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

Subject: Re: iTools questions
Posted by rivers on Fri, 29 Aug 2003 13:20:52 GMT

View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.19b893d31af48ccc9896d9@news.frii.com>...

David, thanks for the reply. I'm getting back into IDL now that the

virtual machine lets me distribute applications to our users. I've
been going down the Python track for a while.

> ; Get a reference to the iTool system object:
>
> theSystem = _IDLitSys_GetSystem()

This is the key, as I discovered late last night. However, this appears to be undocumented routine, and the only reference to the iTool system object that I could find in the documentation says that users should never need to worry about it! I am puzzled by the fact that the iTools are highly object oriented, but when you create one there is no way to get an IDL object reference back for what you've created. You can only get an object identifier (string), which you can't turn into an IDL object reference without an initial object reference to start with.

In any event, this works. It looks to be very powerful. The data we collect are often scans (1-D, 2-D), and by simply putting the scan data into the data manager it should be possible to let users visualize their data any way they want without writing any visualization code at all. Pretty nifty.

Cheers,
Mark

Subject: Re: iTools questions
Posted by David Fanning on Fri, 29 Aug 2003 13:56:40 GMT
View Forum Message <> Reply to Message

Mark Rivers writes:

> I am puzzled by the fact
> that the iTools are highly object oriented, but when you create one
> there is no way to get an IDL object reference back for what you've
> created. You can only get an object identifier (string), which you
> can't turn into an IDL object reference without an initial object
> reference to start with.

Yes, this is a bit strange, although the system relies exclusively on "identifiers" rather than object references to navigate the object hierarchy, which can be a useful notion if there is some way to discover the hierarchy. What I have found strange is that I haven't found any description of the hierarchy anywhere. (And as I say, I've only read the documentation through twice, so it is early in the game.)

I took a guess that the "Data Manager" would be in
the system container from other, oblique references.
Maybe by fumbling around enough we will be able to
piece it together over time. :-)

(This reminds me of my choice of Best Book of Summer.
I thought Gavin Menzies's book, 1421: The Year China
Discovered the World, was terrific. This is how
science should be done! I highly recommend this
fascinating story.)

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

Subject: Re: iTools questions
Posted by Chris[2] on Tue, 02 Sep 2003 18:25:10 GMT

View Forum Message <> Reply to Message

Hi Mark,

Not sure if you already discovered this, but you can also simply use an
overplot:

```
iimage, dist(100)
iimage, dist(256), /overplot
```

This will add the second image to the window. It will appear on top of the
first, so you would need to do an Edit->Ordering->Send to back to make the
first one reappear.

Now, if you want to *replace* the first image, you could simply do:

```
iimage, dist(100)
iimage, dist(256), /view_next
```

The "view_next" keyword places the resulting image into the next available
view in a gridded view layout. Since you only have 1 view, it just replaces
the original image.

If you look up iimage in the IDL Reference Guide, you can read the docs for
the OVERPLOT, VIEW_GRID, VIEW_NEXT, and VIEW_NUMBER keywords.

-Chris Torrence
Research Systems, Inc.

"Mark Rivers" <rivers@cars.uchicago.edu> wrote in message
news:62c206f3.0308281405.3b04c494@posting.google.com...
> 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)
>
...
> Cheers,
> Mark Rivers

Subject: Re: iTools questions
Posted by David Fanning on Tue, 02 Sep 2003 19:19:30 GMT

View Forum Message <> Reply to Message

Chris Torrence writes:

> The "view_next" keyword places the resulting image into the next available
> view in a gridded view layout. Since you only have 1 view, it just replaces
> the original image.
>
> If you look up iimage in the IDL Reference Guide, you can read the docs for
> the OVERPLOT, VIEW_GRID, VIEW_NEXT, and VIEW_NUMBER keywords.

Well, I wouldn't have guessed *that* from reading the
documentation for VEIW_NEXT, but there you go. :-)

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