Subject: Re: _extra and call_method
Posted by David Fanning on Tue, 01 Feb 2005 15:40:16 GMT
View Forum Message <> Reply to Message

Ben Tupper writes:

> Just got to thinking on this some more. Here's one item I thought you might be
> willing to share your thoughts on: in the case of multiple inheritances does
> each keyword get checked in for each generation of inheritance and if they do,
> does it matter?
>
> For example...
>
> PRO APPENDAGE__DEFINE, class
> class = {APPENDAGE, isJointed:0}
> END
>
> PRO LEG__DEFINE, class
> class = {ARM, INHERITS APPENDAGE, hasOpposingThumb: 0}
> END
>
>
> So, if I use the generic keyword checking for LEG does "isJointed" get checked
> twice? Propbably it doesn't matter a hoot if they do get checked twice.
>
>
>
> PRO LEG::SetProperty, _EXTRA = extra
>
> ... do that neat keyword checking thing here ...
>
> self->APPENDAGE::SetProperty, _Extra = extra
> END

The generic methods short-circuit the normal chaining of
keywords through SetProperty methods (could be a bad thing,
I guess, in the wrong hands) by setting the structure field
of self directly. If the field is there, it gets set. End of
story.

Of course, not all properties act this way. Sometimes when you
set one property, you are obliged to set others, etc. This
mechanism doesn't help with that, and, in fact, probably shouldn't
be used if you are doing things like that.

I use the generic methods mostly for quick and dirty objects,
or for when I am in the "development as thinking" stage of a
project. For the real-deal objects, I prefer to define the keywords,

since then I have the chance to document them. Makes it easier for
the end-user. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
