Subject: Re: Autorotate for iTools?
Posted by David Alexander on Wed, 11 Jan 2006 18:03:48 GMT
View Forum Message <> Reply to Message

Ken,

You could even write a custom operation that shows up in the Operations menu...

With the input info, this is really a two-step process inside a loop. You can call the Rotate method on the relevant dataspaceroot object, then access the File Export operation programmatically to save the image.

The dataspaceroot object subclasses from IDLitVisualization, so it inherits the Rotate method. There are actually several data space objects in each view, and you want to call Rotate on the highest level data space, ie, the one that contains all the others. This is an object of type IDLitVisDataSpaceRoot. Its identifier ends with "Data Space Root", so you can get it like this:

```
 id=oTool->FindIdentifiers("*data space root")
 oDSRoot=oTool->GetByIdentifiers(id)
```

However, if you have more than one view, you'll have to factor that in when calling FindIdentifiers.

You could also base the operation on which visualization is selected. You can call IDLitWindow::GetSelectedItems to see what is selected. This will return the visualization, e.g., an object of type IDLitVisSurface if you're using surfaces. So do something like this:

```
 oWin=oTool->GetCurrentWindow()
 oSelected=oWin->GetSelectedItems(COUNT=count)
 if count gt 0 then begin
   ;check to see if the visualization you're interested in is in the
list
   ;Let's say you're interested in the surface vis, and it's selected.
   oDS=oSurface->GetDataspace()
   oDS->GetProperty,PARENT=oDSRoot

   ;Then rotate
   oDSRoot->Rotate,axis,angle

   ;then save the image using the IDLitOpFileExport operation -
   ;You'll want to turn off the operation's UI dialog for this, and
set properties on the
   ;operation before calling.
```

```
  endif
```

I can go into more detail with the IDLitOpFileExport business if you want.

Dave