Subject: Re: General widget programming questions
Posted by davidf on Fri, 12 Jan 2001 15:01:28 GMT
View Forum Message <> Reply to Message

Jason P. Meyers (jpm7934@cis.rit.edu) writes:

> We have recently started writing some widget based programs in our
> IDL programming class and it has gotten me to thinking about a few
> issues I would like to better understand before I get too far into our
> "final" project. For starters, I have read the relevant chapters in
> Dave Fanning's second edition. So far, that has been the best help in
> understanding widget based programming. (I have even used some of
> Dave's info to correct misinformation presented in class!)

Be careful with that book. It has been known
to turn around and bite you when you least
expect it. :-(

> I like the ability to create user defined event structures. In my
> current homework project, I have started using these to pass information
> from one widget to another. I basically pack the information I want to
> send to another widget in an event structure and give it a descriptive
> name and then send it along to the appropriate widget. So far, this
> seems like a reasonable thing to do. However, I got to thinking about
> potential problems. First, what happens if other events (i.e. user
> generated) are waiting in the queue? I assume they will get acted on
> first and this may or may not cause a problem for the original event
> handler which sent an event to a fellow widget. If this is the case, is
> there any way to give a particular event a higher priority over other
> events?

There is no way to give events priority. *All* events generated
normally, or sent via SEND_EVENT, are placed on the queue and
are handled in the order received. This is ordinarily a good
thing, and is especially advantageous to people just getting
started in widget programming, because it prevents an awful
lot of problems. For example, there is never any problem
with the info structure not being where it is suppose to be.
If each event handler checks it out, then checks it in before
exiting, then each event can find it and be assured the information
in it is current.

But as you get more advanced in widget programming it sooner
or later occurs to you that with the big bucks you are being
paid you ought to be a little more clever than to be a slave
to SEND_EVENT. And you think to yourself, "My God, I'm just
going to call that event handler directly. After all, it is
nothing more than another IDL procedure or function."

And you can do that. I've done it. Lot's of times. But it's kind of like putting the short side of the board against the fence on the table saw. You better be damn careful you know what you are doing. And for goodness sake, don't stand directly behind the saw!

By calling the event handler directly, you obviously short-circuit the event queue. Sometimes this means you have to check the info structure in before you make the call. Sometimes you have to do other things. (When things get really complicated, you might want to put the info structure in a pointer and pass that around, since then you don't have to worry about checking out/checking in.)

But I will say that it is quite easy to get too cute with all this. I think the simpler you can make event handling, the better off you are. Good ol' XManager, one-at-a-time event handling is good enough for 99% of your applications, I think.

> I haven't seen a good example or even an explanation of how and/or why it would be useful to use the Event.Handler value which is stored in all event structures. I thought that I read somewhere in Dave's book that he was going to address this but, I didn't see any references to using it in the various chapters on widget programming. (Dave, am I blind or did I get confused with something I may have scanned in the IDL online references?)

The handler field is used quite a lot in building compound widgets. I intended to add a chapter on compound widgets in the book, but I got so sick and tired of reading the darn thing that my will collapsed before I could get it written. It's still on my list for the "next" book. :-)

> As always, I appreciate any and all insight people might have. Thanks in advance for advice/answers that come flowing. Finally, if Dave (or anyone else) has recommendations for additional high quality educational/tutorial information that picks up where Dave leaves off in his book, please let us all know.

Ronn Kling has a nice book out that explains quite a number of useful widget and programming techniques. You might have a look at his web page:

http://www.rkling.com/

Cheers,

David
--
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

Subject: Re: General widget programming questions
Posted by Pavel A. Romashkin on Fri, 12 Jan 2001 16:55:55 GMT

View Forum Message <> Reply to Message

I will just say that in those few lines of code I ever wrote, I never
had a problem with events queue slowing me down. On today's computers,
anyway. I guess, I can't click fast enough - about 120 clicks per minute
is all I can dish out, and the CPU apparently keeps up at 500 MHz. The
only slowdown I ever see is the graphics card, and not when I have tons
of events but when those OG 3D transforms are performed on more than
100K objects at the same time. Then I can see performance degradation.
But I have not had to short-cut the old Xmanager. Although I want to
read it out of curiosity :-)

Cheers,
Pavel

P.S. Let's just say I am not advanced enough. David, you'll have to
expand your widget and object sections do move me forward and develop
the need in direct calls to event handlers :-)

David Fanning wrote:
>
> Jason P. Meyers (jpm7934@cis.rit.edu) writes:
>
>> We have recently started writing some widget based programs in our
>> IDL programming class and it has gotten me to thinking about a few
>> issues I would like to better understand before I get too far into our
>> "final" project. For starters, I have read the relevant chapters in
>> Dave Fanning's second edition. So far, that has been the best help in
>> understanding widget based programming. (I have even used some of
>> Dave's info to correct misinformation presented in class!)
>
> Be careful with that book. It has been known
> to turn around and bite you when you least
> expect it. :-(
>
>> I like the ability to create user defined event structures. In my
>> current homework project, I have started using these to pass information

>> from one widget to another. I basically pack the information I want to
>> send to another widget in an event structure and give it a descriptive
>> name and then send it along to the appropriate widget. So far, this
>> seems like a reasonable thing to do. However, I got to thinking about
>> potential problems. First, what happens if other events (i.e. user
>> generated) are waiting in the queue? I assume they will get acted on
>> first and this may or may not cause a problem for the original event
>> handler which sent an event to a fellow widget. If this is the case, is
>> there any way to give a particular event a higher priority over other
>> events?
>
> There is no way to give events priority. *All* events generated
> normally, or sent via SEND_EVENT, are placed on the queue and
> are handled in the order received. This is ordinarily a good
> thing, and is especially advantageous to people just getting
> started in widget programming, because it prevents an awful
> lot of problems. For example, there is never any problem
> with the info structure not being where it is suppose to be.
> If each event handler checks it out, then checks it in before
> exiting, then each event can find it and be assured the information
> in it is current.
>
> But as you get more advanced in widget programming it sooner
> or later occurs to you that with the big bucks you are being
> paid you ought to be a little more clever than to be a slave
> to SEND_EVENT. And you think to yourself, "My God, I'm just
> going to call that event handler directly. After all, it is
> nothing more than another IDL procedure or function."
>
> And you can do that. I've done it. Lot's of times. But it's
> kind of like putting the short side of the board against the
> fence on the table saw. You better be damn careful you know
> what you are doing. And for goodness sake, don't stand directly
> behind the saw!
>
> By calling the event handler directly, you obviously short-circuit
> the event queue. Sometimes this means you have to check the info
> structure in before you make the call. Sometimes you have to do
> other things. (When things get really complicated, you might want
> to put the info structure in a pointer and pass that around,
> since then you don't have to worry about checking out/checking in.)
>
> But I will say that it is quite easy to get too cute with
> all this. I think the simpler you can make event handling,
> the better off you are. Good ol' XManager, one-at-a-time
> event handling is good enough for 99% of your applications,
> I think.
>

>> I haven't seen a good example or even an explanation of how and/or why
>> it would be useful to use the Event.Handler value which is stored in all
>> event structures. I thought that I read somewhere in Dave's book that
>> he was going to address this but, I didn't see any references to using
>> it in the various chapters on widget programming. (Dave, am I blind or
>> did I get confused with something I may have scanned in the IDL online
>> references?)
>
> The handler field is used quite a lot in building compound
> widgets. I intended to add a chapter on compound widgets in
> the book, but I got so sick and tired of reading the darn thing
> that my will collapsed before I could get it written. It's still
> on my list for the "next" book. :-)
>
>> As always, I appreciate any and all insight people might have. Thanks
>> in advance for advice/answers that come flowing. Finally, if Dave (or
>> anyone else) has recommendations for additional high quality
>> educational/tutorial information that picks up where Dave leaves off in
>> his book, please let us all know.
>
> Ronn Kling has a nice book out that explains quite a number
> of useful widget and programming techniques. You might have
> a look at his web page:
>
>   http://www.rkling.com/
>
> Cheers,
>
> David
> --
> 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

Subject: Re: General widget programming questions
Posted by Paul van Delst on Fri, 12 Jan 2001 17:18:26 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" wrote:
>

<snip>

>
> Cheers,

> Pavel
>
> P.S. Let's just say......

Hang on a minute... "Let's just say" is a copyrighted coyote postscript prefix. Isn't it?

Apparently, DF's significant influence on the IDL community is not just encouraging people to walk-the-walk, but talk-the-talk too.

:o)

paulv

p.s. Let's just say I thought it was funny at the time. Sigh.

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301) 763-8000 x7274 There shallow draughts intoxicate the brain,
Fax: (301) 763-8545 And drinking largely sobers us again.
Email: pvandelst@ncep.noaa.gov Alexander Pope.

Subject: Re: General widget programming questions
Posted by davidf on Fri, 12 Jan 2001 17:46:37 GMT

View Forum Message <> Reply to Message

Pavel A. Romashkin (pavel.romashkin@noaa.gov) writes:

> P.S. Let's just say I am not advanced enough. David, you'll have to
> expand your widget and object sections do move me forward and develop
> the need in direct calls to event handlers :-)

One of the places I frequently use direct calls to event handlers is in my compound widget objects. As you know, most events in compound widgets are handled by an internal event handler. But occasionally you want to send the internal event (after extensive remodelling, usually) to some other event handler.

Normally, the internal event handler is associated with the top-level base of the compound widget (identified in the event handler as event.handler, by the way, NOT event.top). One way to send events on is to write the internal event handler as a function. You get the internal event into the function, re-arrange it, maybe

add some fields to it, etc. and pass it along as the result
of the function, where it merrily makes its way up
the widget hierarchy.

This works great so long as the user doesn't want to
assign an event handler procedure or function to the
compound widget. (See any RSI-supplied compound widget,
for example.) It doesn't work so great when you would
like to re-direct the event somewhere else.

Since I want my compound widget to look and feel as much
as possible like a simple widget, I usually define
Event_Pro and Event_Func keywords for them. What I do,
then, in the event handler method function (remember, I always
write these as objects these days) is, when I am finished
processing the event, and have the new event packaged up
the way I want it, is call the specified event handler
procedure or function directly. (I use Call_Procedure
or Call_Function, of course, but you get the idea.)

If I *do* make the call directly, I turn the event
structure into a 0 and return that as the result of
the method function. The event is "swallowed". If the user
hasn't specified an event handler, then I just return
the event structure as the result of the method function, and
the event bubbles up the hierarchy. This way I can have
my cake and eat it too. :-)

Cheers,

David

--
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

Subject: Re: General widget programming questions
Posted by davidf on Fri, 12 Jan 2001 17:52:58 GMT

View Forum Message <> Reply to Message

Paul van Delst (pvandelst@ncep.noaa.gov) writes:

> Hang on a minute... "Let's just say" is a copyrighted
> coyote postscript prefix. Isn't it?

Let's just say like most things having to do with Coyote,
this one was stolen from a person whose work and humor
I admired over in rec.wooodworking. :-)

Cheers,

David

--
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

Subject: Re: General widget programming questions
Posted by Pavel A. Romashkin on Fri, 12 Jan 2001 18:00:27 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> "Pavel A. Romashkin" wrote:
>> P.S. Let's just say......
>
> Hang on a minute... "Let's just say" is a copyrighted coyote postscript prefix. Isn't it?
>
> Apparently, DF's significant influence on the IDL community is not just encouraging people to
> walk-the-walk, but talk-the-talk too.
>
> :o)
>
> paulv
>
> p.s. Let's just say I thought it was funny at the time. Sigh.

Oops. David is gonna sue me now for copyright violation. Why did you
have to make this notice in public, Paul?
I think you're right. *Let's just say* that the entire newsgroup is
saturated with cheerful spirit of the Coyote :-)

Cheers,
Pavel

Subject: Re: General widget programming questions

Posted by Jason P. Meyers on Sat, 13 Jan 2001 06:29:34 GMT
View Forum Message <> Reply to Message

Thanks for all of the comments and feedback (everyone!) In case other people were trying to use the link below, there is a minor typo, it should be:

http://www.rlkling.com/

Thanks again,
Jason Meyers
PhD Student, Center for Imaging Science
Rochester Institute of Technology
jpm7934@rit.edu

David Fanning wrote:

> < Stuff Deleted >
>
> Ronn Kling has a nice book out that explains quite a number
> of useful widget and programming techniques. You might have
> a look at his web page:
>
>   http://www.rkling.com/