Subject: Re: is it possible to pass named variables through _extra
Posted by Martin Schultz on Wed, 27 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Mirko Vukovic wrote:
>
> so, this is the story,
>
> I have a call to contour burried deep in some subroutine. And sometimes
> I would like to do a
> contour,foo,path_xy=path_xy
> i.e., get the coordinates of the contours. Using _extra does not seem
> to work. If in the top routine I specify
> path_xy=var
> where var is undefined, I get an error message,
> If I define var to be an array, or something like that IDL complains
> that the expression must be a named variable.
>
> Any ideas?
>
> tia,
>

The problem must be buried somewhere in your subroutine as you say. Here is how IDL's contour behaves depending on the parameters you give it and whether you pass a "named variable" :

```
a=fltarr(10,10)
x=findgen(10)*!PI/18.
for i=0,9 do a(*,i) = 2*sin(!PI/(i+1)-x)


; #1 : (see manual) works !
print,'  #1 : ---------------------------------- '
contour,a,color=1,/follow,path_info=info1,path_xy=path1
help,info1,path1,/stru

; #2 : (no path) works !
print,'  #2 : ---------------------------------- '
contour,a,color=1,/follow,path_info=info2
help,info2,/stru

; #3 : (no info) works !
print,'  #3 : ---------------------------------- '
contour,a,color=1,/follow,path_xy=path3
help,path3,/stru

; #4 : (no variable parameter) ERROR !
```

```
print,'  #4 : ---------------------------------- '
contour,a,color=1,/follow,path_xy=1

; #5 : (no variable parameter) ERROR !
print,'  #5 : ---------------------------------- '
contour,a,color=1,/follow,path_info=1
end
```

This program will actually stop at step 4 with the error message you quoted.
You don't need to define infoN or pathN beforehand, but you cannot pass a value (as in steps 4 and 5). This is also what _EXTRA does, since there is no way that _EXTRA can know what you want to extract from it later it only passes by value and can therefore not work.

Good luck,
Martin

 ---------- ----------Dr.
Martin Schultz e-mail: mgs@io.harvard.edu
Center for Earth&Planet. Sci.
Harvard University
Pierce Hall phone: (617)-496-8318
29 Oxford St fax : (617)-495-5192
Cambridge, MA-02138
 ---------- ----------
check out http://www-as.harvard.edu/people/staff/mgs/idl !
 ---------- ----------

Subject: Re: is it possible to pass named variables through _extra
Posted by J.D. Smith on Mon, 01 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> Mirko Vukovic wrote:
>>
>> so, this is the story,
>>
>> I have a call to contour burried deep in some subroutine. And sometimes
>> I would like to do a
>> contour,foo,path_xy=path_xy
>> i.e., get the coordinates of the contours. Using _extra does not seem
>> to work. If in the top routine I specify
>> path_xy=var
>> where var is undefined, I get an error message,
>> If I define var to be an array, or something like that IDL complains

>> that the expression must be a named variable.
>>
>> Any ideas?
>>
>> tia,
>>
>
> The problem must be buried somewhere in your subroutine as you say. Here
> is how IDL's contour behaves depending on the parameters you give it and
> whether you pass a "named variable" :
>
> a=fltarr(10,10)
> x=findgen(10)*!PI/18.
> for i=0,9 do a(*,i) = 2*sin(!PI/(i+1)-x)
>
> ; #1 : (see manual) works !
> print,' #1 : ---------- '
> contour,a,color=1,/follow,path_info=info1,path_xy=path1
> help,info1,path1,/stru
>
> ; #2 : (no path) works !
> print,' #2 : ---------- '
> contour,a,color=1,/follow,path_info=info2
> help,info2,/stru
>
> ; #3 : (no info) works !
> print,' #3 : ---------- '
> contour,a,color=1,/follow,path_xy=path3
> help,path3,/stru
>
> ; #4 : (no variable parameter) ERROR !
> print,' #4 : ---------- '
> contour,a,color=1,/follow,path_xy=1
>
> ; #5 : (no variable parameter) ERROR !
> print,' #5 : ---------- '
> contour,a,color=1,/follow,path_info=1
> end
>
> This program will actually stop at step 4 with the error message you
> quoted.
> You don't need to define infoN or pathN beforehand, but you cannot pass
> a value (as in steps 4 and 5). This is also what _EXTRA does, since
> there is no way that _EXTRA can know what you want to extract from it
> later it only passes by value and can therefore not work.
>
> Good luck,
> Martin

>
>

here is a re-post of a discussion of the problem from a few weeks ago,
the gist of which is that _EXTRA keywords are currently passed by value
only, but that a fix may be in the offing.

JD

I wrote:

>
> 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

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
