Subject: VMS input file line counting procedure
Posted by pendleton on Tue, 09 Nov 1993 22:36:52 GMT
View Forum Message <> Reply to Message

While we're still on the subject, and for the sake of completeness, I've
included at the end of this post a VMS command procedure which may be used to
count the lines in an input file prior to declaring IDL input array sizes.
I've tested it on my system and it seems to work fine. If you run into
any errors, please let me know about them.

Jim Pendleton, Programmer Analyst/Technical Services Specialist
GRO/OSSE, Dept. Physics & Astronomy
Northwestern University
j-pendleton@nwu.edu (708) 491-2748 (708) 491-3135 [FAX]
----------8<---------- Cut Here ---------->8----------

```
$! LINE_COUNT.COM
$!
$! Jim Pendleton, GRO/OSSE Northwestern University
$! j-pendleton@nwu.edu
$! 11/09/93
$!
$! Purpose:
$! Use the DCL APPEND command to count the lines in a file, writing the
$! number of lines to SYS$OUTPUT.  The anticipated use for this routine is
$! through the IDL SPAWN command, whereby a routine can first count the lines
$! of an input file of unknown size before declaring input arrays.  This saves
$! on paging I/O and memory fragmentation, and is an alternative to using array
$! append operations.  See the example below.
$!
$! Input Parameters:
$! P1 : The name of the file whose lines are to be counted.  Note that
$!      for indexed files, this method will not work in all probability.
$! Output:
$! The number of lines in the file is written to SYS$OUTPUT.  Unless an
$! abort condition arises (or the user has VERIFY turned on), this should
$! be the only output from this procedure. In IDL, the results written to
$! SYS$OUTPUT in a SPAWN-ed procedure can be captured by a string array
$! passed as the second parameter to IDL's SPAWN command.
$!
$! IDL Example:
$! Spawn, '@Line_Count ' + FileName, NLines
$! A = FltArr(Long(NLines(0)))
$! Tmp = 0.
$! I = 0L
$! OpenR, Unit, FileName, /Get_LUN
$! While (not EOF(Unit)) Do Begin
$!  ReadF, Unit, Tmp
```

```
$!  A(I) = Tmp
$!  I = Temporary(I) + 1L
$! EndWhile
$! Close, Unit
$!-
$ On Error Then Goto Abort
$ On Control_Y Then Goto Abort
$ CountTmp1 = P1
$ CountTmp2 = "Sys$Scratch:CountTmp2_ZZZ.TMP"
$!
$! We've got to turn off verify here, or we screw up SYS$OUTPUT.
$!
$ Assign/User 'CountTmp2 Sys$Output
$ Append/Log 'CountTmp1 NL:
$!
$! Read the file containing the APPEND statistics.
$!
$ Open/Read File 'CountTmp2
$ Read/End=Done_With_Append File Line
$Done_With_Append:
$ Close File
$ Delete/NoLog/NoConfirm 'CountTmp2;*
$!
$! Extract the "number of records" field from the APPEND/LOG status line.
$!
$ Paren = F$Locate("(", Line)
$ SubLine = F$Extract(Paren + 1, 999, Line)
$ RecPos = F$Locate(" rec", SubLine)
$ SubLine = F$Extract(0, RecPos, SubLine)
$ NRecords = F$Edit(SubLine, "Compress")
$ Write Sys$Output NRecords
$ Exit
$Abort:
$ On Error Then Continue
$ Close File
$ On Error Then Continue
$ Delete/NoLog/NoConfirm 'CountTmp2;*
$ Exit
```