

Using an Excel Spreadsheet with PC SAS®, no Gymnastics required!

Diane Hall, Averion Inc., Framingham, Massachusetts

ABSTRACT

Do you regularly analyze data received in Excel spreadsheets using SAS? Do you wish for a single-step method for reading this data into a permanent dataset for analysis?

This paper will show you how to include your Excel spreadsheet in SAS using FILENAME and INFILE statements. This method does not require that you first save your file as TAB delimited. You are able to use the file as you received it. This saves processing time for the user, especially if your source for these Excel spreadsheets sends new updated files frequently.

THE SETUP

After setting up the program for a certain Excel file, you will only need to know 3 things each time a new spreadsheet comes in.

- (1) The ROW numbers containing the data.
- (2) The COLUMN numbers containing the data.
- (3) Record length.

THE DATA

VITALS.XLS

	A	B	C	D	E	F	G	H	I
1	Site Number	Patient	Visit Date	Age	Gender	Height (in)	Weight (lbs)	Temperature (f)	Comment
2	001	101	1/15/2004	18	M	75	252	98.4	
3	001	102	2/17/2004	47	M	69	207	97.9	
4	001	103	2/2/2004	28	F	68	151	97.4	
5	001	104	3/1/2004	52	M	67	186	98.1	This is a comment field (longer than 8 characters).
6	001	105	1/31/2004	39	F	62	133	98.3	
7	002	201	3/27/2004	26	F	70	172	97.8	
8	002	202	4/12/2004	31	F	65	155	98.2	
9	002	203	4/3/2004	60	M	73	194	97.6	
10	002	204	2/28/2004	42	M	68	200	98.9	
11	002	205	1/22/2004	55	F	64	137	98.6	
12	002	206	2/4/2004	43	M	66	168	98.2	

THE PIECES

NOXWAIT and NOXSYNC -- The NOXWAIT automatically closes the prompt window after the command has completed. The NOXSYNC instructs the system to process as soon as the command is issued.

SLEEP -- Sleep is used to allow the Excel window to initialize prior to attempting to open the file. The length of the wait period can be adjusted up or down as needed.

DDE -- Dynamic Data Exchange

R2C1:R12C9 -- Specifies the Row/Column range that contains the data. (Need to Know #1 and #2) The row input begins on 2 since there are field descriptions in row 1.

NOTAB -- SAS expects the delimiter to be a TAB when using DDE. This option allows for other delimiters.

DSD -- SAS will interpret 2 consecutive delimiters as a missing value for the variable. This also removes quotes from character variables, as necessary.

DLM='09'x -- Defines the delimiter as a TAB.

MISSOVER -- SAS will set the remaining variables to missing, if the data input line ends prior to filling all the variables being input, instead of continuing on to the next line to fill those missing variables.

PAD LRECL=300 -- Sets the width of the input buffer, and expands all records to that width using trailing blanks. (Need to Know #3)

INPUT varname :\$50 -- Sets the field width to 50 characters. The ":" must be present.

THE PROGRAM

```
options NOXWAIT NOXSYNC ;

*** Open Excel and VITALS data file ***;
x "start excel";

data _null_ ;
  x = SLEEP(6) ;
run ;

filename cmds DDE 'excel|system' ;
filename vitals DDE "Excel|H:\NESUG\VITALS.XLS!R2C1:R12C9" notab ;

data _null_ ;
  file cmds ;
  put '[FILE-OPEN("H:\NESUG\VITALS.XLS")]' ;
run ;

*** Create VITALS SAS dataset ***;
libname data 'H:\NESUG\' ;

data data.vitals ;
  infile vitals dsd dlm='09'x missover pad lrecl=300 ;

  input      SITE_NO $
            PAT_NO $
            VISIT_DT :mmddyy10.
            AGE $
            SEX $
            HGT_IN $
            WGT_LB $
            TEMP_F $
            COMMENT :$50.
            ;

  SITE_NUM = input(SITE_NO,3.) ;

  label SITE_NO = "Site Number" ;
  label SITE_NUM = "Site Number Numeric" ;
  label PAT_NO = "Patient Number" ;
  label VISIT_DT = "Visit Date" ;
  label AGE = "Age" ;
  label SEX = "Gender" ;
  label HGT_IN = "Height (Inches)" ;
  label WGT_LB = "Weight (Pounds)" ;
  label TEMP_F = "Temperature (Deg F)" ;
  label COMMENT = "Comment" ;

  format VISIT_DT mmddyy10. ;

run ;

(code continued on next page...)
```

```

*** Close Excel ***;
data _null_ ;
  file cmds ;
  put "[FILE-CLOSE()]" ;
  put "[QUIT()]" ;
run ;

proc print data = data.vitals width=min ;
run ;

```

THE RESULT

proc print of SAS dataset

SITE_NO	PAT_NO	VISIT_DT	AGE	SEX	HGT_IN	WGT_LB	TEMP_F	COMMENT	SITE_NUM
001	101	01/15/2004	18	M	75	252	98.4		1
001	102	02/17/2004	47	M	69	207	97.9		1
001	103	02/02/2004	28	F	68	151	97.4		1
001	104	03/01/2004	52	M	67	186	98.1	This is a comment field (longer than 8 characters)	1
001	105	01/31/2004	39	F	62	133	98.3		1
002	201	03/27/2004	26	F	70	172	97.8		2
002	202	04/12/2004	31	F	65	155	98.2		2
002	203	04/03/2004	60	M	73	194	97.6		2
002	204	02/28/2004	42	M	68	200	98.9		2
002	205	01/22/2004	55	F	64	137	98.6		2
002	206	02/04/2004	43	M	66	168	98.2		2

QUESTIONS

Is there any way to avoid having to change my program with respect to rows, columns, and lrecl once it's set up? I really want a run-and-done method.

When you think practically, you realize that given most datasets used in analysis, the number of variables (columns) is not likely to change. For instance, responses to questions on a questionnaire or data gathered on a CRF page. The change in the row/column combination is much more likely to be in the number of rows in the data sent. To simplify the upkeep where rows are concerned, you can set the ending value to something very large. In our example above, **R2C1:R12C9**, you would achieve the same resulting dataset using **R2C1:R100C9**.

Similarly, you can set the LRECL variable to be overly large to avoid frequent changes in value as well.

Unfortunately, this trick does not work with the column definitions. These must represent the columns actually appearing in the spreadsheet.

OK, this looks like a great method for getting an Excel spreadsheet into SAS, however my spreadsheet contains multiple worksheets. Will this work for me?

Yes. You can read in data from an Excel workbook with multiple worksheets using this method. Simply change the file reference from **VITALS.XLS!R2C1:R12C9** to **[VITALS.XLS]SHEETNAME!R2C1:R12C9** This specifies which sheet you wish to access in the workbook. The default sheet is the first sheet if one is not specified.

CONCLUSION

You can bring an Excel spreadsheet directly into SAS for use in analysis without the gymnastics of creating delimited flat files or using yet another outside program to accomplish the conversion. The benefit for me has been the ability to simply run an already created program and have the needed SAS dataset created for me, including any formatting and created variables, all in one step.

ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are registered trademarks of their specific companies.

CONTACT INFORMATION

Diane Hall
Averion, Inc.
4 California Avenue
Framingham, MA 01701
(508) 416-2659
DHall@averioninc.com