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
