

Fun Shelf Scripts

Version up sops that have a "version" parm - I use this to version up a lot of exports in tops or rops:

```
import hou
nodes = hou.selectedNodes()
for x in nodes:
    ver = x.parm("version").eval() + 1
    x.parm("version").set(ver)
```

Send a ROP fetch to TOPs name "topnet1" for convenient cooking.:

```
import hou
nodes = hou.selectedNodes()
topnet = hou.node("/obj/topnet1")
frb = round((hou.playbar.frameRange()[1] - hou.playbar.frameRange()[0]) / 2)

if not topnet:
    topnet = hou.node("/obj").createNode("topnet")
    topnet.setName("topnet1")
if not hou.node("/obj/topnet1/rangegenerate_start"):
    rg = topnet.createNode("rangegenerate")
    rg.setName("rangegenerate_start")
for x in nodes:
    ver = x.parm("ver")
    path = x.path()
    name = x.name()
    if x.parm("extra"):
        name = name + x.parm("extra").eval()
    else: pass
```

```

topcache = topnet.createNode("ropfetch")
p = hou.IntParmTemplate("version", "Version", 1)
g = topcache.parmTemplateGroup()
g.append(p)
topcache.setParmTemplateGroup(g)
topcache.parm("version").set(ver)
topcache.parm("roppath").set(path)
topcache.parm("framesperbatch").set(frb)
topcache.setName(name)

```

Send SOP to a ROP net named "ropnet1", adds version parm to update the original SOP version cache:

```

import hou
nodes = hou.selectedNodes()
ropnet = hou.node("/obj/ropnet1")
for x in nodes:
    name = x.name()+"_"+str(x.parent())
    path= x.path()
    if x.type().name() == "rop_geometry":
        ver = x.parm("vp_version")
    elif x.type().name() == "rop_alembic":
        ver = x.parm("vp_version")
    else:
        ver = x.parm("version")
    topcache = ropnet.createNode("fetch")
    topcache.parm("source").set(path)
    p = hou.IntParmTemplate("version", "Version", 1)
    g = topcache.parmTemplateGroup()
    g.append(p)
    topcache.setParmTemplateGroup(g)
    topcache.parm("version").set(ver)
    topcache.setName(name)

```

Make a file node from a rop export node:

```

import hou, re
nodes = hou.selectedNodes()
parent = nodes[0].parent()
nodeList = []
for x in nodes:
    name = x.name()+"_load"

    version = x.parm("vp_version").eval()
    path= x.parm("sopoutput").eval()
    path = str(path)
    newpath1 = re.sub("_v\d+", "_v{0}", path).format("`padzero(3,ch('version'))`)")
    newpath1 = re.sub("\\.d+", ".$F4", newpath1)

    file = parent.createNode("file")
    try:
        file.setName(name)
    except:
        continue
    file.setColor(hou.Color(0,0,1))
    file.parm("file").set(newpath1)
    p = hou.IntParmTemplate("version", "Version", 1)
    g = file.parmTemplateGroup()
    g.append(p)
    file.setParmTemplateGroup(g)
    file.parm("version").set(version)
    nodeList.append(file)

parent.layoutChildren(items=nodeList)

```

Open Directory from Tops

```

import hou, re
nodes = hou.selectedNodes()

for x in nodes:
    rpath = x.parm("roppath").eval()
    tnode = rpath.rpartition("/")[0]
    dirpath = hou.node(tnode).parm("file").eval()
    dirpath = dirpath.rpartition("/")[0]

```

```
hou.ui.showInFileBrowser(str(dirpath))
```

Get Alembic camera, import it, and link to other nodes:

```
import hou, os

#set parent network
network = hou.node("/obj")

#get shot number and cache directory
shot = os.getenv("SHOT")
shotcam = shot.split("/")[-1] + "_cam"
cache = os.getenv("CACHE") + "/layout/cam"

#get alembic file
getcampath = hou.ui.selectFile(cache, "SelectCamera", pattern="*.abc")
getcampath = hou.expandString(getcampath)

#make alembic node, set name, connect to SCALE node and add to CAMERA NODE
alnode = network.createNode("alembicarchive", shotcam)
alnode.parm("fileName").set(getcampath)
alnode.parm("buildHierarchy").pressButton()
alnode.setInput(0, hou.node("/obj/SCALE"))
hou.node("/obj/CAMERA").parm("CAMERA").set(f"/obj/{alnode}/CAM/CAMShape")
```

Revision #6

Created 30 January 2024 17:33:47 by Anthony

Updated 12 December 2024 18:35:19 by Anthony