Subject: Re: passing string parameter
Posted by Tim Patterson on Thu, 17 Oct 1996 07:00:00 GMT
View Forum Message <> Reply to Message

daytime visitor ssg login wrote:

>
> Hello, I want to write a small help procedure to display breif summaries
> of home made commands.
>
> IDL> helpme, command_name_string
>
> Is there a way to pass command_name_string to the procedure helpme
> without quotes?
>
> George McCabe, george@replica.gsfc.nasa.gov

if the command is for example, sortnames, I don't think
it's possible to pass it into a procedure as

```
 IDL> helpme, sortnames
```

(as you'vbe obviously discovered! :)

You have to use

```
 IDL> helpme, 'sortnames'
```

Or do it in two stages

```
 IDL> sortnames='sortnames'
 IDL> helpme, sortnames
```

You could use this second method by having a file that
initialises the helpme tool. This file could define
string variables containing the command names e.g.

```
 ;; list of commands
 sortnames = 'sortnames'
 printnames = 'printnames'
 anothercommand = 'anothercommand'
```

and so on

Then when you start IDL, the user enters

```
 IDL > @helpme_init
```

to define all the variables and thereafter he can just use the helpme system as

```
 IDL> helpme, printnames
```

and so on. Not elegant, but it should work ok.

Maybe somebody will come up with a better idea! Be interesting to see

  Tim

--

        http://condor.lpl.arizona.edu/~tim/

Subject: Re: passing string parameter
Posted by Peter Mason on Fri, 18 Oct 1996 07:00:00 GMT

View Forum Message <> Reply to Message

On Wed, 16 Oct 1996, daytime visitor ssg login wrote:

> Hello, I want to write a small help procedure to display breif summaries
> of home made commands.
>
> IDL> helpme, command_name_string
>
> Is there a way to pass command_name_string to the procedure helpme
> without quotes?

If the people using your programs only use IDL to run applications - i.e., don't use IDL's command-line functionality, then you might consider running a little "shell" program to catch certain errors.

I've included an example "shell" program at the end of this posting. It reads a string from the console, calls EXECUTE() to execute it, and traps & fixes "undefined variable" errors by turning "variables" into string constants.
Because it uses EXECUTE(), it doesn't have all of the functionality of the (proper) IDL command line. e.g., You can't run executive commands (like ".run") or bring up online help from it. You're also not supposed to be able to define new variables (e.g., a=100) in it, but this seems to work (at least, with full IDL on DEC OSF).
I'm sure that any IDL command-line jockey will consider it to be a truly disgusting hack, but here it is anyway.

Peter Mason

```
pro shell

; This will stop execute() reporting errors, and will
; also catch ^D etc
catch,e &if (e ne 0) then goto,here

go=1 &s=''
while(go) do begin
here:
  try=0
  read,s,prom='IDL 2> '
  t=strupcase(strtrim(s,2))
  if (t eq 'EXIT') or (t eq 'QUIT') then go=0 $
  else if (t eq '?') then begin
    print,'Sorry, you can''t activate online help from here' &goto,here
  endif else begin
there:
    if (execute(s) ne 1) then begin     ;some error in execute()
      if (!error eq -128) then begin    ;undefined var - handle it!
        try=try+1 &if (try gt 0) then goto,here      ;only try to fix once
        j=rstrpos(s,',')        ;just before problem varname in user's command
        if (j le 0) then goto,here      ;? - just back to prompt
        t=strmid(s,j+1,strlen(s)-j-1)   ;the problem varname
        s=strmid(s,0,j+1)+'"'+strtrim(t,2)+'"'  ;stringify the "varname"
        goto,there                      ;NOW try the execute
      endif else begin  ;some other error - just report & back to prompt
        print,!err_string &print,!syserr_string &goto,here
      endelse
    endif
  endelse
endwhile
return      ;you might want EXIT here instead
end
```

Subject: RE: passing string parameter
Posted by davidf on Sun, 20 Oct 1996 07:00:00 GMT

View Forum Message <> Reply to Message

George McCabe <george@replica.gsfc.nasa.gov> writes:

> I want to write a small help procedure to display brief summaries
> of home made commands.

> IDL> helpme, command_name_string

> Is there a way to pass command_name_string to the procedure helpme
> without quotes?

If I understand this question correctly, I think you want to be able to say:

```
HELPME, myCommand
```

And have your HELPME procedure interrpret "myCommand" as a string. I can't think of any way to do this.

But I am sensitive to the problem. Let me suggest an alternative approach. I define a HELP keyword for all my programs, so that if I can't remember how to use them, I call them with the HELP keyword set. If the HELP keyword is set, they print out a short helpful message about how to use the program and return. Here is an example of a little program named DOIT:

```
PRO DOIT, data, HELP=help
IF KEYWORD_SET(help) THEN BEGIN
   PRINT, 'DOIT, data'
   RETURN
ENDIF
PLOT, data
END
```

I picked up this little trick from Ray Sterner at Johns Hopkins Applied Physics Lab, an excellent IDL programmer who always writes programs that are *fully* documented.

Yours,

David

--
David Fanning, Ph.D.
Phone: 970-221-0438
Fax: 970-221-4728
E-Mail: davidf@fortnet.org