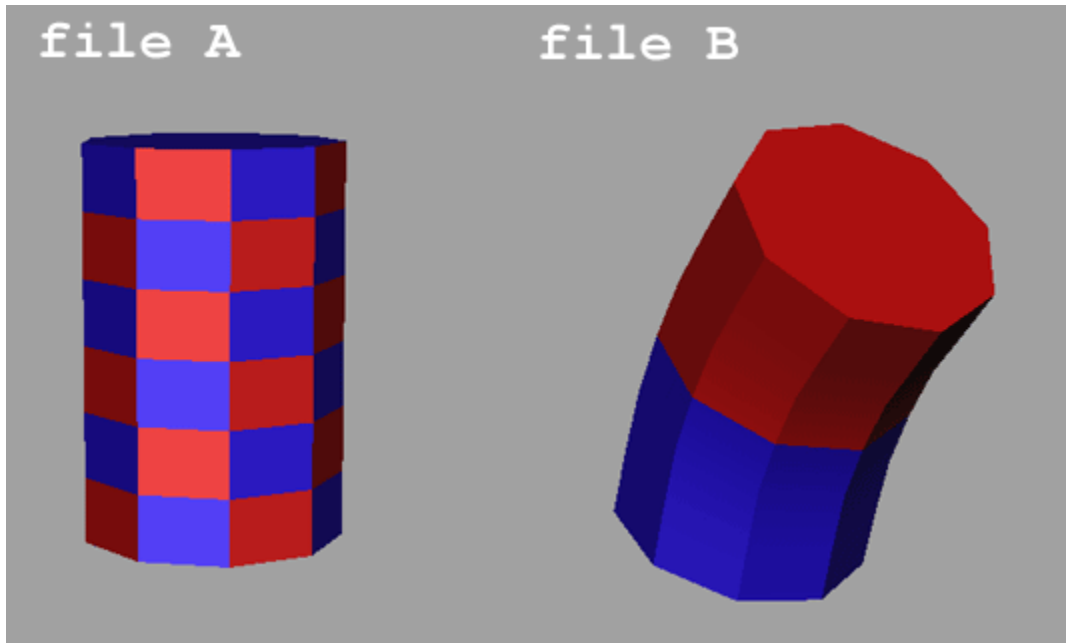


How to update material assignments on a referenced model.

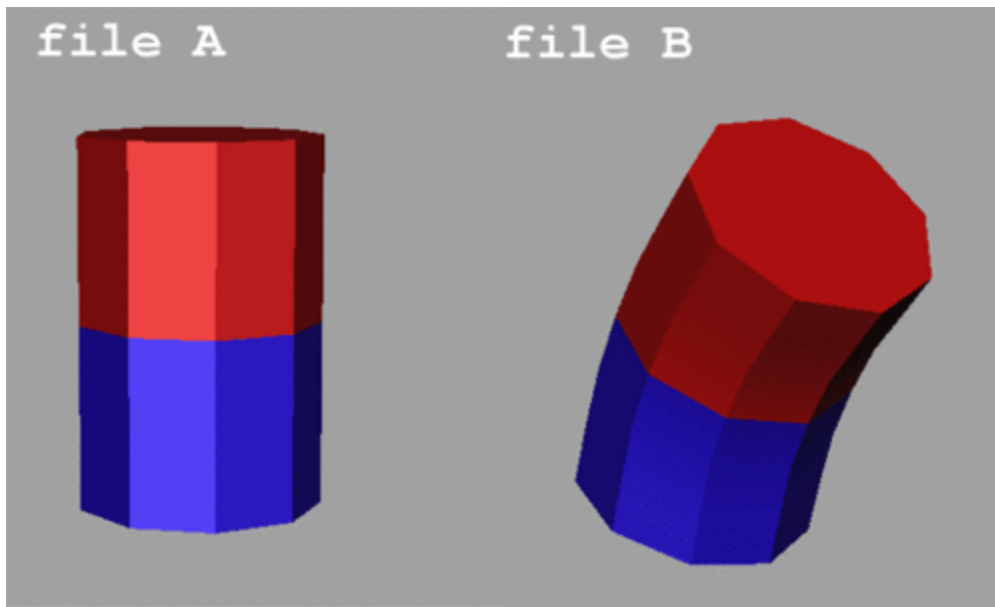
So you want to take your character model (let's call it fileA) and reference it into a new scene (fileB) so you can rig it... That way, your buddy can refine the model while you work on the rig.



Then you realise that none of the material assignments your colleague makes in fileA are visible to you on the referenced model in fileB.

You turn to your buddy...

1) The set up:



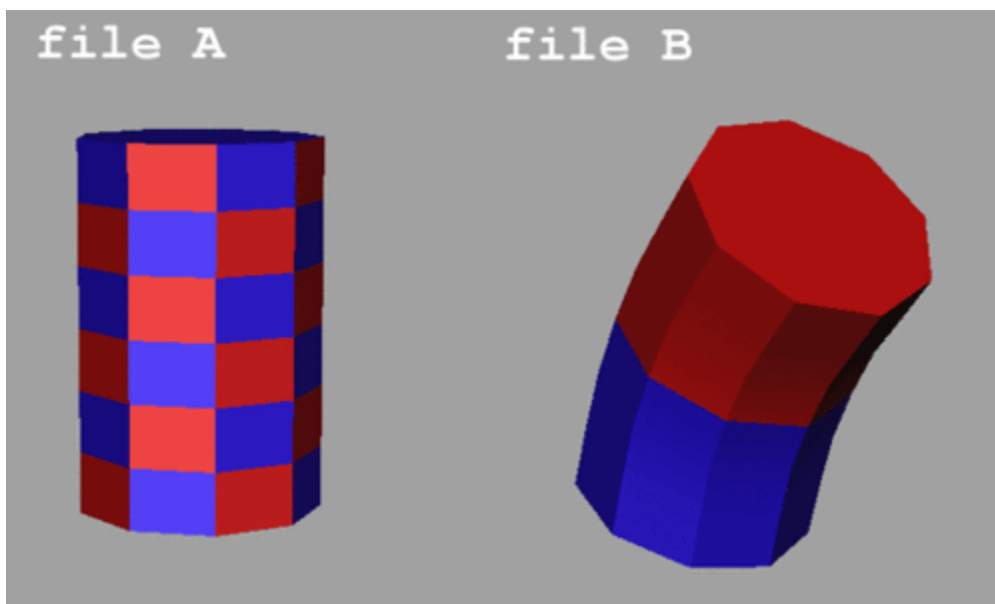
fileA consists of a polyCylinder.

The cylinder's faces have been assigned different materials.

fileB contains the rig, and a reference of fileA.

`fileA:polyCylinder` has been bound to a simple skeleton.

2) The problem:



When you modify the material assignment in fileA, you expect the changes to be visible once you reload the reference in fileB.

However, as you can see, the modifications are not apparent in fileB.

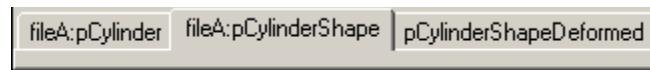
3) The issue:

It follows that since you reference the contents of fileA into fileB, you are working with the same object.

That's true up until you add a deformer. Once you do so, Maya updates the DG to look like this:

```
fileA:polyCylinderShape > skinCluster >  
polyCylinderShapeDeformed
```

..where the referenced node `fileA:polyCylinderShape` has become an intermediate node.



I'd like you to select the `fileA:polyCylinderShape` node and, in the Object Display tab of the AE, turn off the Intermediate Object attribute. The shape will appear in the viewport and you'll see that the correct material assignments reflect the changes you made to fileA.

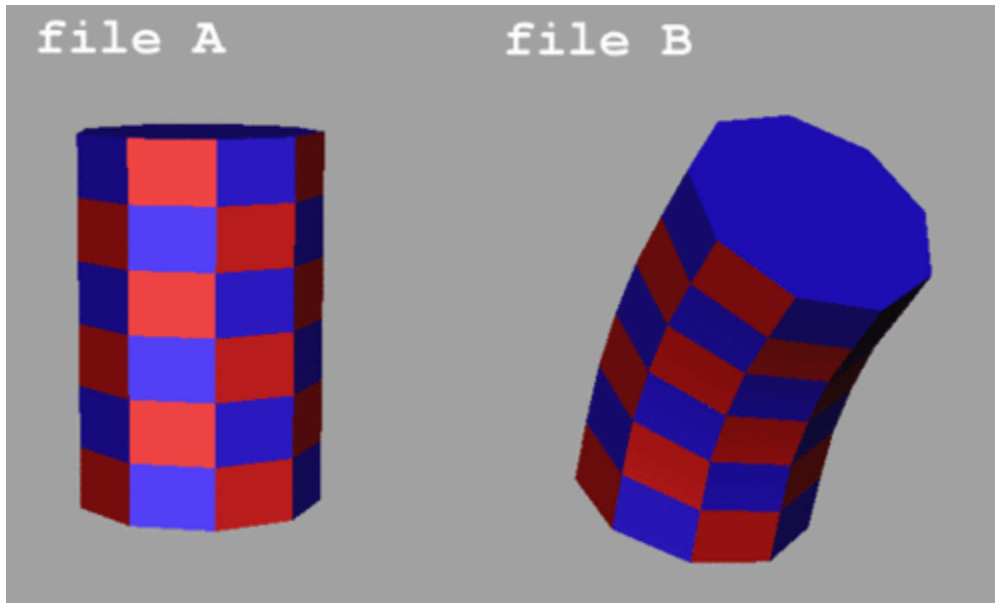
To resolve the problem, we have to somehow share the shading group sets between the `fileA:polyCylinderShape` and the visible `polyCylinderShapeDeformed` nodes.

If the material assignment had been made on a per object basis, this would not be a problem, since both our shape nodes share the same transform. In this case, however, the material assignment has been made at the component (face) level, and the problem is that the shading group sets connected to `fileA:pCylinderShape` don't automatically get connected to the `polyCylinderShapeDeformed` node.

4) The solution:

To pass the connections across, we'll use the MEL script `deformerAfterObjectSetMod (sourceShape, destShape)`, like this:

```
deformerAfterObjectSetMod fileA:pCylinderShape  
pCylinderShapeDeformed;
```



Use the MEL `deformerAfterObjectSetMod` to share shading group sets.