Subject: Re: Reading in data question
Posted by David Fanning on Fri, 17 Apr 2009 05:04:19 GMT
View Forum Message <> Reply to Message

tomandwilltamu08@gmail.com writes:

> Hi IDL gurus,
>
> I am having a hard time thinking of how to read in the following
> data... Its an ascii file. It has hundreds of sections with a title,
> then a table of numbers, then another string with a different table of
> a different size like so:
>
> One name
> 1 42 3.14 Blah
> 2 77 4.13 String
>
> Another name
> 1 11 1.34 String
> 2 22 1.43 Blah
> 3 33 3.41 String
>
> Third name
> 1 44 1.23 Something
> 2 55 2.34 String
> 3 66 3.45 String
> 4 77 4.56 String
> 5 88 5.67 String
>
> ..... and there are hundreds of these . The number of columns is
> fixed, but the number of rows is variable.
>
> What would be a good way to read this in in IDL? Is there a good way
> that I could read them in one by one? I obviously can't specify the
> number of rows of each array a priori, but I could possibly specify
> the total number of arrays ahead of time.

OK, how about this. I just saved your example data in
a file named data.txt, and I wrote the following code to
read the data.

```
FUNCTION UnpackData, dataStruct

   struct = {int_1:0L, int_2:0L, float_1:0.0, str:""}
   d = Replicate(struct, N_Elements(*dataStruct.ptr))
   FOR j=0,N_Elements(d)-1 DO BEGIN
      parts = StrSplit((*dataStruct.ptr)[j], /Extract)
```

```
      d[j].int_1 = Long(parts[0])
      d[j].int_2 = Long(parts[1])
      d[j].float_1 = Float(parts[2])
      d[j].str = parts[3]
    ENDFOR

    RETURN, d

END ;-------------------------------------------------------

    lines = File_Lines(file)
    d = StrArr(lines)
    openr, 1, file
    readf, 1, d
    close, 1
    index = where(d EQ "", count)
    index = [index, N_Elements(d)]
    data = Replicate({name:"", ptr:Ptr_New()}, count+1)
    startIndex = 0
    FOR j=0, count DO BEGIN
        endIndex = index[j] - 1
        data[j].name = d[startIndex]
        data[j].ptr = Ptr_New(d[startIndex+1:endIndex])
        startIndex = index[j]+1
    ENDFOR
END
```

This consists of a main level program that reads the data file,
and a function UnpackData that unpacks the data that you have
read. I envision it working like this. Suppose you save this
to a file name readit.pro.

```
IDL> .compile readit
IDL> file = 'data.txt'
IDL> .go
IDL> Print, 'Number of data units read: ', N_Elements(data)
     Number of data units read:           3
IDL> a = UnpackData(data[0])
IDL> FOR j= 0,N_Elements(a)-1 DO Help, a[j], /Structure
   ** Structure <17ce540>, 4 tags, length=24, data length=24, refs=2:
   INT_1           LONG                 1
   INT_2           LONG                42
   FLOAT_1         FLOAT           3.14000
   STR             STRING    'Blah'
   ** Structure <17ce540>, 4 tags, length=24, data length=24, refs=2:
   INT_1           LONG                 2
   INT_2           LONG                77
```

```
   FLOAT_1         FLOAT          4.13000
   STR             STRING    'String'
IDL> b = UnpackData(data[1])
IDL> FOR j=0,N_Elements(b)-1 DO Help, b[j], /Structure
   ** Structure <17ce230>, 4 tags, length=24, data length=24, refs=2:
   INT_1           LONG              1
   INT_2           LONG             11
   FLOAT_1         FLOAT          1.34000
   STR             STRING    'String'
   ** Structure <17ce230>, 4 tags, length=24, data length=24, refs=2:
   INT_1           LONG              2
   INT_2           LONG             22
   FLOAT_1         FLOAT          1.43000
   STR             STRING    'Blah'
** Structure <17ce230>, 4 tags, length=24, data length=24, refs=2:
   INT_1           LONG              3
   INT_2           LONG             33
   FLOAT_1         FLOAT          3.41000
   STR             STRING    'String'
```

And so forth. Of course, you can give the structure more useful names, etc. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
