Subject: Autorotate for iTools?
Posted by K. Bowman on Wed, 11 Jan 2006 15:18:44 GMT
View Forum Message <> Reply to Message

Has anyone written something to programmatically rotate an iTools 3-D view and save a sequence of images for animation (a la XOBJVIEW_ROTATE)?

I have learned the basics of doing iTool manipulations programmatically, but am still very much a novice.

I would be quite happy to have something I can run from the command line, passing in the iTool id, rotation axis, number of frames or rotation increment, output directory, and file type (e.g., PNG).

Ken Bowman

Subject: Re: Autorotate for iTools?
Posted by David Alexander on Thu, 12 Jan 2006 23:37:49 GMT
View Forum Message <> Reply to Message

>> id=oTool->FindIdentifiers("*data space root")
>> oDSRoot=oTool->GetByIdentifiers(id)
>
> I could not find anything containing "root"

Wow, you're right. I must admit I didn't actually run these two lines. I figured out the identifier of the data space root a different way, then assumed these two lines would work. FindIdentifiers seems a little buggy to me, because the data space root object is definitely in the tool container hierarchy, so I don't know why it doesn't show up in FindIdentifiers.

So this encouraged me to look around some more, and I discovered a way to do this that is actually documented (!), sort of.

If you can get access to the object that you want to rotate (like the surface object in my example), you can call the GetManipulatorTarget method on the surface to get the object that needs to actually be rotated, like this:

```
oManipTarget=oSurface->GetManipulatorTarget()
if OBJ_VALID(oManipTarget) then $
  oManipTarget->Rotate,axis,angle
```

In this case, oManipTarget will be the data space root object. I think you still need to rotate the data space root, because if you rotate one of the other dataspaces you might succeed in rotating the

visualization, but not the selection graphics.

> Is there a direct way to get an iTool object reference given the iTool
> identifier?

Well, if you already have a reference to an object in the iTools hierarchy (in fact, it can be ANY object that subclasses from IDLitContainer), you can call its GetByIdentifier method. As long as you pass in a full identifier, it will give you the object you want, even if that object is in a different tool. For example (this is contrived, but shows what I'm talking about):

```
iimage
void=itGetCurrent(TOOL=oImageTool)
iplot,IDENTIFIER=idPlot
```

So now you have the object reference to the iImage tool, and the identifier to the iPlot tool. To get the object reference to the iPlot tool:

```
oPlotTool=oImageTool->GetByIdentifier(idPlot)
```

Voila!

Dave
