Subject: Re: string definition question
Posted by Paul Van Delst[1] on Tue, 14 Jan 2003 19:28:03 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:
>
> On Tue, 14 Jan 2003 10:55:24 -0500, Paul van Delst
> <paul.vandelst@noaa.gov> wrote:
>
>> Hello there,
>>
>> Although I should probably know the answer to this, since all my IDL reference books have
>> been borrowed, hope you don't mind me asking here.
>>
>> I'm a bit anal about argument checking in IDL. After establishing that the correct number
>> of arguments has been passed using:
>>
>> n_arguments = 1
>> IF (N_PARAMS() LT n_arguments) THEN $
>> MESSAGE, 'Invalid number of arguments.', $
>> /NONAME, /NOPRINT
>>
>>
>> My standard method for checking string arguments (like filenames to read) is something
>> like:
>>
>> IF (N_ELEMENTS(FileNAME) EQ 0) THEN BEGIN
>> MESSAGE, 'Input FileNAME argument not defined!', $
>> /NONAME, /NOPRINT
>>
>> IF (STRLEN(FileNAME) EQ 0) THEN $
>> MESSAGE, 'Input FileNAME argument not defined!', $
>> /NONAME, /NOPRINT
>>
>> If I pass a zero-length string, e.g. FileNAME='', the N_ELEMENTS() test passes so I always
>> test for a non-zero string length (it's happened). If I combine the two tests using AND,
>> then if the variable is undefined, the STRLEN() function generates an errors (since its
>> argument must be defined).
>>
> Hi Paul,
>
> I can't think how to get away from the two step test. Usually the
> ARG_PRESENT() test is reserved for optional output keywords. I would
> stick to N_ELEMENTS() and N_PARAMS()
>
> You could squish the two statements into one - but there are still two
> tests involved.
>

```
>   If (n_elements(FileName) NE 0) Then $
>      If (StrLen(FileName[0]) EQ 0) Then $
>      Print, "Ain't nuttin here, Paul"  Else $
>      Print, "Here's the string" + fileNAME[0] Else $
>      Print, "fileNAME argument is required, ya' know"
```

Here was my solution:

```
FUNCTION Valid_String, Input_String
  IF ( N_ELEMENTS( Input_String ) EQ 0 ) THEN RETURN, 0
  IF ( STRLEN( Input_String ) EQ 0 ) THEN RETURN, 0
  RETURN, 1
END
```

Use of the KEYWORD_SET() function (as suggested by Mika) would reduce this to a single line. I would rather imbed the use of KEYWORD_SET() for this purpose in the same routine above so later readers of the software (including, or especially, me) don't get confused since the string to test may be a regular, non-keyword argument.

Anyway....in the routine where I want to test the filename string:

```
PRO testybits, FileNAME

  CATCH, Error_Status
  IF ( Error_Status NE 0 ) THEN BEGIN
    CATCH, /CANCEL
    MESSAGE, !ERROR_STATE.MSG, /CONTINUE
    IF ( N_ELEMENTS( FileID ) NE 0 ) THEN FREE_LUN, FileID
    RETURN
  ENDIF

  IF ( Valid_String( FileNAME ) EQ 0 ) THEN $
    MESSAGE, 'Input FileNAME argument not defined!', $
            /NONAME, /NOPRINT

  PRINT, FileNAME

  CATCH, /CANCEL

END
```

paulv

p.s. I guess I should really call it PVD_Valid_String. ha bloody ha. :o)

--
Paul van Delst

CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545