Subject: iTool Example
Posted by p.sommer on Mon, 13 Dec 2004 17:41:14 GMT
View Forum Message <> Reply to Message

Group,

In a recent thread, there was some discussion about how I used the iTool framework to build a data acquisition (DAQ) client. I was expressing some positive experiences having used the iTools for this project which was challenged by some. At the time, I offered to reform the example so that the user community could take a look at a fully featured, user defined iTool. It is now available for anyone interested:

http://www.rsinc.com/codebank/search.asp?search=category&amp ;product=IDL&catid=43

Now, if I may put my sales hat on, in addition to providing an excellent platform for rapid application development, I also made some mention about making sure the problem fit the tool which was also challenged. In short, what I meant by that statement was to try using an existing iTool [iPlot, iContour, iSurface, iVolume, iImage or iMap] on similar data that one needs to support to make sure the framework can support it. For example, since the iTools are based on IDL's Object Graphics, much of the data needs to reside in physical memory to maintain interactivity. Without going into too much detail, I think you power users know what I mean.

Someone also asked for more detail for why one would want to start using the iTool framework at all for building new tools. My rational is that the framework takes care of a lot of important details like setting up views, operators, manipulators, file I/O, property sheets, annotation, zoom control, undo/redo buffer and now with 6.1, macro recording. It's just very nice to have all this taken care of so I can focus on the elements that make the tool truly unique as apposed to spending hours and hours building up the basic functionality end users have come to expect from a nice piece of software. Okay...enough said. Go easy on me as I am learning right along with you. Just hope the example helps someone, somewhere.

-Paul

Subject: Re: iTool Example
Posted by Robert Barnett on Thu, 27 Jan 2005 04:31:01 GMT
View Forum Message <> Reply to Message

Hi,

I'm currently writing some tools for arranging image, plots and tables
from many different sources including graphics devices (e.g ZBuffer) and
pixeldata (e.g 24 bit image or 8 bit image with a LUT). These sources
are to be laid out over one or several pages. The pages are then
converted into DICOM images and sent to a remote location. The process
must be automatic without requiring user interaction.

Most of my code makes use of object graphics. This is the kind of
project that I thought would be suitable for iTools. However, I found
that iTools is really just geared towards interactive visualisation
(funny that). The closest I have come to iTools is to make use of
widget_propertysheet as a debugging tool and IDLitContainer to hold
objects.

It seems to me that iTools was never designed for medical imaging. This
was probably so because Watsyn was under development by RSI at the time.
  Although the user interface is completely useless to me, I really like
the underlying concepts within iTools. Notably:
* Having Identifiers to access objects easily
* Using data observers/notifiers for event driven programming

Has anyone extended the iTools framework for their own use without
actually using the iTools UI?

p.sommer@comcast.net wrote:

> Group,
>
> In a recent thread, there was some discussion about how I used the
> iTool framework to build a data acquisition (DAQ) client. I was
> expressing some positive experiences having used the iTools for this
> project which was challenged by some. At the time, I offered to reform
> the example so that the user community could take a look at a fully
> featured, user defined iTool. It is now available for anyone
> interested:
>
>
> http://www.rsinc.com/codebank/search.asp?search=category&amp ;product=IDL&catid=43
>
>
> Now, if I may put my sales hat on, in addition to providing an
> excellent platform for rapid application development, I also made some
> mention about making sure the problem fit the tool which was also
> challenged. In short, what I meant by that statement was to try using
> an existing iTool [iPlot, iContour, iSurface, iVolume, iImage or iMap]

> on similar data that one needs to support to make sure the framework
> can support it. For example, since the iTools are based on IDL's
> Object Graphics, much of the data needs to reside in physical memory to
> maintain interactivity. Without going into too much detail, I think
> you power users know what I mean.
>
> Someone also asked for more detail for why one would want to start
> using the iTool framework at all for building new tools. My rational
> is that the framework takes care of a lot of important details like
> setting up views, operators, manipulators, file I/O, property sheets,
> annotation, zoom control, undo/redo buffer and now with 6.1, macro
> recording. It's just very nice to have all this taken care of so I can
> focus on the elements that make the tool truly unique as apposed to
> spending hours and hours building up the basic functionality end users
> have come to expect from a nice piece of software. Okay...enough said.
> Go easy on me as I am learning right along with you. Just hope the
> example helps someone, somewhere.
>
> -Paul
>
>

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223

Subject: Re: iTool Example
Posted by David Fanning on Thu, 27 Jan 2005 05:54:06 GMT
View Forum Message <> Reply to Message

Robert Barnett writes:

> Has anyone extended the iTools framework for their own use without
> actually using the iTools UI?

I have a framework that uses property sheets, messaging, error handling,
etc., etc., really much of the stuff that is done in iTools.
In my library all the widgets have been turned into objects.
The application you write is an object hierarchy. Events and
messages are pretty much synonymous.

You write event handling methods. There is no need to pass
info structures around, etc. The framework is graphics independent,
so you can use object graphics or direct graphics code, whatever
works for you. The best thing about the library is that you
can actually understand it. :-)

Are we talking money, or just interest? :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: iTool Example
Posted by David Fanning on Thu, 27 Jan 2005 05:59:27 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I have a framework that uses property sheets, messaging, error handling,
> etc., etc., really much of the stuff that is done in iTools.
> In my library all the widgets have been turned into objects.
> The application you write is an object hierarchy. Events and
> messages are pretty much synonymous.

Oh, I failed to mention that I have all that annotation
stuff figured out for direct graphics. So you can type
text in a window and immediately drag it around in the
window, etc. Right clicking brings up its Control Panel
(property sheet), where you can control all its properties.
Boxes, ellipses, arrows, text, etc. can be grouped, aligned,
distributed, and saved in an annotation layer. PostScript
output is exactly what you would expect. I've never seen
anything remotely like it in direct graphics. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
