Subject: Re: How can I use the /multiple keyword with WIDGET_LIST
Posted by Martin Schultz on Thu, 21 Dec 2000 12:21:20 GMT
View Forum Message <> Reply to Message

Nicholas Keat wrote:

>
> I have created a widget based application for processing CT scanner
> image files that allows the user to select images from a list, and
> analyse them for certain image quality parameters.
>
> What I would like to be able to do is to select, say, three images from
> a list of ten for analysis. From the IDL documentation (ha!) it seems
> that I can use the '/multiple' keyword with widget_list to allow
> multiple selections, which it seems to do. Running my app with a
> /multiple added to my widget_list statement allows me to press 'Shift'
> and 'Ctrl' to highlight more than one element on the list. The problem
> is that the events generated by making multiple selections seem to be
> exactly the same as those created from making single selections. How do
> I tell if the user has pressed 'Shift' or 'Ctrl' to make an extended
> selection? Is it possible without extended hacks to track key presses?
> I've looked at the usual source of IDL explanations (www.dfanning.com)
> without finding any references... Surely someone else must have had to
> do this before!
>
> Nick
> --
> Nicholas Keat
> Senior Physicist - ImPACT Group, St George's Hospital
> London SW17 0QT
> Tel +44 (0)20 8725 3366   http://www.impactscan.org

hmmmm. If David comes over his hangover, he might be better suited to
answer this question. But as far as I can say, there is no need to
have different events. If you act on a "select" event, you would just
inquire about the currently selected items (Widget_Info) no matter if
any of these had been selected before or not. If you want to manage
your image display windows and only (re)display images that are
"freshly" selected, you will have to keep track of the selected images
yourself and store the previous selection somewhere -- a good place
seems the UValue field of the TLB widget (or for principalists, it's
first child).

Hope this helps,
Martin

--
[[[[[[[[[

[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

Subject: Re: How can I use the /multiple keyword with WIDGET_LIST
Posted by davidf on Thu, 21 Dec 2000 15:16:03 GMT
View Forum Message <> Reply to Message

Nicholas Keat (nick@impactscan.org) writes:

> I have created a widget based application for processing CT scanner
> image files that allows the user to select images from a list, and
> analyse them for certain image quality parameters.
>
> What I would like to be able to do is to select, say, three images from
> a list of ten for analysis. From the IDL documentation (ha!) it seems
> that I can use the '/multiple' keyword with widget_list to allow
> multiple selections, which it seems to do. Running my app with a
> /multiple added to my widget_list statement allows me to press 'Shift'
> and 'Ctrl' to highlight more than one element on the list. The problem
> is that the events generated by making multiple selections seem to be
> exactly the same as those created from making single selections. How do
> I tell if the user has pressed 'Shift' or 'Ctrl' to make an extended
> selection? Is it possible without extended hacks to track key presses?
> I've looked at the usual source of IDL explanations (www.dfanning.com)
> without finding any references... Surely someone else must have had to
> do this before!

Why is everyone talking so loudly this morning... :-(

I think Nick is right here. The IDL documentation
doesn't help you out much. Especially if you are reading
the Widget_List documentation, which doesn't help you
out at all, as far as I can see.

The problem with list widgets is that you get a selection
event through your list event handler each and every time
you make a selection. But to find out what those selections
are, you have to use Widget_Info with the list widget
identifier and the LIST_SELECT keyword set. For multiple
selections you will get a vector of selection indices
as the result of this function. And then, of course,
you will have to devise some way to turn the indices into
the actual selection values. (Normally the values are

stored in the User Value of the list widget just for this
purpose.)

All this is well and good, but as an interface design
it's fairly hopeless. What you want is some indication
from the user that you are suppose to *do* something
with the selection. With a single selection, of course,
you can make the list widget disappear and go act on
the selection. With multiple selections, you almost
always have to have some kind of button around that
says "Go Do It", or something like that, because otherwise
you have no idea when the user is *finished* making
selections.

So, since you have to have a button, and you need some
way to store the values, and you probably would like
to manage the list events in an event handler separate
from your other event handlers (since most events it
generates are probably safe to ignore), and ...

Well, this list of requirements is just about the perfect
definition for why you want to build a compound widget.

I've done something similar for the droplist widget, which
I tend to use more often than list widgets. You can see the
result in FSC_Droplist, a program on my web page. I wrote
it so that I can get the actual "value" of the droplist
selection in an event structure that comes from the droplist.
I was tired of always having to look elsewhere for the value.

And while I was at it, I fixed a couple of other things that
annoyed me with droplists. Like the spacing of the text in
the droplist, etc. I wrote this compound widget as an object,
but you wouldn't have to do this. I just did it because I
wouldn't even *think* of writing a compound widget as anything
but an object these days. But the code is really simple.
And I think you could easily apply the principles to a
CW_MULTILIST program.

http://www.dfanning.com/programs/cw_droplist.pro

I think this could probably be knocked out in the couple
of hours left until the next Christmas party. At least it
could if I could just find the Nutmeg...

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: How can I use the /multiple keyword with WIDGET_LIST
Posted by Alex Schuster on Thu, 21 Dec 2000 15:58:42 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

> Why is everyone talking so loudly this morning... :-(

> http://www.dfanning.com/programs/cw_droplist.pro

Close! Actually, it's http://www.dfanning.com/programs/fsc_droplist.pro

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

Subject: Re: How can I use the /multiple keyword with WIDGET_LIST
Posted by davidf on Thu, 21 Dec 2000 16:07:48 GMT

View Forum Message <> Reply to Message

Alex Schuster (alex@pet.mpin-koeln.mpg.de) writes:

> Close! Actually, it's http://www.dfanning.com/programs/fsc_droplist.pro

Thanks, Alex. I just found my glasses, stuck up there
with the Mistletoe. :-)

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/

Subject: Re: How can I use the /multiple keyword with WIDGET_LIST
Posted by John-David T. Smith on Thu, 21 Dec 2000 18:30:20 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Nicholas Keat (nick@impactscan.org) writes:
>
>> I have created a widget based application for processing CT scanner
>> image files that allows the user to select images from a list, and
>> analyse them for certain image quality parameters.
>>
>> What I would like to be able to do is to select, say, three images from
>> a list of ten for analysis. From the IDL documentation (ha!) it seems
>> that I can use the '/multiple' keyword with widget_list to allow
>> multiple selections, which it seems to do. Running my app with a
>> /multiple added to my widget_list statement allows me to press 'Shift'
>> and 'Ctrl' to highlight more than one element on the list. The problem
>> is that the events generated by making multiple selections seem to be
>> exactly the same as those created from making single selections. How do
>> I tell if the user has pressed 'Shift' or 'Ctrl' to make an extended
>> selection? Is it possible without extended hacks to track key presses?
>> I've looked at the usual source of IDL explanations (www.dfanning.com)
>> without finding any references... Surely someone else must have had to
>> do this before!
>
> Why is everyone talking so loudly this morning... :-(
>
> I think Nick is right here. The IDL documentation
> doesn't help you out much. Especially if you are reading
> the Widget_List documentation, which doesn't help you
> out at all, as far as I can see.
>
> The problem with list widgets is that you get a selection
> event through your list event handler each and every time
> you make a selection. But to find out what those selections
> *are*, you have to use Widget_Info with the list widget
> identifier and the LIST_SELECT keyword set. For multiple
> selections you will get a vector of selection indices
> as the result of this function. And then, of course,
> you will have to devise some way to turn the indices into
> the actual selection values. (Normally the values are
> stored in the User Value of the list widget just for this
> purpose.)
>

> All this is well and good, but as an interface design
> it's fairly hopeless. What you want is some indication
> from the user that you are suppose to *do* something
> with the selection. With a single selection, of course,
> you can make the list widget disappear and go act on
> the selection. With multiple selections, you almost
> always have to have some kind of button around that
> says "Go Do It", or something like that, because otherwise
> you have no idea when the user is *finished* making
> selections.

Unless you don't care, and always want to update in real time based on the current selection. This is similar to the always-update vs. update-when-dropped slider widget controversy which has raged for decades. If I wanted immediate updating for newly selected images, I'd just catch the WIDGET_LIST events, interrogate the widget with widget_info(wL,/LIST_SELECT), and update my images based on that.

JD