
Subject: Re: string definition question

Posted by [btupper](#) on Tue, 14 Jan 2003 18:13:41 GMT

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

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"
```

Ben
