Subject: Re: call_method
Posted by Mark Hadfield on Fri, 07 Sep 2001 04:31:29 GMT
View Forum Message <> Reply to Message

From: "David Fanning" <david@dfanning.com>
> P.S. Let's just say that if you wanted to enlighten us
> as to *why* you do this all the time, I would be all
> ears. :-)

Well, not all the time.

OK, to get specific, I have a general purpose graphics object called MGHgrGraph, based on IDLgrView. MGHgrGraph has a method called NewAtom that creates a graphics atom and attaches it to the graphics tree. Creating an object graphics plot usually consists of creating an MGHgrGraph then calling NewAtom several times. The function form of the NewAtom method returns a reference to the atom; the procedure form doesn't. Most of the time I don't need the reference, so I call the procedure form; sometimes I do need the reference, so I call the function form. (The code is a bit more readable when the procedure form is used, I think.)

The function definition contains all the code to implement NewAtom. The procedure form is a wrapper that looks like this

```
pro MGHgrGraph::NewAtom, P1, P2, P3, P4, RESULT=result, _EXTRA=extra

   compile_opt IDL2

   case n_params() of
      0: result = self->NewAtom( _EXTRA=extra )
      1: result = self->NewAtom( P1, _EXTRA=extra )
      2: result = self->NewAtom( P1, P2, _EXTRA=extra )
      3: result = self->NewAtom( P1, P2, P3, _EXTRA=extra )
      4: result = self->NewAtom( P1, P2, P3, P4, _EXTRA=extra )
   endcase

end
```

The RESULT keyword is for when I change my mind & decide I *do* want the object reference after all.

> It must have something to do with all those sheep. :-)

I don't farm sheep but do I farm olive trees. They have the advantage that they don't move around so much, but on the other hand they aren't as warm & cuddly.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--