
Subject: keyword inheritance in object graphics
Posted by [dg86](#) on Wed, 02 Sep 2015 14:58:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Folks,

I'm trying to write a wrapper routine that creates an object plot, overlays some arrows on the plot, and then returns the plot object. I'd like to use keyword inheritance to provide options for both `plot()` and `arrow()` within my routine. The idea is something like

```
function arrowplot, x, y, _extra=extra
  pl = plot(x, y, _extra = extra)
  ndx = [0, 1]
  ar = arrow(x[ndx], y[ndx], _extra=extra)
  return, pl
end
```

I assumed that the `plot(...)` and `arrow(...)` functions would each use the keywords they understand and ignore the rest. Instead, the `arrow(...)` function throws an error whenever it encounters an unfamiliar keyword in the `extra` structure.

Is this correct behavior, or is it a bug? I've tested this in IDL 8.4.

Many thanks,

David

Subject: Re: keyword inheritance in object graphics
Posted by [penteado](#) on Wed, 02 Sep 2015 19:50:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, September 2, 2015 at 7:58:32 AM UTC-7, David Grier wrote:

> Is this correct behavior, or is it a bug? I've tested this
> in IDL 8.4.

As I understand it, this is intended behavior, because unlike iTools (which ignore any keywords they do not know, even if not passed with `_extra`), Function Graphics (not Object Graphics) have keyword checking code in them (the `VerifyProperty` method). However, you can use the /undocumented keyword, which would disable this checking:

```
p=plot(/test,/somethingthatdoesnotexist,/undocumented)
```

Subject: Re: keyword inheritance in object graphics
Posted by [dg86](#) on Wed, 02 Sep 2015 20:20:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, September 2, 2015 at 3:50:16 PM UTC-4, Paulo Penteado wrote:
> On Wednesday, September 2, 2015 at 7:58:32 AM UTC-7, David Grier wrote:
>> Is this correct behavior, or is it a bug? I've tested this
>> in IDL 8.4.
>
> As I understand it, this is intended behavior, because unlike iTools (which ignore any keywords they do not know, even if not passed with `_extra`), Function Graphics (not Object Graphics) have keyword checking code in them (the `VerifyProperty` method). However, you can use the `/undocumented` keyword, which would disable this checking:
>
> `p=plot(/test,/somethingthatdoesnotexist,/undocumented)`

Dear Paulo,

Thanks very much for this extremely helpful fix.

I could not be more jazzed that my code is going to rely on the "undocumented" boolean keyword, which indeed is undocumented. :)

Subject: Re: keyword inheritance in object graphics
Posted by [wlandsman](#) on Wed, 02 Sep 2015 20:21:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Neat! Looking in the `lib/graphics` directory I see that several new graphics functions (e.g. `LEGEND`, `COLORBAR`) have the `/UnDocumented` keyword.

But I have to ask. Is the `/undocumented` keyword documented somewhere? --Wayne

On Wednesday, September 2, 2015 at 3:50:16 PM UTC-4, Paulo Penteado wrote:
> On Wednesday, September 2, 2015 at 7:58:32 AM UTC-7, David Grier wrote:
>> Is this correct behavior, or is it a bug? I've tested this
>> in IDL 8.4.
>
> As I understand it, this is intended behavior, because unlike iTools (which ignore any keywords they do not know, even if not passed with `_extra`), Function Graphics (not Object Graphics) have keyword checking code in them (the `VerifyProperty` method). However, you can use the `/undocumented` keyword, which would disable this checking:
>
> `p=plot(/test,/somethingthatdoesnotexist,/undocumented)`
