M. Katz writes:

> One main source of heartburn in objects comes from passing parameters
> through _Extra and _Ref_Extra. I use them all the time and am still
> confused by them.

I'm not too unhappy with keyword inheritance. In general,
I think it works quite well. I've finally figured out
_EXTRA and _REF_EXTRA, I think. When I am expecting results
back from keywords (for example, all GetProperty methods), I
always use _REF_EXTRA on the method definition line, but
I *always* use _EXTRA when passing the extra keywords along
to the superclass method. Somehow, someway, this *always*
works! I've got object templates now that help me remember
these simple things, so when I'm building a new object, I'm
more likely to get it right. (Amazing how much templates
helped with memory leakage problems, too.)

A bigger problem initially was passing keywords along
that weren't defined anywhere and not knowing about it.
So we use _STRICT_EXTRA on the "atom" class object, which
is inherited by everyone, to trap unhandled keywords. In other
words, if a keyword gets to this level and it hasn't been
defined yet, someone used the wrong keyword (or misspelled
it, more likely).

Occasionally, I have more trouble with this, but not too
often, and most of the time it is easily solved by just
defining a few more keywords.

> All this talk of objects and pointers is making me think of the IDL
> vs. Matlab threads that were going on 2 months back. For all the pros
> and cons on the syntax and speed, how great is it that IDL allows us
> to use object and pointers! to dream these great abstractions (and
> spend our weekends debugging.)

Well, Amen to this, I guess... :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting

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