Subject: Re: IDL 5.2 GUI Builder Tutorial ?
Posted by davidf on Fri, 21 May 1999 07:00:00 GMT

View Forum Message <> Reply to Message

Karl Young (kyoung@itsa.ucsf.edu) writes:

> In my style of always looking for a quick fix I was wondering if anyone
> knew of any kind of tutorial on the GUI builder in IDL 5.2. From some
> brief fiddling it looks reasonably easy to use but I'd like to see a
> couple of simple examples of building an interface and filling in the
> code so's to get started on the right foot. I'd be amazed if David
> didn't have something up his sleeve but I haven't scoured his web page
> very thouroughly lately. Thanks for any tips (including any advice to
> stay away from the GUI builder ...)

The problem with the GUI Builder is that it is just that--a GUI Builder. No one I know builds GUI's just for the fun of it. Most people build applications. The GUI Builder is decidedly NOT an application builder, although I think this is RSI's intention for it, eventually.

So, to give you an example. You open the thing up and build your nice interface. The API is slick, very Visual Basic looking, and quite easy to use. Your expectations are rising. Although it is not *exactly* WYSIWYG, it's pretty darn close and better than what you might have expected if you had tried to build an application builder on your own with IDL's widgets. :-)

Finally, you get something that looks pretty good and your hit the button that generates the actually IDL code. Cool! Although it uses Stein Vidar's-preferred method of a huge big event handler, I've been down this road too and can't fault RSI for making this choice.

But, of course, the code doesn't *do* anything. All the event handlers are stubbed out "shells", there are no positional or keyword parameters defined to get data into the program, no provision for passing information around in the widget program, etc. Oh, oh. Bad news here. You are going to have to learn some widget programming anyway or you aren't going to be able to get any but the very simplest programs to work!

Now, leave aside the handicap of having to work with code you didn't exactly write. We have all been down *that* road and no one wants to be reminded of it.

Forge ahead to actually adding some code so that the
program does something useful.

Hey. Not too bad! Things are cooking now.

But, whoops! You forgot you wanted the user to
be able to apply the FOO procedure, or your $*&^@
boss, who couldn't pour piss out of a boot if the
directions were written on the sole, doesn't like
the interface design, or whatever it is. But you
have to make a design change.

Now you have a bit of a problem.

If you go back to the GUI-Builder, you are going
to lose all that work you just went to adding code
that actually makes your program work, because the
GUI-Builder doesn't know about (or even care about,
if you really want the truth) how many hours you
have spent slaving over a hot computer.

If you make a *new* design, you are going to have
to save the file under a different name, cut and
paste all the useful bits from the old program into
the new one, finally re-name the new program, etc.
Doesn't sound like productive programming time to
me. And how many errors do you think you will
introduce in the cut and paste operations?

Alright. So forget that. You're sticking to the program
that works. But guess what? You can't use the GUI-Builder
now. Because the GUI-Builder builds GUIs and now *you*
are building applications. Something else, entirely.

And here is more bad news. You are going to have to
make your design modifications onto a GUI program
scaffold that was decidedly a compromise to begin with.
That means that it wasn't written from the get-go
with the notion that it might actually have to be
changed. (To my mind this means it was designed by
a computer scientist rather than a scientist who
uses a computer, but never mind about that.) So
now you have to know about widget programming AND
you are saddled with a program you didn't write
and wasn't designed in the way you would have designed
it were you writing it from scratch yourself.

And all of a sudden, the notion of a GUI-Builder,

which seemed like it was going to save you loads of
time and effort, feels like a stone around your
neck and you feel yourself sinking into that
programming quagmire faster and faster.

My suggestion? Pay the $75, get the book, and
learn how to write a widget program the right way
to begin with. Besides lining my pocket, you'll
thank me every way to Sunday every time you want
to add a new feature to your program. :-)

Cheers,

David

P.S. The above description of the GUI-Builder is my
personal (and probably biased) opinion and has absolutely
nothing whatsoever to do with anything else. Including,
some might argue, the facts of the case. I strongly
recommend you try the GUI-Builder yourself and form
your own opinions. :-)

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