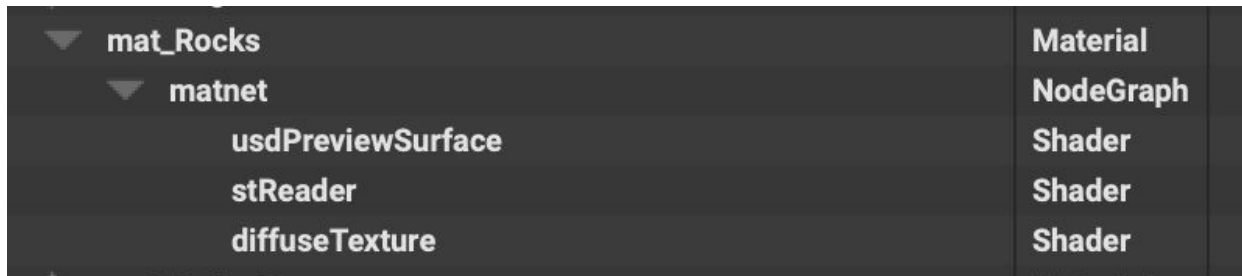


DIFFUSE variant



Each variant is a 'Material' prim with an underlying 'NodeGraph'

PBR variant



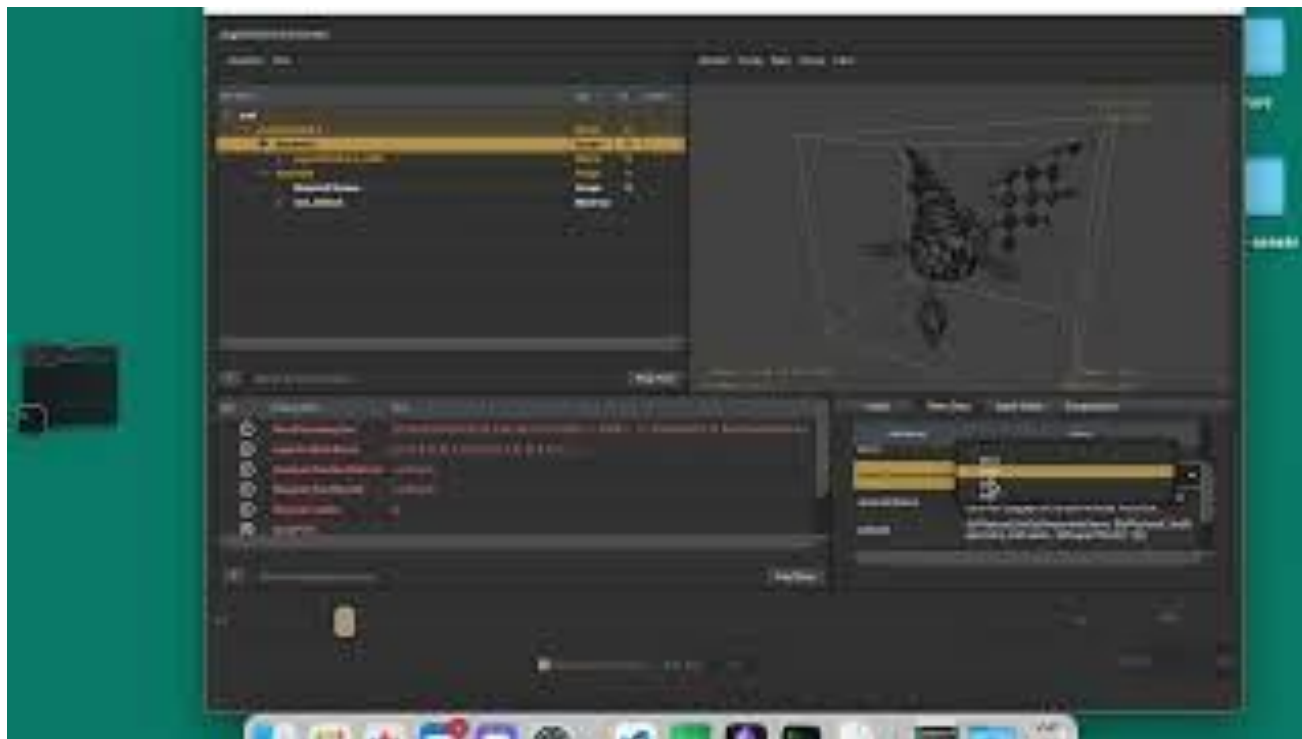
3. I bind my material to the previously created class

at `./{ASSET_NAME}/contrib/material/material.usda`

```
def Scope "MaterialClasses"  
{  
  over "class_Rocks"  
  {  
    | prepend rel material:binding = </campfire/Materials/mat_Rocks>  
  }  
}
```

3. Additional Resources

Structure Walkthrough Video in USD View



Very flattened overview of structure - [PDF Link](#)

```
#usda 1.0

def Xform "campfire"
{
  def Scope "Geometry" ( prepend variantSets = "varset_Geometry" )
  {
    def Xform "campfire_LOD0" ( apiSchemas = ["MaterialBindingAPI"] ) {
      rel material:binding = </campfire/Materials/mat_Rocks>
    }
  }

  def Scope "Materials" ( prepend variantSets = ["varset_Rocks", "varset_Logs"] )
  {
    def Material "mat_Rocks" {}
    def Material "mat_Logs" {}

    def Scope "MaterialClasses"
    {
      class "class_Rocks" ( apiSchemas = ["MaterialBindingAPI"] ) {
        rel material:binding = </campfire/Materials/mat_Rocks>
      }
      class "class_Logs" ( apiSchemas = ["MaterialBindingAPI"] ) {
        rel material:binding = </campfire/Materials/mat_Logs>
      }
    }
  }
}
}
```