

The Simple Application of SAS DDE in Excel

Danting Guo, Fountain Medical Development, Inc., Nanjing, China

ABSTRACT

Dynamic Data Exchange (DDE) is a method of dynamically exchanging information between Windows applications. The DDE mechanism enable SAS to control Excel or Word. Basically, what you can do with Excel/Word using keyboard and mouse can be replicated programmatically in SAS. Although the process of exporting the data from a SAS data set to an Excel file is relatively simple, but SAS output need to be edited to meet the statistical requirements, someone has to manually create and format an Excel graph, which can be a time consuming process and the editing process will be also very easy to make mistakes. This paper mainly introduces how to use DDE within SAS to control excel and provides some examples, such as how to send data or results generated in SAS to specific cell locations in an Excel worksheet and so on, which may greatly reduce the amount of work for the following edit process.

INTRODUCTION

Many methods can be available to export data, why use DDE? Actually, these alternatives are useful under the right conditions. Such as, PROC EXPORT is simple to use, but are limited in the way you can define your data use. The most things that can be done manually in Excel that can be automated from within a base SAS program by using DDE. DDE and SAS join forces to make Excel spreadsheet more useful. Because using DDE link makes SAS power to populate a user-friendly spreadsheet, and as needed format data and send data or results generated in SAS to specific cell locations in an Excel worksheet. To create complex, highly formatted Excel spreadsheets from a SAS program, DDE is more flexible and useful.

THE BASICS

We can use the x command to invoke Excel.

```
option noxwait noxsync;
```

```
x "C:\Program Files\Microsoft Office\Office\excel.exe" ; * opens a new workbook;
```

The NOXWAIT allows you to use the X command, which opens outside programs, without typing "exit" before returning to SAS. The NOXSYNC turns off the normal buffering initiated with the X Window System.

Running the above two lines will open a blank Excel spreadsheet. Then, we can indicate with a filename the sheet and cells of the open sheet that we will write to from SAS.

```
data _null_;
```

```
    rc=sleep(5);* wait for 5 seconds;
```

```
run;
```

The sleep command lets SAS wait for a few seconds until Excel is fully running.

One way, we can use DDE commands to open existing workbook, change sheet name, save finished workbook and exit excel.

```
filename cmd dde 'Excel|System';
```

```
data _null_;
```

```
    file cmd;
```

```
    put '[Open("c:\temp\example.xlsx")]';
```

```

x=sleep(3);
put '[workbook.name("sheet1","example data")]';
put '[save()]';
x=sleep(1);
put '[error(false)]';
put '[quit()]';
run;

```

The error(false) command tells Excel that you don't want a prompt to ask for confirmation of intention. This example is the most basic application, it also can do some things such as select worksheet, cells, rows, columns, formatted cells, control/change levels for Excel subtotals and so on.

Other way, we know that the core of DDE output is in a filename statement and a file, put within a data step. The filename defines the output location and put selects data fields to be output.

```
filename sample1 dde 'excel|T1!r1c1:r1c2';
```

Above, we are creating a filename sample1 that will write to the sheet and cells indicated T1 (the default name of a new Excel file), from the first cell in the first row ("r1c1") to the second cell in the first row ("r1c2").

Next, we can run the data step below. It includes a file statement referring to the specified location in the open Excel file. We create two variables, x, y, with one value each. Then we put the two variables into Excel.

```

data _null_;
  file sample1;
  x = "Bace";
  y = "Change";
  put x y;
run;

```

We can see that these character values have been put into the first two cells of the first row in Excel.

A	B
Bace	Change

Table 1. Example for Putting into Variables to Excel

Next, we can indicate the next block of cells that we wish to write to.

```

filename sample2 dde 'excel|T1!r2c1:r11c2';
data _null_;
  file sample2;
  do i=1 to 10;
    x=i+3;
    y=(x+10)/2;
    put x y;
  end;
run;

```

We can see that the Excel sheet now contains these values.

A	B
Base	Change
4	7
5	7.5
6	8
7	8.5
8	9

Table 2. Example for Putting into Values to Excel

In Excel, Table 2 shows where the filename statement will put raw data. Row and column locations can be larger than needed, in case you're not sure how many rows will be exported. If fewer rows and columns than needed are defined in the filename statement, the extra data will not be output. For example, if you have eight columns of data and the output is set up as r1c1:r6c7, the eighth field will not be output to Excel.

THE NOTICE ABOUT NOTAB OPTION

It works well with numeric data and text without imbedded blanks. However, the purpose of DDE is to exchange data from a dataset into precise columns in Excel. If Excel meets a space, it assumes "new column". When the NOTAB option is placed in a filename statement that uses the DDE device-type keyword, the SAS System accepts character delimiters other than tabs between variables. '09'x is the ASCII hexadecimal code for the invisible TAB character, TABs here are being explicitly re-inserted into the source code so that variable values will occupy adjacent cells in the spreadsheet.

The follow codes can show a simple comparison:

```
filename outxls1 dde 'excel|sheet1!r1c1:r1c10' notab;
filename outxls2 dde 'excel|sheet1!r2c1:r2c10' notab;
filename outxls3 dde 'excel|sheet1!r3c1:r3c10';
data _null_;
  x="SAS Data"; y="Example for DDE";
  file outxls1;
  put x y;
  file outxls2;
  put x '09'x y;
  file outxls3;
  put x y;
run;
```

A	B	C	D	E
SAS Data Example for DDE				
SAS Data	Example for DDE			
SAS	Data	Example	for	DDE

Table 3. Example for NOTAB Option

THE SIMPLE APPLICATION

This use of DDE can be very useful if you have multiple data sets for which you want to do the same analysis and present results in a consistent template in Excel. We created a results template in Excel for presenting the frequency of scores and chi-square test by two groups in a dataset. The values have been blank and we saved the spreadsheet as example.xlsx and named the template sheet T2.

A	B	C	D	E	F	G
	Group A		Group B			
Score Group	Frequency	Percent (%)	Frequency	Percent (%)	Chi-square	P-value

Table 4. Template for the Frequency of Scores and Chi-square Test by Two Groups

Next, we can look at an example dataset with two groups and calculate the frequency of scores and chi-square test;

```
proc sort data=data.score out=score;
  by group;
run;
proc freq data=score noprint;
  table score_grp*sgroup/nocum out=out1;
  by group;
run;
proc freq data=score noprint;
  table group*score_grp/chisq;
  output out=out2 chisq;
run;
data out;
  merge out1(where=(group="A"))
        out1 (where=(group="B") rename=(count=count1 percent=percent1))
        out2 (keep=_pchi_ p_pchi);
run;
```

open our template spreadsheet and use a filename statement to indicate the three empty cells in our template.

```
filename sample2 dde 'excel|[example.xlsx]T2!r3c1:r5c7' notab;
data _null_;
  set out;
  file sample2;
  format percent percent1 8.1 _pchi_ p_pchi 8.3;
  put sgroup '09'x count '09'x percent '09'x count1 '09'x percent1 '09'x _pchi_
'09'x p_pchi '09'x;
  file sample3;
run;
```

A	B	C	D	E	F	G
	Group A		Group B			
Score Group	Frequency	Percent (%)	Frequency	Percent (%)	Chi-square	P-value
0 - 13	8	30.8	22	61.1	8.627	0.013
14 - 20	12	46.2	5	13.9	.	.
> 20	6	23.1	9	25	.	.

Table 5. The Frequency of Scores and Chi-square Test by Two Group Subjects

After running the SAS code, we can now look at our filled-in template. Our results have been formatted according to the formats for the cells to which they are exported.

We can also select the cells to be formatted by DDE commands and use Excel workbook which is from

above example.

```
filename example dde 'Excel|System';
data _null_;
  file example;
  put '[workbook.activate("T2")]';
  put '[select("r3c1:r5c1")]';
  put '[format.font("Calibri",11,true,false,false,false,5)]';
  put '[alignment(7)]';
  put '[select("r1c2:r1c5")]';
  put '[patterns(1,,3)]';
  put '[select("r2c1:r2c7")]';
  put '[format.font("Calibri",11,true)]';
  put '[save.as()]';
  put '[error(false)]';
run;
```

A	B	C	D	E	F	G
	Group A		Group B			
Score Group	Frequency	Percent(%)	Frequency	Percent(%)	Chi-square	P-value
0 - 13	8	30.8	22	61.1	8.627	0.013
14 - 20	12	46.2	5	13.9	.	.
> 20	6	23.1	9	25	.	.

Table 6. Example for Using DDE Commands to Format Cells

According to the Table 6, it selects specific cells and change font, style, size, color and background, which can also use a mouse. For this example, I just wanted to illustrate what you can do with Excel/Word using keyboard and mouse can be replicated programmatically in SAS.

CONCLUSION

In the statistical process, we often use Excel spreadsheet, it is easy to maintain and share, we hope to use the simplest tools to do the most effective works. SAS is a superior tool to process data and summarize results, most things that can be done manually in Excel that can be automated from within a base SAS program by using DDE. Using DDE link makes SAS power to populate a user-friendly spreadsheet, and as needed update the results.

DDE and SAS join forces to make Excel spreadsheet more useful. It is undeniable that DDE has some advantages, however, it also has some disadvantages. One of the biggest advantages to Excel is that it will perform the same calculations or graphics analysis over and over on new data supplied to a standard area. One of the biggest disadvantages of Excel is that it expects the data to be in exactly the same place every time. So we should pay attention to its characters, and then have a choice of reasonable application.

REFERENCES

- SAS Help and Documentation about Dynamic Data Exchange (DDE).
- SAS Institute Inc. The complete Guide to the SAS Output Delivery System. Version 8. Cary, NC, USA: SAS Institute Inc.2000.
- Xiaoyan Tian. 2000.3. SAS Application of DDE Technology in Statistical Reporting Automation.
- Rebecca Symes. Using DDE to connect SAS® results with MS-Excel®.

ACKNOWLEDGMENTS

Special thanks to Xuhua Deng for taking his precious time to review my paper.

CONTACT INFORMATION

Name: Danting Guo

Enterprise: Fountain Medical Development Co., Ltd

Address: Room 403, Building 43, No.70, Headquarter Base, Phoenix Road, Jiangning District

City, State ZIP: 211100

Work Phone: 025-86155129

Fax: 95040100263-552194

E-mail: danting.guo@fountain-med.com