Subject: Re: variable types
Posted by David Foster on Mon, 23 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> David R. Klassen wrote:
>>
>> I'm in the process of writing a quick and dirty program to
>> go through a text file and grab the lines that are ordered
>> pairs of numbers and plot them. The problem comes with the
>> fact that some of these data files have a line or two of
>> text at the top.
>>
>> When I read each line of the file I parse it along white-spaces
>> or tabs uisng the PARTS function (no problem so far). Then I
>> assign the first two parts to my x and y arrays. The problem
>> is that if the line read was a line of text, the parts can not
>> be converted from string type to float type (it turns out that
>> a string '   13.456' *can* be turned into a float=13.456).
>>
>> My question: is there a way to test the variable type before
>> I make the assignments?
>>
>
> yes, you will probably have to read in your line as string (it is really
> a pity that the '$' format code does not work during input !), then you
> can either go David's way to catch the error or test for a number
> yourself:
>   readf,s [,format='(A)']
>   test = strpos('0123456789.+-',strmid(strtrim(s,1),0,1)) ge 0
>
> Fortran data files sometimes use a line format like
>   N   x1 x2 x3 ...   xN, so you need to extract the first number
> before you can read the others. In these cases, you should also read
> the line into a string, [test the first character for a number] and
> then extract the numbers using READS.
>

Martin -

If you do a lot of text parsing you may want to try out my
GET_TOKEN.PRO routine that reads data of BYTE, INT, LONG, FLOAT or
DOUBLE type from a string, maintaining a pointer into the string
to allow for sequential parsing of data. The code is fairly ugly
but it works really well. For your data it would look like:

```
for i = 0, n_elements(lines) - 1 do begin
```

```
 readf, unit, string, format='(a)'
 p = 0
 val1 = get_token(string, p, /flt, error_value='ERROR')
 if (strtrim(val1,2) ne 'ERROR') then begin
  ; p = p + 1    ; SEE NOTE BELOW (**)
  array(i,0) = val1
  val2 = get_token(string, p, /flt, error_value='ERROR')
  array(i,1) = val2       ; Maybe check this value too
 endif
endfor
```

(**) If the numbers are separated by non-whitespace character(s), you
will need to increment the pointer appropriately.

You can download GET_TOKEN.PRO by anonymous FTP:

 bial8.ucsd.edu : pub/software/idl/share/idl_share.tar.gz

This includes many other routines as well, so don't extract into !PATH !

Hope this helps.

Dave
--

```
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~
    David S. Foster         Univ. of California, San Diego
     Programmer/Analyst     Brain Image Analysis Laboratory
     foster@bial1.ucsd.edu  Department of Psychiatry
     (619) 622-5892         8950 Via La Jolla Drive, Suite 2240
                            La Jolla, CA  92037
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~
```