Subject: Re: fsc_field double
Posted by David Fanning on Thu, 05 Feb 2009 14:53:59 GMT
View Forum Message <> Reply to Message

AndiBiffar@googlemail.com writes:

> i am using david fannings fsc_field. I initialize the field in the
> following way
>
> param_text = fsc_field(textbase,title=param_labels(i), value = double
> (0.0),Decimal=2,LabelSize= 55,/noedit,xsize=9,$
>   uname = param_labels(i), /highlight,/frame,/
> cr_only,uvalue=i,event_pro='setparam')
>
> if a user selects a certain option in my gui the fields should become
> editable and set to default values defined as double default = double
> (1000) e.g.
>
> textID = widget_info(event.top,find_by_uname=param_labels(i))
> widget_control, textID, set_value=default, editable=editable
>
> However even though i initialized the field with a double idl
> complains that it is unable to convert double to string.
>
> WIDGET_CONTROL: String expression required in this context: DEFAULT
>
> Since the field is not set to a different value before i don't
> understand what is going wrong here

FSC_Field has a great deal more functionality than CW_Field. To
achieve all that functionality, I had to write the program as
an object. But then I couldn't get anyone to use it because
objects scared the daylights out of them. So, it got wrapped up
in a way that made it possible for most people to use it just
like they used CW_Field. You got most of the benefits without
having to learn anything new.

Anyway, this is the root of your problem. You are trying to
short-circuit some of the added functionality by the way you
are trying to set the value. The problem can be solved by saving
the underlying field object when you create the field, and by
using that to set your value:

```
 param_text = fsc_field(textbase,title=param_labels(I), %
    value = double(0.0),Decimal=2,LabelSize= 55,/noedit,xsize=9,$
    uname = param_labels(I), /highlight,/frame,$
   /cr_only,uvalue=I,event_pro='setparam', OBJECT=parma_text_object)
```

Then, later:

```
info.parma_text_object -> SetProperty, Value=Double(1000), NoEdit=0
```

Now, the problem becomes how to find the object reference. I put
everything I need to run my program (which in this case includes
the object reference) into an info structure, and pass that around
to my event handlers. You don't seem to be doing that here. So,
how can you find the object reference?

Well, it is stored in the program, in the user value of the
first child of the top-level base of this compound widget,
which happens to be the label widget.

Now, as it happens, you are using the UNAME of the widget
to search for it.

```
textID = widget_info(event.top,find_by_uname=param_labels(I))
```

This may have been a programming mistake on my part (since I
don't use UNAME for anything generally), but that UNAME is being
passed along to the text widget. Had you assigned the name with
the NAME keyword (instead of UNAME) in the definition statement,
the same code would have returned the identifier of the base
at the top of the compound widget hierarchy.

If you fnnd the text widget, then you are looking for its sibling.
If you find the base widget, then you are looking for its first child.

It turns out it is MUCH easier to look for a child, rather
than a sibling that was born *before* you. (Please don't
ask me why.)

So, here is my recommendation. Change the UNAME keyword to
NAME in the definition of the field:

```
param_text = fsc_field(textbase,title=param_labels(I), %
    value = double(0.0),Decimal=2,LabelSize= 55,/noedit,xsize=9,$
    NAME = param_labels(I), /highlight,/frame,$
  /cr_only,uvalue=I,event_pro='setparam)
```

Then, find the object reference like this:

```
baseID = widget_info(event.top,find_by_uname=param_labels(I))
labelID = Widget_Info(baseID, /CHILD)
Widget_Control, labelID, Get_UValue=fieldObj
fieldObj -> SetProperty, Value=default, NoEdit=0
```

Problem solved. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")