Subject: Re: Top 10 IDL Requests
Posted by kashyap on Tue, 25 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Ah, the specific reason I came up with that request was that I was trying
to have CURVEFIT pass strange keywords down to a curve-defining function
(which is called via CALL_PROCEDURE), and failing. In that particular
case, the fix is simple of course (edit CURVEFIT).

Also, I was not aware that the function definition does not *have* to
have the "_extra" -- not quite what the help file says:

 You can pass keyword parameters to called routines
 by value by adding the formal keyword parameter
 "_EXTRA" (note the underscore character) to the
 definition of your routine. ...

now that I am primed to it, I notice that it goes on to say:

 When the keyword _EXTRA appears in a procedure or
 function call, its argument is either a structure
 containing additional keyword/value pairs which
 are inserted into the argument list ...

It's not tremendously clear that putting the _extra in a function
definition is optional. I suppose it helps only if the function
in question calls other functions or subroutines to which it must
pass keywords.

Thanks for the clarifications,
Vinay

In article <on1z0hrdn4.fsf@cow.physics.wisc.edu>,
Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote:
> davidf@dfanning.com (David Fanning) writes:
>
>> Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:
>>
>>> davidf@dfanning.com (David Fanning) writes:
>>>> Vinay L. Kashyap (kashyap@head-cfa.harvard.edu) writes:
>>>> > 2. _EXTRA
>>>> >
>>>> > Please consider having all built-in commands accept _EXTRA as a keyword.
>>>>
>>>> Uh, this is the way it works. :-)
>>>
>>> Uh, not quite. There are some built in commands that don't accept any
>>> keywords at all. The _EXTRA keyword doesn't work for them, *even* if

>>> the value passed is empty!
>>>
>>> Why is this important? Makes it a pain to write a wrapper procedure
>>> or function.
>>
>> Alright, I must be obtuse today, but I can't figure out why
>> it would be hard to write wrapper routines for commands that
>> don't take keywords. Surely in writing the wrapper you give
>> at least *some* thought to what keywords you might expect
>> to be passed. Adding an _Extra to such a command seems
>> excessively anal at the very least, and certainly unnecessary. :-)
>>
>> And what commands did you have in mind? I've never encountered
>> a built-in command that didn't accept this keyword mechanism.
>
> Hrmm. The moment I need to find an example, and I can't find it.
> Arghh. An example of a built-in command that doesn't take keywords is
> EMPTY, but I agree that it's a pretty lame example. I actually would
> hope that *all* procedures and functions could be called with _EXTRA,
> whether or not they actually accept keywords. For, example, this
> statement
>

```
>   call_procedure, 'EMPTY', _EXTRA=null
```

>
> will fail no matter what, even if null is an undefined variable.
> Shouldn't IDL be smart enough to test whether the _EXTRA value is
> undefined before it crashes?
>
> I'm always looking for ways to avoid special cases in wrapper
> routines. Real world examples of such unavoidable abominations are
> given below.
>
> Craig
>
>

```
> (from XFWINDOW in XFWINDOW_CALL_PROCEDURE)
>   sz = size(key)
>   if sz(sz(0)+1) EQ 8 then begin ;; Keywords are present
>      xfwindow_rekey, key
>      case n_args of
>          0: call_procedure, cmd, _extra=key
>          1: call_procedure, cmd, x0, _extra=key
>          2: call_procedure, cmd, x0, x1, _extra=key
>          3: call_procedure, cmd, x0, x1, x2, _extra=key
>          4: call_procedure, cmd, x0, x1, x2, x3, _extra=key
>          5: call_procedure, cmd, x0, x1, x2, x3, x4, _extra=key
>      endcase
>   endif else begin              ;; No keywords are present
>      case n_args of
>         0: call_procedure, cmd
>         1: call_procedure, cmd, x0
>         2: call_procedure, cmd, x0, x1
>         3: call_procedure, cmd, x0, x1, x2
>         4: call_procedure, cmd, x0, x1, x2, x3
>         5: call_procedure, cmd, x0, x1, x2, x3, x4
>      endcase
>   endelse
>
> (from MPFIT in MPFIT_CALL)
>   if proc then begin
>      if n_params() EQ 3 then begin
>         if n_elements(extra) GT 0 then $
>           call_procedure, fcn, x, f, fjac, _EXTRA=extra $
>         else $
>           call_procedure, fcn, x, f, fjac
>      endif else begin
>         if n_elements(extra) GT 0 then $
>           call_procedure, fcn, x, f, _EXTRA=extra $
>         else $
>           call_procedure, fcn, x, f
>      endelse
>   endif else begin
>      if n_params() EQ 3 then begin
>         if n_elements(extra) GT 0 then $
>           f = call_function(fcn, x, fjac, _EXTRA=extra) $
>         else $
>           f = call_function(fcn, x, fjac)
>      endif else begin
>         if n_elements(extra) GT 0 then $
>           f = call_function(fcn, x, _EXTRA=extra) $
>         else $
>           f = call_function(fcn, x)
>      endelse
>   endelse
>
> --
>  --------------------------------------------------------- --------------
> Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>  --------------------------------------------------------- --------------
```

--

kashyap@head-cfa.harvard.edu   617 495 7173 [CfA/P-146] 617 496 7173 [F]