Subject: Inherited keywords. Was: Arg_Present
Posted by J.D. Smith on Tue, 05 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Update on the Inherited keyword return variable problem...

The official word from the RSI tech support is as follows:

>   I'm sorry to say that there does not appear to be a good
> workaround.
> I spoke with some of the developers, and all they suggested was
> to NOT override GetProperty and instead make a method called
> MyGetProperty that handled all of your properites...
>   That is not an acceptable solution in my book. So, I have
> submitted
> a feature request to change _EXTRA to pass variables by reference
> instead of by value. When the _EXTRA keyword was added to IDL,
> most keywords were input, so passing by value was not a problem.
> Now, with widgets and objects many keywords are output values,
> so it is probably time for a change.
>   Again, I am sorry that we don't have a better solution for you at
> the
> present time.   Thank you for pointing out this problem with our
> product.
>
> Best Regards,
>
> Jeremy Gebben
> Technical Support Engineer
> Research Systems, Inc.
> support@rsinc.com

in response to my message:

>
> I think I've figured out the real source of the problem...inherited
> keywords can only be passed by *value*, presumably because they are
> encoded in a structure.   This means arg_present() *is* functioning
> correctly when it doesn't consider _EXTRA keywords to be return
> variables -- they in fact are not. This makes it difficult to "chain
> up
> the class tree" on methods which return things through keywords...
> e.g.
> GetProperty. For example, if I made my own subclass to IDLgrModel,
> and
> overrode the GetProperty method, I'd have to explicitly include all of
> IDLgrModel's GetProperty keywords in the declaration of my class's
> GetProperty method if I wanted to be able to get properties

> established
> in the superclass. I could not say, e.g.:
>
> ```
>      pro subclass::GetProperty, mykey1=mk,_EXTRA=e
>           self->IDLgrModel::GetProperty(_EXTRA=e)
>           if arg_present(mk) then mk=self.somevalue
>      end
> ```
>
> and hope to have valid values returned through _EXTRA keywords. Since
> explicity including all the keywords of the superclass defeats the OOP
> methodology, I have changed my GetProperty method so that it returns a
> structure containing the relevant data, which I create and append to
> on the fly, chaining up the class tree as necessary.
>
> Perhaps there is some resolution of this that I'm missing, but,
> apparently, creating a subclass of any of the Object Graphics classes
> would make it difficult to override and chain GetProperty (and perhaps
> other methods?) in a straightforward fashion.
>
> Thanks,
>
> JD