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

Subject: Re: Autorotate for iTools?
Posted by David Alexander on Wed, 11 Jan 2006 18:08:43 GMT
View Forum Message <> Reply to Message

Forgot to add this:

After calling the Rotate method, you should refresh the view:

oTool->RefreshCurrentWindow

Subject: Re: Autorotate for iTools?
Posted by Kenneth P. Bowman on Thu, 12 Jan 2006 22:18:05 GMT
View Forum Message <> Reply to Message

In article <1137002628.421292.26430@g49g2000cwa.googlegroups.com>,
"David Alexander" <davidessandro@yahoo.com> wrote:

Thanks very much for the help with this, David. After I posted my original query, I managed to get the second part of this task working (writing the image to a file), after much trial and error. Your suggestions were a big help with the first part.

> 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)

I could not find anything containing "root" but this works:

```
IF (N_ELEMENTS(view) EQ 0) THEN view = 'VIEW_1'     ;Default view to
rotate
data_space_id  =  itool_obj -> FindIdentifiers('*' + view + '*DATA
```

SPACE', /VISUALIZATION)

Is there a direct way to get an iTool object reference given the iTool identifier? What I am doing now is:

previous_id = itGetCurrent()
ITCURRENT, itool_id
temp = itGetCurrent(TOOL = itool_obj)

At the end of the program I set the current iTool to previous

ITCURRENT, previous_id

When I feel brave in a few days, I think I'll post this code so that everyone can point out the dumb ways I am doing things. This is a brave new world for us unreconstructed Fortran programmers.

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

I don't think I'm ready for that yet. I'll save it for the advanced course.

Ken
