Subject: Re: iTools questions
Posted by David Fanning on Fri, 29 Aug 2003 05:34:35 GMT
View Forum Message <> Reply to Message

Mark Rivers writes:

> I've started to work with the iTools, and they look great. I have a q
> question I wonder if anyone can help me with.
>
> If I create an instance of an iTool, for example, iImage, how can I
> force the iTool to display a new data set from my application (or from
> the IDL command line for that matter)? For example:
> IDL> a = dist(100)
> IDL> iImage, a, ident=id
> IDL> b = dist(256)
>
> Now I would like to display "b" in my iImage. I don't want to use the
> File/Import/IDL Variable, I want to be able to do this via an IDL
> procedure or function call. If I have the object reference for the
> oParmSet that iImage creates as IDL variable I can do it easily.

After my rather lukewarm comments about iTools a week or so ago, I've had to learn more about them (I finally got them installed!). I guess I would have to say I am warming up to them a little bit. At the very least I can appreciate the enormous effort that has gone into the system. It's pretty neat. I don't, however, find them particularly easy to use. For example, I've read the iTool Developer's Guide twice and I *still* don't find any mention whatsoever of a "view" or a "scene". I find that strange (and maybe a bit disturbing) for a system that relies on object graphics. At the very least it leaves me scratching my head about how I would build a tool of my own.

Anyway, I think I know how to solve your problem. A quick look at the iImage code shows the parameter set identified as "Image Parameters", and the image data identified as "ImagePixels". So I proceeded this way:

```
; Set up the iImage tool with an image.

ini_image = LoadData(7)
iImage, ini_image, IDENTIFIER=myImageTool

; Get a reference to the iTool system object:
```

```
  theSystem = _IDLitSys_GetSystem()

  ; The image parameters are stored in the Data Manager.
  ; Get them.

  imageParams = theSystem -> $
    GetByIdentifier("/Data Manager/Image Parameters")

  ; Get the original data out of this object and display
  ; it to be sure you know what you are doing. :-)

  imageObject = imageParams -> $
    GetByIdentifier("Imagepixels")
  ok = imageObject -> GetData(orig_image)
  TV, orig_image

  ; Replace the original image with a new image.

  newImage = LoadData(5)
  ok = imageObject -> SetData(newImage)
```

Walla! The new image axes are even scaled appropriately.
Got to like that! :-)

You might need my LOADDATA program to run this code:

  http://www.dfanning.com/programs/loaddata.pro

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155