
Subject: Re: object widgets

Posted by [David Fanning](#) on Fri, 11 Jan 2002 14:57:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

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

> Or has David been cut off from his Dedicated Satellite Link? :)

No, no. But I did find my New Year's Resolutions list.

The first item is this:

1. Get a life! Give those poor folks on the Newsgroup a break!

I'm exercising restraint this year. :-)

There hasn't been too much written down about object widgets yet, although I understand that may be changing soon. But basically, these objects are written with some kind of GUI method to create the interface and the self object is the former info structure that holds the program information required to run the program.

The trick, if there is one, is to get widget events into object methods where they can access the self information. I usually do this by placing in the UValue of those widgets which will generate events I want to respond to a "message" structure. This anonymous structure contains an object reference (typically, but not always to the "self") and the name of an event handler method to be called on the object.

Here is part of a GUI method from a program I wrote for a client:

```
fileID = Widget_Button(menuID, Value='File')
button = Widget_Button(fileID, Value='Acquire New Image...', $
    UValue={object:self, method:'AcquireNewImage'})
button = Widget_Button(fileID, Value='Open Existing Image...', $
    UValue={object:self, method:'OpenExistingImage'})
button = Widget_Button(fileID, Value='Restore Session', $
    UValue={object:self, method:'RestoreSession'}, /Separator)
button = Widget_Button(fileID, Value='Save Session', $
    UValue={object:self, method:'SaveSession'})
button = Widget_Button(fileID, Value='Quit', $
    UValue={object:self, method:'QuitProgram'}, /Separator)
```

The event handler for *all* the events generated by the program is very simple. Here it is:

```
PRO COMPANY_ANALYSIS_EVENT, event
Widget_Control, event.id, Get_UValue=message
Call_Method, message.method, message.object, event
END
```

The event handler simply gets the "message" stored in the UValue of the widget causing the event, and calls the appropriate method for that object, using Call_Method.

The event handler methods, then, are written **exactly** like your former widget event handlers. Here is an example of one:

```
PRO COMPANY_ANALYSIS::SaveSession, event
```

```
; This method saves the current session.
```

```
; Error handling.
```

```
Catch, theError
```

```
IF theError NE 0 THEN BEGIN
```

```
  Catch, /Cancel
```

```
  ok = Error_Message(Traceback = self.debug)
```

```
  RETURN
```

```
ENDIF
```

```
GetName:
```

```
filename = Dialog_Pickfile(Title='Save Session As...', $
```

```
  File='petrographic_session.pws', $
```

```
  Filter='*.pws', Path=self.directory, /Write)
```

```
IF filename EQ "" THEN RETURN
```

```
checkFile = FindFile(filename, Count=count)
```

```
IF count EQ 1 THEN BEGIN
```

```
  answer = Dialog_Message('File exists. OK to overwrite?', /Question)
```

```
  IF StrUpCase(answer) EQ 'YES' THEN Save, self, File=filename ELSE $
```

```
    GoTo, GetName
```

```
ENDIF ELSE Save, self, File=filename
```

```
END
```

Please ignore the GOTO statement. I think it was 3:30 in the morning when I wrote this particular piece of lousy code. :-)

That's about it. Pretty simple. But extraordinarily powerful.

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155
