
Subject: Re: At Last! A Subsitute for CW_Field.
Posted by [davidf](#) on Thu, 18 Nov 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Martin Schultz (m218003@modell3.dkrz.de) provides a few suggestions for improvement, then asks this:

- > Third: why is this not an object? ;-) Indeed it would make sense to provide
- > the functionality of this thing as object, so you could for example extend
- > the "heart" of it (the validation routine) to allow for hex numbers or
- > number ranges, etc.

Yes, indeed, it definitely *should* be written as an object. But my original goal was to create a drop-in replacement for CW_FIELD. I thought that was an hour job, and it turned into something approaching three days and two VERY late nights. Have you ever tried to decipher RSI-supplied code. :-)

But you are correct, that validation routine is the heart of the matter and it would be a whole lot easier to extend it if it was an object method.

- > Then again: with an object you would require two files:
- > coyote_field.pro
- > and coyote_ofield__define.pro
- > so people wouldn't be able to get it running ;-)

I woke up this morning thinking about this. (Do you have any idea how depressing it is to be this much in love with a **programming** language, for God's sake!) Anyway, I think the thing to do is to leave the user interface alone, so it **can** be a drop-in replacement, but turn the heart of the program into an object. The outside world could get access (should they want or need it) to the object nature via a GetGuts keyword. (I'd probably spend an hour thinking of a better name, but that's what comes to mind at the moment.)

It just all required more effort than I was ready to give at 2:30 AM. :-)

- > And this brings up the point how to best link objects and ignorant users.
- > Should one provide a default object in the widget function and allow for
- > a predefined object to be passed as a substitute? Hence,
- > wID = coyote_field(...)
- > would use the coyote_ofield object with the functionality as present,
- > whereas

```
> wID = coyote_field(...,object=obj_new("hex_field"))  
> would pass responsibilities on to this other thing.
```

No, I think the point of objects is that they will behave in a particular way unless you override that behavior by writing replacement methods, for example. You must just supply the user with opportunity and clear instructions for how to do so.

```
> Cheerios, (I love them and haven't found them over here)
```

What, no Cheerios!? My family and I are currently in the processes of planning a summer trip to Germany. We may have to re-think it with this news. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155
