Subject: Re: Creating Panels in iTools
Posted by David Alexander on Wed, 07 Dec 2005 17:46:14 GMT
View Forum Message <> Reply to Message

James Everton wrote:
```
>  To let you know, I decided to keep the panel I made on the right side
>  using "the simple method" and my boss didn't seem to mind that very
>  much. Also, I was able to make the event listener on the draw widgets
>  that I created in my panel. The problem I'm having now is that I'm not
>  sure how to initially put my images into the draw widgets of the panel,
>  nor exactly how to retrieve the images from the parameter set of the
>  iTool to put into the widgets, or the main window when one is clicked.
>
>  In my code right now, I created some temporary images in my initial
>  procedure and put them into the parameter set for the iTool using this
>  syntax, where 'image1' is the image:
>  /---------------------------------
>  oParmSet->Add, OBJ_NEW('IDLitDataIDLImagePixels', image1, $
>                          NAME='Image Planes', $
>                          IDENTIFIER='ImagePixels', $
>                          _EXTRA=_extra), PARAMETER_NAME='IMAGE1'
>  \---------------------------------
>
>  Also, when I created the draw widgets in my panel, I attempted to
>  create an array of pointers to the draw widgets (to be used later when
>  I wanted to
>  /---------------------------------
>  ; Create the draw widget
>  wTV1 = widget_draw(wBase, xsize=128, ysize=128, /BUTTON_EVENTS, $
>                           EVENT_PRO = 'tv1_event')
>  ; Make the array
>  TVs = ptrarr(7, /allocate_heap)
>  TVs[0] = ptr_new(wTV1)
>  \---------------------------------
>  However, when I used this code and did some checks to see what type of
>  variable *TVs[0] was (using the 'help' procedure), I got the response
>  that it was a LONG value, and not anything particularly useful.
>
>  To sum up my questions in the order I should probably implement them:
>  + How do I get identifiers to the draw widgets so I can put them into
my 'TVs' array?
>  + How do I retrieve the images that I put into the parameter set of the
iTool?
>  + How do I use the identifier to a draw widget to then 'TV' the image
from the parameter set?
```

For the draw widgets:

One idea: you could give each draw widget a unique name (using the UNAME keyword when calling WIDGET_DRAW), then in your panel event handler get the UNAME of the widget that sent the event like this:

```
uname=WIDGET_INFO(event.id,/UNAME)
```

then create a CASE block that would do the right thing depending on which draw widget was clicked.

To get the images in the draw widgets, you'll need access to the data manager where you stored the images. You can do this with the UI object that's passed to your panel code as the 2nd argument:

```
oParamSet=oUI->GetByIdentifier("/Data Manager/paramsetname")
```

where 'paramsetname' is the identifier you gave to the parameter set that contains the images. Then you can grab that parameters from the paramset, and then the data. Something like this:

```
oParam=oParamSet->GetByName("image1")
result=oParam->GetData(image1)
```

The LONG value you see when calling HELP on the draw widget is the widget ID of the widget (all widgets are given a unique ID which is a long). To get access to the window in the draw widget, you need to do:

```
WIDGET_CONTROL,wTV,GET_VALUE=win
```

If you're using direct graphics, 'win' will be the window ID (that you would pass to WSET). If you're using object graphics, 'win' will be the object reference to the IDLgrWindow object in the draw widget.

Data Manager/Updating the Image:

When you initially load your data, you would presumably create your own parameter set containing all the image data, and store it in the data manager. This parameter set would not be associated with any particular visualization. Then when you want to update the image data in the image visualization (in the event handler of your panel, for example), you would do something like this:

```
;Get the image data:
oParamSet=oUI->GetByIdentifier("/Data Manager/paramsetname")
oParam=oParamSet->GetByName("image1")

;Get the image visualization.
;If you have more one image visualization in your tool, this may need
```

to be changed.
```
idImage=oTool->FindIdentifiers("*image",/VISUALIZATIONS)
oImage=oTool->GetByIdentifier(idImage)

;Update the image visualization with the new data
oImage->OnDataChangeUpdate,oParam,"IMAGEPIXELS"
```

I think this will automatically redraw the image visualization, but if
it doesn't, you may additionally need to call:

```
oTool->RefreshCurrentWindow
```

You can initially create the image visualization by either basing your
tool on iimage, and calling iimage with the first image as the command
line argument. Or you could use the Insert->Visualization menu item to
create the image. A third (undocumented) way is to call the
CreateVisualization method on the system object
(IDLitSystem::CreateVisualization).