
Subject: wrapper routines

Posted by [thompson](#) on Thu, 29 Apr 1993 21:03:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

A while back there was a discussion about "wrapper" routines, i.e. routines that provide additional functionality to standard IDL built-in routines. It started because I posted a routine (PLOT_DROP) which wrapped around IDL's PLOT command. I don't want to get into the details of the program, but part of the code looked like the following

```
COMMAND = "PLOT,X,Y,/NODATA"
IF N_ELEMENTS(XTITLE) EQ 1 THEN COMMAND = COMMAND + ",XTITLE=XTITLE"
IF N_ELEMENTS(YTITLE) EQ 1 THEN COMMAND = COMMAND + ",YTITLE=YTITLE"
IF N_ELEMENTS(TITLE) EQ 1 THEN COMMAND = COMMAND + ",TITLE=TITLE"
IF N_ELEMENTS(XTYPE) EQ 1 THEN COMMAND = COMMAND + ",XTYPE=XTYPE"
IF N_ELEMENTS(YTYPE) EQ 1 THEN COMMAND = COMMAND + ",YTYPE=YTYPE"
IF N_ELEMENTS(XRANGE) EQ 2 THEN COMMAND = COMMAND + ",XRANGE=XRANGE"
IF N_ELEMENTS(YRANGE) EQ 2 THEN COMMAND = COMMAND + ",YRANGE=YRANGE"
TEST = EXECUTE(COMMAND)
```

In other words, PLOT_DROP took many of the same keywords that PLOT did. However, I couldn't just say in the routine

```
PLOT,X,Y,/NODATA,XTITLE=XTITLE,YTITLE=YTITLE,TITLE=TITLE,XTYPE=XTYPE, $
YTYPE=YTYPE,XRANGE=XRANGE,YRANGE=YRANGE
```

because the user may not have passed all the keywords to the wrapper routine.

However, if the wrapper routine instead works by calling another IDL ".pro" procedure, then I don't have to worry about it. For instance, if I had another routine, say PLOT_MYDATA, that was wrapped around PLOT_DROP, then I can simply put the command

```
PLOT_DROP,XTITLE=XTITLE,YTITLE=YTITLE,TITLE=TITLE,XTYPE=XTYPE, $
YTYPE=YTYPE,XRANGE=XRANGE,YRANGE=YRANGE, <arguments>
```

in PLOT_MYDATA without worrying about whether all these keyword values were actually defined or not. An IDL procedure can't distinguish between a keyword that wasn't passed and a keyword that was passed with an undefined value.

It sure would be nice if IDL built-ins worked the same way!

Bill Thompson
