
Subject: Re: Object Widgets

Posted by [Mark Hadfield](#) on Wed, 10 Nov 1999 08:00:00 GMT

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

A few more items of feedback on your widget classes, Struan:

1. In your widget cleanup routine it is a good idea to check that the object is valid before trying to destroy it, i.e.

```
pro SLFoWid_Cleanup, myID
```

```
    widget_control, myID, get_uvalue=myObjRef
    if obj_valid(myObjRef) then begin
        myObjRef -> GetProperty, no_block=no_block
        if no_block eq 1 then obj_destroy, myObjRef $
        else myObjRef -> cleanup
    endif
```

```
end ; pro SLFoWid_Cleanup
```

This is advisable because it is possible for the object to have been destroyed (by OBJ_DESTROY, or HEAP_GC) behind XMANAGER's back.

2. Similarly it is a good idea in the object cleanup routine to check if the widget hierarchy is still valid and, if it is, destroy it, i.e.

```
pro SLFoWid::Cleanup
```

```
    print, 'SLFoWid::Cleanup'
    print, ' widget ID: ', self.myWidID
    print, ' object ID: ', self
```

```
    if widget_info(self.myWidID, /VALID_ID) then widget_control, self.myWidID, /DESTROY
```

```
end ; pro SLFoWid::Cleanup
```

3. There is a problem in heap cleanup for blocking widgets. If I create a new widget with

```
o = obj_new('SLFow_minimal')
```

then hit the quit button, the heap is cleaned up. But if I do the same with

```
o = obj_new('SLFow_minimal', /BLOCK)
```

then the 'SLFow_minimal' object is left on the heap. The problem, as you have noted in comments in your code, is that for a blocking widget the

procedure that handles the 'quit' event is called from the XMANAGER event loop, which is called from inside the Init method, and you can't destroy an object from inside it's own Init method. One solution is for SLFow_minimal::Init to check whether it has been called with the BLOCK keyword set. If it has, then it needs to leave out the call to Xmanage, and leave this up to the user. This is an unfortunate complication. I guess the other solution is for 'SLFow_minimal' to prevent the BLOCK keyword from being passed to 'SLFoWid' so that the application always runs non-blocking. (Perhaps you'd already thought of all this.)

Anyway, thanks very much for publishing this stuff. It provides a framework for writing much cleaner widget applications.

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand
