Subject: More Joy of Widgets (was: IDL vs. PV-WAVE)
Posted by Struan Gray on Sat, 14 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Edward S. Meinel, meinel@altair.aero.org writes:
>
> Jonathan Rogness <rogness@NO.sg1.SPAM.cr.usgs.gov> sez:
>>
>> I'd be interested in hearing some of these people speak up, just because
>> I'm curious about how exactly they incorporate widgets into their
>> research.
>
> OK, I'll bite. I started using widgets because I got tired of typing
> lots of commands on the command line. I wrote my own image processing
> widget since no existing program satisfied my needs. Now when I want
> to try a new algorithm, I write a .pro file for the main processing
> and add a couple of lines to the main widget to assign the processing
> to a menu. It has made algorithm development fast and easy.

Ditto here. If I have an idea for something I muck around with the command line at first but pretty early on I start to construct files called something like 'temptest.pro' and if that grows so that it has more than one or two keywords or looks like being something I'll use on lots of different datasets it rarely takes much time to put together a widget for the user interface. My only complaint is that recovering from a bug-induced crash seems much much harder if the xmanager is running (and 4.01 has some nasty xmanager bugs on my PCI Macintosh) so functional routines tend to get fairly thoroughly debugged before I'll call them from a widget.

My widgets tend only to handle interaction with the user, and any routine that actually does something gets compiled as a seperate .pro file so that I can call it from the command line should I want to. So far everything works under the Mac 5.0 pre-release, including the things that were crash-prone under 4.01 such as calling applescripts to manipulate the file system.

My only beef with IDL is that - claims to the contrary notwithstanding - the plotting is not what I consider to be publication quality. IDL lets me analyse and process my data very effectively but I always export to programs like Photoshop and the excellent Mac plotting package Igor when I want to produce a publishable graphic with good-looking annotations and labelling.

IDL 5.0 does offer better graphics than 4.01, along with a consistent colour model and much better ways to interact with plots, but at present it does not offer enough usable tools to stitch together the low level object graphics routines without a lot of work

by the programmer. Similarly, the overall objects framework is well thought out and powerful but the current class library is a bit limited. My feeling is that 5.0 is a big improvement on 4.01 but at present doesn't offer much more than I could have got by buying some cross-platform graphics and user interface libraries for a low level object oriented language like C++. I hope that once RSI have chased down the worst of the bugs they will start to look at how to save the user application development time by providing some high-level object classes for obvious tasks like plotting generalised datasets of various dimensions. Of course, David might beat them to it :-)

Struan

Subject: Re: More Joy of Widgets
Posted by wonko on Sat, 28 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

struan.gray@sljus.lu.se (Struan Gray) wrote:

>> Jonathan Rogness <rogness@NO.sg1.SPAM.cr.usgs.gov> sez:
>>>
>>> I'd be interested in hearing some of these people speak up, just
>>> because I'm curious about how exactly they incorporate widgets into
>>> their research.
[...]

> put together a widget for the user interface. My only complaint is
> that recovering from a bug-induced crash seems much much harder if the
> xmanager is running (and 4.01 has some nasty xmanager bugs on my PCI
> Macintosh) so functional routines tend to get fairly thoroughly
> debugged before I'll call them from a widget.

I only run into problems with the xmanager if it wants to call a routine I didn't define yet. When other errors occur, I just type RETURN and the program usually continues.

My widget programs usually consist of few .pro files. For a program xyz I have XYZ as startup file, a short xyz_init.pro, a xyz_widgets.pro for defining and changing widgets, a xyz_events.pro with all event handling procedures and xyz.pro with general routines. During the programming period I have a STOP button somewhere in my application with an event handler 'stop_event' (defined in xyz_init.pro, not in xyz_events.pro), which just puts me back to the command line via the STOP command. From there it's nice I can inspect all variables, but the greatest adavantage is that I can even re-compile program code and continue.
When a routine crashes, RETURN almost always puts me back in to the main

event loop. Then I go to the command line, fix the bug, recompile the changed file with .COMP, and continue after an RETURN.
Or I can add new widgets, add an event handling routine to xyz_events.pro, recompile it and continue.

It's just annoying that sometimes I get 'Structure nested too deeply' errors, after too many crashes and returns. I hope this will be gone with 5.0, I don't think those errors are my fault.

Alex

--

Alex Schuster Wonko@weird.cologne.de
alex@pet.mpin-koeln.mpg.de
