



Setting GuessingRows when importing Excel Files

**Dr. Arthur Tabachneck, Director
Data Management**

atabachneck@ibc.ca

**a program written by last year's SAS-L MVP (datanull)
who feels that most of the credit should go to Microsoft's Scripting Guys
(<http://www.microsoft.com/technet/scriptcenter/sgwho.msp>)**



Suppose you have data in an Excel workbook

```
data wont_work;  
  set sashelp.class;  
  output;  
  output;  
run;
```

```
proc export data= work.wont_work  
  outfile= "c:\wont_work.xls"  
  dbms=EXCEL2000 replace;  
  sheet="Sheet1";  
run;
```



TASS Meeting

Setting GuessingRows when Importing Excel Files

Open the Excel file we just created (i.e., c:\wont_work.xls)

and

at row 27
change the age from
12
to
unknown

then resave the file

Microsoft Excel - wont_work.xls

File Edit View Insert Format Tools SAS Data Window Help Adobe PDF Type a question for help

Active Data: Active Worksheet

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Sex	Age	Height	Weight						
2	Alfred	M	14	69	112.5						
3	Alfred	M	14	69	112.5						
4	Alice	F	13	56.5	84						
5	Alice	F	13	56.5	84						
6	Barbara	F	13	65.3	98						
7	Barbara	F	13	65.3	98						
8	Carol	F	14	62.8	102.5						
9	Carol	F	14	62.8	102.5						
10	Henry	M	14	63.5	102.5						
11	Henry	M	14	63.5	102.5						
12	James	M	12	57.3	83						
13	James	M	12	57.3	83						
14	Jane	F	12	59.8	84.5						
15	Jane	F	12	59.8	84.5						
16	Janet	F	15	62.5	112.5						
17	Janet	F	15	62.5	112.5						
18	Jeffrey	M	13	62.5	84						
19	Jeffrey	M	13	62.5	84						
20	John	M	12	59	99.5						
21	John	M	12	59	99.5						
22	Joyce	F	11	51.3	50.5						
23	Joyce	F	11	51.3	50.5						
24	Judy	F	14	64.3	90						
25	Judy	F	14	64.3	90						
26	Louise	F	12	56.3	77						
27	Louise	F	Unknown	56.3	77						
28	Mary	F	15	66.5	112						
29	Mary	F	15	66.5	112						
30	Philip	M	16	72	150						
31	Philip	M	16	72	150						

Ready NUM



Now import the spreadsheet:

```
proc import data= WORK.TEST1  
    datafile= "s:\wont_work.xls"  
    dbms=EXCEL2000 replace;  
    sheet="Sheet1";  
    mixed=yes;  
    getnames=yes;  
run;
```



TASS Meeting

Setting GuessingRows when Importing Excel Files

Unless you or someone
at your office has already
changed the Guessingrows
setting in your system's
Window Registry

the imported file will
look like the one shown
to the right

where the variable age
is numeric
and has a missing value
for the 26th record

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alfred	M	14	69	112.5
3	Alice	F	13	56.5	84
4	Alice	F	13	56.5	84
5	Barbara	F	13	65.3	98
6	Barbara	F	13	65.3	98
7	Carol	F	14	62.8	102.5
8	Carol	F	14	62.8	102.5
9	Henry	M	14	63.5	102.5
10	Henry	M	14	63.5	102.5
11	James	M	12	57.3	83
12	James	M	12	57.3	83
13	Jane	F	12	59.8	84.5
14	Jane	F	12	59.8	84.5
15	Janet	F	15	62.5	112.5
16	Janet	F	15	62.5	112.5
17	Jeffrey	M	13	62.5	84
18	Jeffrey	M	13	62.5	84
19	John	M	12	59	99.5
20	John	M	12	59	99.5
21	Joyce	F	11	51.3	50.5
22	Joyce	F	11	51.3	50.5
23	Judy	F	14	64.3	90
24	Judy	F	14	64.3	90
25	Louise	F	12	56.3	77
26	Louise	F	.	56.3	77
27	Mary	F	15	66.5	112
28	Mary	F	15	66.5	112



and suppose that

- **you don't know enough about changing registry settings**

and/or

- **you don't like to work with the Window's registry**



a solution

**a relatively small SAS program that can both change
and restore your Window's Registry Settings**



```
guessingrows.sas *
libname STORPGM (work); *Where you actually store your stored data step programs;

data _null_ / pgm=STORPGM.SetGuessingRows;
  attrib set length=$32;
  attrib GuessRows length=8;
  set = upcase(symget('SET'));
  if set eq 'RESET' then do;
    if not symexist('CurrentGuessRows') then do;
      putlog 'ERR' 'OR: Global macro variable "CurrentGuessRows" not defined cannot reset Guessing Rows';
      stop;
    end;
    GuessRows = input(symget('currentGuessRows'),f16.);
  end;
  else GuessRows = input(resolve('%sysevalF(&set,Integer)'),f16.);
  if missing(guessRows) or sign(guessRows) lt 0 then do;
    putlog 'WARNING: Bad ' GuessRows=;
    stop;
  end;
  putlog 'NOTE: Setting MicroSoft Jet Engine Guessing Rows=' GuessRows;
  length script command filevar $256;
  script = catx('\',pathname('WORK'),'GuessRows.vbs');
  filevar = script;
  file dummy1 filevar=filevar recfm=v lrecl=512;
  put 'Set objShell = WScript.CreateObject("WScript.Shell")';
  put 'const target = "HKLM\SOFTWARE\Microsoft\Jet\4.0\Engines\Excel\TypeGuessRows"';
  put 'With objShell';
  put '  GuessRows = .RegRead (target)';
  put '  WScript.Echo "Current Value of GuessRows=" & GuessRows';
  put '  .RegWrite target,' guessRows ', "REG_DWORD"';
  put '  GuessRows = .RegRead (target)';
```



TASS Meeting

Setting GuessingRows when Importing Excel Files

```
guessingrows.sas *
put ' WScript.Echo "GuessRows Set to GuessRows=" & GuessRows';
put ' End with';
filevar = catx('\',pathname('WORK'),'dummy.vbs');
file dummy1 filevar=filevar;
filevar = script;
infile dummy2 filevar=filevar end=eof;
putlog 'NOTE: The SCRIPT';
do _n_ = 1 by 1 until(eof);
  input;
  putlog 'NOTE- ' _n_ z3. +2 _infile_;
end;
command = catx(' ', 'cscript', quote(trim(script)));
infile dummy pipe filevar=command lrecl=1024 end=eof;
putlog 'NOTE: The SCRIPT output';
do _n_ = 1 by 1 until(eof);
  input @;
  if index(_infile_, 'Current Value of GuessRows')
    then input @'GuessRows=' CurrentGuessRows;
  else input;
  putlog 'NOTE- ' _n_ z3. +2 _infile_;
end;
if not symexist('CurrentGuessRows')
  then call symputX('CurrentGuessRows', CurrentGuessRows, 'G');
else call symdel('CurrentGuessRows');
stop;
run;
%macro GuessRows(set);
  data pgm=STORPGM.SetGuessingRows;
  run;
  %put NOTE: Macro &systemacroname ending execution;
%mend GuessRows;
```



after running that code,

calling the macro, as follows, will capture your Registry's current setting

```
%GuessRows(reset);
```

then, calling the macro, as follows, will set your Registry to the maximum setting

```
%GuessRows(set=ffffx);
```

then, if you rerun the proc import, the age field will import correctly

finally, calling the macro w/reset will reset your Registry to its previous setting

```
%GuessRows(reset);
```



if you want to understand the program:

the program uses a number of calls and functions which I wasn't familiar with

- `data _null_ / pgm=` - creates a stored compiled DATA Step Program
- `libname somename (work)` - will assign a libname to your work directory
- `stop` - stops the execution of the current section
- `symexist` - returns an indication of the existence of a macro variable
- `symget` - returns the value of a macro variable during DATA step execution
- `putlog` - creates a logical-name in your process-level logical name table
- `sign` - returns the sign of a value
- `catx` - concatenates char strings, removes leading and trailing blanks + inserts separators
- `filevar` - specifies the current output file for PUT statements
- `call symputx` - assigns a value to a macro variable and removes leading and trailing blanks



made possible because of SAS-L

(<http://www.listserv.uga.edu/cgi-bin/wa?A0=sas-l&D=1&H=0&O=D&T=1>)



TASS Meeting

Setting GuessingRows when Importing Excel Files

Questions?