Subject: keyword params in HISTOGRAM
Posted by Mark Fardal on Wed, 20 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hi,

I'm sure this is a stupid question, but I can't figure it out.
Normally when you pass keyword parameters, you can pass something
that's undefined, right? As in this example

```
  pro junk1, a, test=test
  help,test,t
  ;junk2, a, test=test  ; either one of
  junk2, a, test=t      ; these works
  return
  end

  pro junk2, a, test=test

  a = 3.
  return
  end

IDL> junk1, a
 TEST           UNDEFINED = <Undefined>
 T              UNDEFINED = <Undefined>
IDL>
```

However, when I try to pass HISTOGRAM a keyword parameter that's
undefined, I get an error message. The routine here is just a
wrapper function for HISTOGRAM, so I want to pass a number of
keyword parameters through to it.

```
  function histomake, array, binsize=binsize, xpoints=xpoints, $
   input=input, min=min, max=max, omin=omin, omax=omax, reverse=reverse
  [...stuff deleted...]

  n = histogram(array, binsize=binsize, min=min, max=max, $
              omin=omin, omax=omax, reverse=reverse)

  [...stuff deleted...]
  return
  end

IDL> h = histomake(array, bins=10, xpoints=xpoints)
% HISTOGRAM: Variable is undefined: MAX.
% Execution halted at:  HISTOMAKE         52
  /users1/casa/fardal/comp/idl/histomake.pro
```

% $MAIN$

Why is the behavior different here?

Thanks,
Mark

Subject: Re: keyword params in HISTOGRAM
Posted by thompson on Tue, 03 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Mark Fardal wrote:
>
> Hi,
>
> I'm sure this is a stupid question, but I can't figure it out.
> Normally when you pass keyword parameters, you can pass something
> that's undefined, right? As in this example
>
(rest deleted)

Although the example you give isn't optimal, it is correct that you can pass undefined keywords to procedures. This is because inside procedures you can check to see if keywords are passed with either the N_ELEMENTS function or KEYWORD_SET. Thus, passing an undefined keyword is equivalent to not passing it at all, which is a useful thing IMHO. Unfortunately, this doesn't work with built-in IDL routines, which I find a very inconvenient and silly restriction.

Bill Thompson

Subject: Re: keyword params in HISTOGRAM
Posted by Robert Moss on Tue, 03 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Mark Fardal wrote:
>
> Hi,
>
> I'm sure this is a stupid question, but I can't figure it out.
> Normally when you pass keyword parameters, you can pass something
> that's undefined, right? As in this example
>
> pro junk1, a, test=test
> help,test,t

```
>    ;junk2, a, test=test  ; either one of
>    junk2, a, test=t      ; these works
>    return
>    end
>
>    pro junk2, a, test=test
>
>    a = 3.
>    return
>    end
>
>  IDL> junk1, a
>   TEST           UNDEFINED = <Undefined>
>   T              UNDEFINED = <Undefined>
>  IDL>
>
>  However, when I try to pass HISTOGRAM a keyword parameter that's
>  undefined, I get an error message.  The routine here is just a
>  wrapper function for HISTOGRAM, so I want to pass a number of
>  keyword parameters through to it.
>
>    function histomake, array, binsize=binsize, xpoints=xpoints, $
>     input=input, min=min, max=max, omin=omin, omax=omax, reverse=reverse
>    [...stuff deleted...]
>
>    n = histogram(array, binsize=binsize, min=min, max=max, $
>                  omin=omin, omax=omax, reverse=reverse)
>
>    [...stuff deleted...]
>    return
>    end
>
>  IDL> h = histomake(array, bins=10, xpoints=xpoints)
>  % HISTOGRAM: Variable is undefined: MAX.
>  % Execution halted at:  HISTOMAKE         52
>    /users1/casa/fardal/comp/idl/histomake.pro
>  %                      $MAIN$
>
>  Why is the behavior different here?
>
>  Thanks,
>  Mark
```

Your junk example "works" because you are not trying to actually do anything with the test variable in the junk2 routine... hence no error occurs.

The HISTOGRAM routine _is_ trying to do something with its keywords, so

you get an error. I see two reasonable solutions:

1) If you are not using the min, max, etc keywords in the histomake function itself, rather than making them explicit keywords, use keyword inheritance (the _EXTRA keyword) to pass them to HISTOGRAM. (See EXTRA in the online help).

2) Use the CALL_FUNCTION (faster) or EXECUTE (slower) method to call HISTOGRAM inside your makeohist function so that it is called with only the relevant and proper keywords.

At a guess, I'd say that solution 1) is the way to go in general...
keyword inheritance is a Good Thing.

--
Robert M. Moss, Ph.D. - mossrm@texaco.com - FAX (713)954-6911