# Paper 251-28

## SHOW-AND-TELL:
## How to Use MS Office Products to Help With SAS® Applications

Myra A. Oltsik

## ABSTRACT

Do you hate to type? I do. Over the years I've found ways to use other programs to help me develop SAS® applications. This presentation will show you ways to use MS Word, MS Excel, and other products to help "write" SAS® code, no matter what your level of SAS® experience.

## EXCEL INSTRUCTIONS

Even those who use MS Excel a lot may not be familiar with Excel functions that allow one to parse or concatenate fields. Some of these and other functions are listed in Appendix A.

## WORD INSTRUCTIONS

MS Word special characters or marks can help when making a document into a text file. Some of these are listed in Appendix B.

## A SIMPLE EXCEL EXAMPLE

An auto insurance company may have many drivers on a single policy. For each driver you have that driver's name, date of birth, license number, license state, license date, etc. The file you have groups all of these together: first the names, then the birth dates, etc. You want to see all the information for one driver together, and then the information for the next driver. This will mean either typing the field names in order, or cutting and pasting each field for each driver. Instead, let MS Excel rearrange the fields for you.

The following shows the formulas you would use to parse a field name such as DRIVER1:

| | | |
|---|---|---|
| driver1 | =RIGHT(A8,1) | =LEFT(A8,LEN(A8)-1) |
| driver2 | =RIGHT(A9,1) | =LEFT(A9,LEN(A9)-1) |
| driver3 | =RIGHT(A10,1) | =LEFT(A10,LEN(A10)-1) |
| driver4 | =RIGHT(A11,1) | =LEFT(A11,LEN(A11)-1) |

The LEN() function in Excel is similar to the LENGTH() function in SAS®. The use of these functions produces the following for the field names:

| | | |
|---|---|---|
| driver1 | 1 | driver |
| driver2 | 2 | driver |
| driver3 | 3 | driver |
| driver4 | 4 | driver |
| dob1 | 1 | dob |
| dob2 | 2 | dob |
| dob3 | 3 | dob |
| dob4 | 4 | dob |
| license_no1 | 1 | license_no |
| license_no2 | 2 | license_no |
| license_no3 | 3 | license_no |
| license_no4 | 4 | license_no |

With this, you can now sort the rows by the second column. Next, concatenate the field name with a comma, using the ampersand (&) and double quotes[1] around the comma [shown in the middle column below], to produce a list for a PROC SQL statement[2]:

| | | |
|---|---|---|
| driver1 | =A8&"," | driver1, |
| dob1 | =A9&"," | dob1, |
| license_no1 | =A10&"," | license_no1, |
| driver2 | =A11&"," | driver2, |
| dob2 | =A12&"," | dob2, |
| license_no2 | =A13&"," | license_no2, |
| driver3 | =A14&"," | driver3, |
| dob3 | =A15&"," | dob3, |
| license_no3 | =A16&"," | license_no3, |
| driver4 | =A17&"," | driver4, |
| dob4 | =A18&"," | dob4, |
| license_no4 | =A19 | license_no4 |

The following is a PROC SQL statement that reorders the list of field names and sorts the file at the same time:

[1] Excel does not allow the single quote (') to be used instead of the double quote (") as SAS® does.

[2] There are several ways to rearrange the variables in a dataset in SAS®. This example shows how to use Excel concatenation.

```
proc sql;
create table new as
   select
      policy_no,
      lname,
      addr1,
      addr2,
      city,
      st,
      zip,
      phone,
      driver1,
      dob1,
      license_no1,
      driver2,
      dob2,
      license_no2,
      driver3,
      dob3,
      license_no3,
      driver4,
      dob4,
      license_no4
   from old
   order by
      policy_no
   ;
quit;
```

## A SIMPLE WORD EXAMPLE

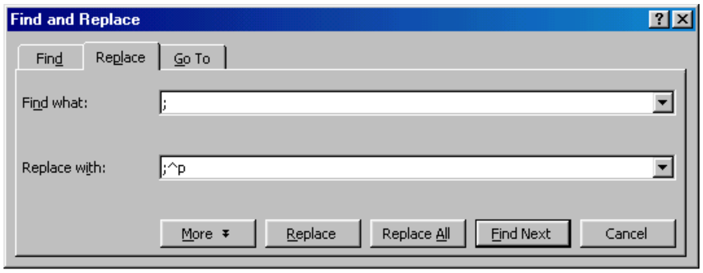
I've often inherited someone else's code that I find difficult to read; each coder has his or her own style. As the previous example shows, I'm a "vertical" coder. When I get code written by a "horizontal" coder, I find it hard to read or modify without first making it vertical. For example, imagine the following on one line:

```
proc sort data=old(keep=date time
name city zip) out=new; by zip;
run;
```

Instead, I want it to look like this:

```
proc sort data=old(keep=
      date
      time
      name
      city
      zip
   ) out=new;
   by zip;
run;
```

MS Word (and Ultra-Edit) can handle tabs and carriage returns in edit mode. In this case, one can globally change all semi-colons to a semi-colon with a carriage return:



Find what: ;

Replace with: ;^p

The ^p adds the carriage return. Similarly, a ^t would add a tab. You could replace two carriage returns with one by changing ^p^p to ^p, etc.

## USING EXCEL TO CREATE INPUT LIST

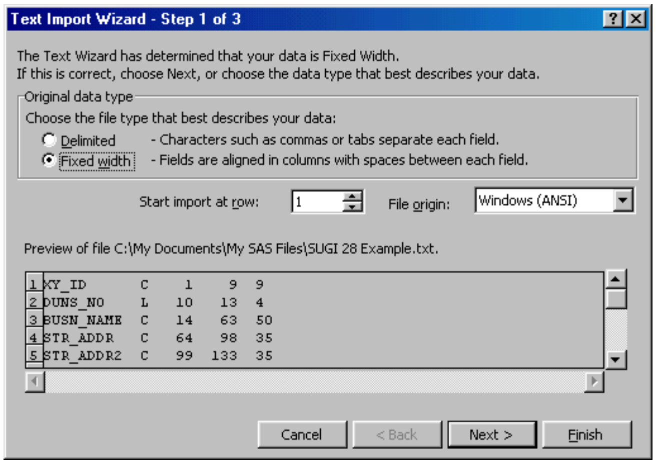
You've received an MVS file in EBCDIC, and you have to read it into ASCII on a PC. Here's the layout you get.

```
XY_ID        C    1    9   9
DUNS_NO      L   10   13   4
BUSN_NAME    C   14   63   50
STR_ADDR     C   64   98   35
STR_ADDR2    C   99  133   35
CITY         C  134  167   34
ST_PROV      C  168  169   2
CNTRY        C  170  171   2
ZIP          C  172  176   5
POSTAL_CD    C  177  186   10
COUNTRY      C  187  200   14
CNTRY_CD     C  201  202   2
XY_IT_CLNT L  203  206   4
CLNT         L  207  210   4
CLNT_CD      C  211  211   1
IT_CLNT      L  212  215   4
IT_CLNT_CD C  216  216   1
HQ_CLNT      L  217  220   4
SALES_AMT    P  221  225   5   0
NAICS_CD     C  226  231   6
CNTRY_QTY    S  232  233   2
ME_OUT_AMT P  234  238   5   2
ME_IN_AMT    P  239  243   5   2
ME_INT_AMT P  244  248   5   2
```

How do you make it look like you need it to look to read in each variable correctly?

```
@   1 xy_id        $ebcdic009.
@  10 duns_no      s370fib004.
@  14 busn_name    $ebcdic050.
@  64 str_addr     $ebcdic035.
@  99 str_addr2    $ebcdic035.
@ 134 city         $ebcdic034.
@ 168 st_prov      $ebcdic002.
@ 170 cntry        $ebcdic002.
@ 172 zip          $ebcdic005.
@ 177 postal_cd    $ebcdic010.
@ 187 country      $ebcdic014.
@ 201 cntry_cd     $ebcdic002.
@ 203 xy_it_clnt   s370fib004.
@ 207 clnt         s370fib004.
@ 211 clnt_cd      $ebcdic001.
@ 212 it_clnt      s370fib004.
@ 216 it_clnt_cd   $ebcdic001.
@ 217 hq_clnt      s370fib004.
@ 221 sales_amt    s370fpd005.
@ 226 naics_cd     $ebcdic006.
@ 232 cntry_qty    s370fib002.
@ 234 me_out_amt   s370fpd005.2
@ 239 me_in_amt    s370fpd005.2
@ 244 me_int_amt   s370fpd005.2
```

Start by reading in the text file containing the layout into Excel. With a .txt extension, Excel will check for a delimiter. In this case there is none. As shown below, make sure “fixed width” is chosen, then follow the wizard to read in the list:



The Text Wizard has determined that your data is Fixed Width.
If this is correct, choose Next, or choose the data type that best describes your data.

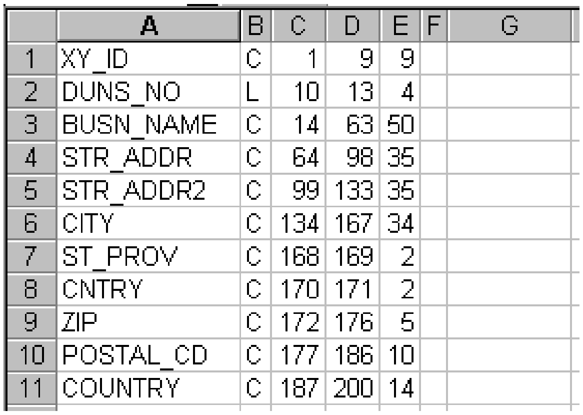
Original data type
Choose the file type that best describes your data:
○ Delimited - Characters such as commas or tabs separate each field.
● Fixed width - Fields are aligned in columns with spaces between each field.

Start import at row: 1 File origin: Windows (ANSI)

Preview of file C:\My Documents\My SAS Files\SUGI 28 Example.txt.

```
1 XY_ID      C    1    9   9
2 DUNS_NO    L   10   13   4
3 BUSN_NAME  C   14   63   50
4 STR_ADDR   C   64   98   35
5 STR_ADDR2  C   99  133   35
```

Your spreadsheet will look like the following:



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | XY_ID | C | 1 | 9 | 9 | | |
| 2 | DUNS_NO | L | 10 | 13 | 4 | | |
| 3 | BUSN_NAME | C | 14 | 63 | 50 | | |
| 4 | STR_ADDR | C | 64 | 98 | 35 | | |
| 5 | STR_ADDR2 | C | 99 | 133 | 35 | | |
| 6 | CITY | C | 134 | 167 | 34 | | |
| 7 | ST_PROV | C | 168 | 169 | 2 | | |
| 8 | CNTRY | C | 170 | 171 | 2 | | |
| 9 | ZIP | C | 172 | 176 | 5 | | |
| 10 | POSTAL_CD | C | 177 | 186 | 10 | | |
| 11 | COUNTRY | C | 187 | 200 | 14 | | |

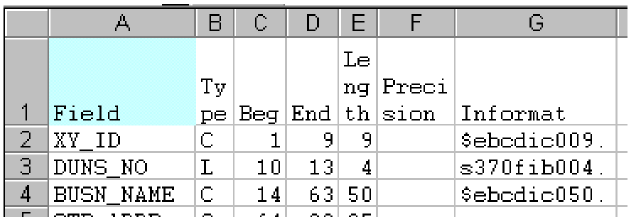
With these columns – field, type, beginning column, ending column, and field length – you have all the information you need to create the position informat code.

The field type and length tell you what type of informat you need.

| | Field Type | Informat Needed |
|---|---|---|
| C | Character | $EBCDICw. |
| L | [Long] Integer | S370FIBw. |
| P | Packed Decimal | S370FPDw. |
| S | Short Integer | S370FIBw. |

By using the Excel “IF” formula combined with the “TEXT” function, which converts numerics into text, you can easily build the informats you need for each variable. The following is typed into each cell in column G below for each field. It may be hard to visualize in this paper, but it will look clearer in your spreadsheet:

```
=IF(B2="C","$ebcdic"&TEXT(E2,"000")
&".",IF(B2="L","s370fib"&TEXT(E2,"0
00")&".",IF(B2="P","s370fpd"&TEXT(E
2,"000")&".",IF(B2="S","s370fib"&TE
XT(E2,"000")&"."))))
```



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Field | Type | Beg | End | Length | Precision | Informat |
| 2 | XY_ID | C | 1 | 9 | 9 | | $ebcdic009. |
| 3 | DUNS_NO | L | 10 | 13 | 4 | | s370fib004. |
| 4 | BUSN_NAME | C | 14 | 63 | 50 | | $ebcdic050. |

Unfortunately, numeric fields with decimal places – precision (column F) greater than zero – require a slight modification. After entering the “IF” formula for each variable, sort the file on the precision field. (You’ll always be able to sort the fields back to their original order by sorting on the Beg column.) For those fields with decimals, substitute the following code:

```
="s370fpd"&TEXT(E2,"000")&"."&F2
```

**OTHER EXAMPLES**

Since it will be easier to see this during the presentation, please refer to the presentation for more information. Other tricks will also be shown there. For a copy of the presentation, please contact me at my0402@mail.com.

Myra A. Oltsik
White Plains, NY

## APPENDIX A: MS Excel Functions[3]

**CONCATENATE (text1,text2,...)** -- text1, text2, ... are 1 to 30 text items to be joined into a single text item. The text items can be text strings, numbers, or single-cell references.
**&** -- can be used instead of CONCATENATE to join text items
**IF(logical_test,value_if_true,value_if_false)** -- logical_test is any value or expression that can be evaluated to TRUE or FALSE
**LEFT(text,num_chars)** [default for num_cars is 1] -- returns the first character or characters in a text string, based on the number of characters you specify
**LEN(text)** -- returns the number of characters in a text string
**REPT(text,number_times)** -- text is the text you want to repeat
**RIGHT(text,num_chars)** [default for num_cars is 1] -- returns the last character or characters in a text string, based on the number of characters you specify
**TEXT(value,format_text)** -- value is a numeric value, a formula that evaluates to a numeric value, or a reference to a cell containing a numeric value. format_text is a number format in text form from in the Category box on the Number tab in the Format Cells dialog box. Format_text cannot contain an asterisk (*) and cannot be the General number format.

## APPENDIX B: MS Word Special Characters[4] [5]

| | |
|---|---|
| ^? | Any Character |
| ^# | Any Digit |
| ^$ | Any Letter |
| ^^ | Caret Character |
| ^n | Column Break |
| ^a | Comment Mark |
| ^l | Manual Line Break |
| ^m | Manual Page Break |
| ^p | Paragraph Mark |
| ^b | Section Break |
| ^t | Tab Character |
| ^w | White Space |

[3] Definitions come from the MS Excel Help Menu
[4] From MS Word Repeat Window
[5] The "p", "t", etc. in these instances *must* be in lower case.