
Subject: programming with itools

Posted by [Anne Martel](#) on Fri, 26 Jun 2009 16:20:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've finally decided to try to figure out how to use iTools and with the help of the examples on michael galloys page, searching through code in the idl distribution and a lot of trial and error I have managed to setup a viewer with 2 windows, one showing a volume and the other showing a slice through the volume.

I am stuck on a couple of things (well more than a couple but this is my immediate problem) and was hoping that the solutions are obvious to people who have played with itools more than I have:

a) why does the line

```
oimplane -> setproperty, orientation = 'Z'
```

not have any impact on the orientation - it always sets to the default x orientation?

b) is there any way of setting the 2nd viewport to iimage, ie no axes or rotations

The procedure I have so far is:

```
ivolume,/test,view_grid=[2,1]
```

```
id = itGetCurrent(tool=otool)
```

```
vol_id = otool->findIdentifiers('*data_space/volume', $  
                                /visualizations)
```

```
ovol = otool->getByIdentifier(vol_id)
```

```
;this bit sets up an image plane
```

```
ops_id = otool->findIdentifiers('*imageplane*', /operations)
```

```
oimplane = otool->getByIdentifier(ops_id)
```

```
;this next line doesn't do anything!
```

```
oimplane -> setproperty, orientation = 'Z'
```

```
result = otool->doAction(ops_id)
```

```
;now grap the data for the image plane
```

```
oSelVis = oTool->GetSelectedItems(count=nSelVis)
```

```
oData = oSelVis->GetParameter('IMAGEPIXELS')
```

```
strToolID = IDLitSys_CreateTool("Image Tool",/view_next,initial_data =  
oData, visualization_type='IDLIMAGE')
```

```
oNewTool = oTool->GetByIdentifier(strToolID)
```

```
oCreateVis = oNewTool->GetService("CREATE_VISUALIZATION")
```

```
oParmSet = OBJ_NEW('IDLitParameterSet', $  
                    DESCRIPTION='Image Plane', NAME='Image Plane',$  
                    TYPE='Image', ICON='image')
```

```
; hook up the plane data
```

```
oParmSet->Add, oData, PARAMETER_NAME='IMAGEPIXELS', /  
PRESERVE_LOCATION  
oCommandSet = oCreateVis->CreateVisualization(oParmSet, "IMAGE")  
; amazingly the plane image updates when the volume plane is moved!
```

any suggestions gratefully received,
Anne
