

Hi,
I got a pro (not sure from where) that maybe does what you want:

```
;+++++  
+++++
```

FUNCTION DATA_INPUT, key, default, FILE=file

```
;;_Purpose:  ASCII data input from runtime file specification  
;  
;;_Description: This is an extension of the FITS format such that a  
single  
; keyword may imply input of up to 35 data items,  
provided the  
; data are non-character, and they are read from a  
single input  
; line, the length of which is < 81 characters.  
; Data delimiter is a comma. The number of data items  
identified  
; is equal to the number of commas found + 1.  
; Character input is constrained to a single item.  
; A '!' marks the start of a comment.  
;  
;;_Language:  IDL V3  
;  
;;_Input Par:  key      Keyword (FITS type) to extract  
; default     Default value assigned if key or file not  
found  
;  
;;_Keywords:  FILE      Input from specified disk file  
;              Default is 'USER.PAR'  
;  
;;_Result:    Value(s) of data following the keyword in the input  
file. The  
; data type is determined at run time.  
;  
;;_History:   TG        28-May-93  Initial programming  
;            TG        10-Feb-96  '!' recognition produces FLOAT  
;  
;;_Copyright: (c) FOCES project, University Observatory Munich  
;  
;-----
```

```

line = STRING(' ',FORMAT='(A80)')           ; Single
line format

ON_IOERROR, no_such_file
IF (NOT KEYWORD_SET(file)) THEN file = 'ORIENTATION.PAR' ;
Default input file
GET_LUN, lun
OPENR, lun, file
k = -1

WHILE (k EQ -1) AND NOT EOF(lun) DO BEGIN      ; Read
until EOF
  READF, lun, line
  keyword = STRMID(line, 0, 8)
  k = STRPOS(keyword, key)
  IF (k NE -1) THEN line = STRMID(line, 10, 70) ; Extract keyword
contents
ENDWHILE

FREE_LUN, lun
IF (k EQ -1) THEN BEGIN                       ; Keyword
not found
  value = default                             ;
Set default
  RETURN, value                               ; ... and return to
calling program
ENDIF

IF (STRPOS(line, '"') EQ 0) THEN BEGIN        ; String value
detected
  tmp = STRPOS(line, '"', 1)
  IF (tmp EQ -1) THEN BEGIN                   ; Second string
delimiter missing
  PRINT, '###ERROR: non-FITS format'
  PRINT, line
  STOP
ENDIF
  value = STRMID(line, 1, tmp-1)               ;
Extract string
ENDIF ELSE BEGIN                             ; Number(s)
detected
  i_elem = 0
  k = -1
  i = 0
  WHILE (i NE -1) DO BEGIN
    i = STRPOS(line, ",", k+1)               ; Detect
next comma

```

```

    j = STRPOS(line, "!", k+1)          ; Detect
exclamation mark
    ii = i
    IF (ii EQ -1) THEN BEGIN
        ii = STRLEN(line)                ; End of
input line
        IF (j NE -1) THEN ii = j-1        ; End of input
with comment
    ENDIF
    tmp = STRMID(line, k+1, ii-k-1)        ; Extract substring
between commas
    CASE (1) OF                          ; Determine
data type
        (STRPOS(tmp, 'D') NE -1): tmp = DOUBLE(tmp)
        (STRPOS(tmp, 'E') NE -1): tmp = FLOAT(tmp)
        (STRPOS(tmp, '.') NE -1): tmp = FLOAT(tmp)
    ELSE:                                tmp = LONG(tmp)
    ENDCASE
    i_elem = i_elem+1
    IF (i_elem EQ 1) THEN value = tmp ELSE value = [value,tmp]
    k = i
    ENDWHILE
ENDELSE
RETURN, value

no_such_file:                          ; File
not found
ON_IOERROR, NULL                        ; Reset
error status
value = default                          ;
Set default
RETURN, value

```

END

e.g. name.par

!Input file for the orientation simulation giving star and spectrograph parameters

```

!
Dtel  = 1.2
Fnum  = 13.0
Groove = 31.6e3

```

a short introduction to the parameters

#Dtel : Diameter of the telescope in m

#Fnum : Fnumber of the telescope

#groove : of the grating in m
