“Using DDE to Communicate between SAS and Excel”

James Hoffman, DBSI Corp., Austin, TX
SCSUG Conference October 26-28, 2003

Abstract:

This presentation will show the audience how to interact with Microsoft Excel via a DDE link from SAS. We go past the macro driven routines that have been presented previously into the specific code required to do various tasks. We will start with how the connection to Excel is made. Then jump into Navigating the spreadsheet and entering variable data and formulas. Following that we look at a full range of formatting options. Finishing up with saving your work and exiting the system.

Introduction:

More and more SAS users are wanting to output directly into Excel spreadsheets. I have put together several small programs that demonstrate various capabilities of using DDE to communicate between SAS and Excel. When ever I am trying to work out a problem, I use one of my sample programs to assist me.

Overview:

We will go through several scenarios to see the capabilities that are available. There are several methods that can be used in starting Excel. We will look at the one I use most often. It really depends on whether or not you expect Excel to already be running or not as to how much checking you do to determine if you need to start it or not.

Starting Excel:

```
%macro start_excel;
filename testin dde 'EXCEL|SYSTEM' notab;

data _null_;
file testin;
run;
%if &syserr>0 %then %do;
      x 'C:\WINNT\Profiles\Administrator\Desktop\Excel.lnk';
      data _null_;
      x=sleep(5);
      run;
%end;
%mend start_excel;
%start_excel;
filename testin clear;
```

Making The Connection:

This is where you identify the scope of your intended operation. The first filename statement is the one I use most of the time. It allows complete freedom to move around the spreadsheet, opening and closing the Excel file. In addition you can select individual worksheet tabs. The second filename statement is very specific. You are limited to specific rows and columns in spreadsheet “Schema_Alpha.xls” and tab “desc”.

```
filename excel dde 'EXCEL|SYSTEM';

options firstobs=2;
filename in_out dde 'Excel|C:\[Schema_Alpha.xls]desc!R1C1:R1400C7';
```

SAS Opens the spreadsheet:

```
data _null_;

file excel lrecl=500;

put '[error(false)]';
put '[Close.All()]';
put '[App.Restore()]';
put '[App.Maximize()]';
put '[open("C:\My Documents\Jims Test Book.xls")]';
put '[New()]'; /*Open a new sheet*/
put '[Workbook.activate("Customer LTV")]';
```

Navigating the spreadsheet:

```
/* Select one cell */
put '[Select("R1C1")]';

/* Select four cells in row 1 */
put '[Select("R1C1:R1C4")]';

/* Select forty cells columns 1 through 4 in rows 1 through 10 */
put '[Select("R1C1:R10C4")]';

/* Select six cells in row 1, columns 1 through 4 and 6 through 8 */
put '[Select("R1C1:R1C4,R1C6:R1C8")]';

/* Select one row down same column */
put '[Select("R[1]C")]';
```

```
/* Select in the same row, one column to the right */
put '[Select("RC[1]")]';

/* Select one row up and one column to the left */
put '[Select("R[-1]C[-1]")]';

/* Select Ten rows down and four columns to the right */
put '[Select("R[10]C[4]")]';
```

Entering values and formulas:

Tabs and Line Feeds

This is a tidbit I found in one of the many articles that have been written about sending output to Excel via DDE. If tab was defined as a variable in the program like “tab = ‘09’x;” that it could cause certain problems, like extra spaces. The fail safe method is to set it as a macro variable or use it as a literal in the put statements. I have included a line feed character because I have used it so solve a particular problem. I have a variable that ended in the percent sign (“%”) and Excel would hang at that point even with a tab following. I found that using the line feed character solved the problem.

```
%let tab='09'x;
%let lf='0a'x;
```

Regular values

This is pretty straight forward. Output values as literals or variables.

```
put '.23' &tab ‘.45’ &tab ‘.97’ &tab;

put value1 &tab value2 &tab value3 &tab;

put '1' '09'x '2' '09'x '3'/
      '4' '09'x '5' '09'x '6'/
      '7' '09'x '8' '09'x '9'/
      '10';
```

Formulas

```
put '[Formula("=IF(ISERR(R2C3/R2C2),""N/A"",(R2C3/R2C2))")]'
&tab;

put '[Formula("=IF(ISERR(R2C4/R2C3),""N/A"",(R2C4/R2C3))")]' &tab;
```

Basic Formatting:

You can either code these statements with each put statement or create variables and use the variable name. I prefer this method. This way I don't have to remember the exact syntax, I just think of what I want done. The values for formats are pretty easy to determine. In Excel select FORMAT|CELLS then highlight the category and type. Then click on custom in the category column, the value for you format will be highlighted in the sample window.

Data type

```
Retain _dollar2 _dollar0 _percent2 _percent0 _number $40 (
      '[Format.Number("$#,##0.00")]'
      '[Format.Number("$#,##0")]'
      '[Format.Number("0.00%")]'
      '[Format.Number("0%")]'
      '[Format.Number("#,##0")]'  );
```

Alignment

```
Retain _Right_Just _Left_Just _Center $40 (
      '[Alignment(4)]'
      '[Alignment(2)]'
      '[Alignment(3)]'  );
```

Borders

There are fourteen different types including no border. The positional values for this statement is defined in the following comment. When you are dealing with a single cell you can simply use a value for "outline" or "outline color" or you can specify each segment of the cell. My sample code prints the various styles of borders that are available. The value used for the border is centered in the cell.

```
* Border options
      outline,oleft,oright,otop,obottom,
      shade,
      ocolor,lcolor,rcolor,tcolor,bcolor;

put '[Select("r2c1")]';
do i = 0 to 13;

outline=0;
oleft=i;    oright=i;   otop=i;     obottom=i;
shade=0;
ocolor=0;   lcolor=0;   rcolor=0;   tcolor=0;   bcolor=0;

put '[Border(' outline ',' oleft ',' oright ',' otop ',' obottom ','
      shade ',' ocolor ',' lcolor ',' rcolor ',' tcolor ',' bcolor ')]'
      '[Alignment(3)]';
put i &tab &tab;
if not mod(i+1,4) then put '[Select("r[2]c[-8]")]';
end;

put '[Select("r10C1")]';
```

Shading

There are eighteen shading patterns. This will display all of them and put the number associated with each one, centered in bold type in the cell.

```
put '[Select("r1c1")]';
do i= 1 to 18;
      put   '[Alignment(3)]' /
            '[FONT.PROPERTIES(,"bold")]' /
             '[PATTERNS(' i ')]' i &tab;
      if i = 9 then put '[Select("R[1]C[-9]")]';
end;

put '[Select("r5c5")]';
```

Fonts

Changing fonts and their properties is controlled by the “FONT.PROPERTIES” statement. The following comment describes the parameters that are available. If a parameter is blank or missing, that particular setting remains as it was. The first three are probably used the most. They control the font, the style of font, and size.

```
*FONT.PROPERTIES(font, font_style, size, strikethrough, superscript,
subscript, outline, shadow, underline, color, normal, background,
start_char, char_count);

Retain _Italic _Bold _Regular  $40 (
      '[FONT.PROPERTIES(,"italic")]'
      '[FONT.PROPERTIES(,"bold")]'
      '[FONT.PROPERTIES(,"regular")]'  );

put '[FONT.PROPERTIES("arial", "regular", "14")]';

put '[Select("R1C1")]';
put 'This is a regular heading with subscript and superscript' &tab;
put '[Select("RC[-1]")]';
put '[FONT.PROPERTIES("arial", "bold", "14", "FALSE", "FALSE", "TRUE",
, , "2", , , , "32", "9")]';
```

```
put '[FONT.PROPERTIES("arial", "bold italic", "14", "FALSE", "TRUE",
"FALSE", , , , , , , "46", "11")]';

put '[Select("R5C2")]';
put _Italic;
```

Miscellaneous Processes:

```
put '[Workbook.activate("Long Formulas")]';
put '[Copy("r1c14")]';
put '[Workbook.activate("Sheet1")]';
put '[Paste("r6c14")]';

* PASTE.SPECIAL(paste_num, operation_num, skip_blanks, transpose)
put ‘[PASTE.SPECIAL(2)]’ /* Pastes only the formula */
put ‘[PASTE.SPECIAL(3)]’ /* Pastes only the value */
put ‘[PASTE.SPECIAL(4)]’ /* Pastes only the format */
```

Saving and Quitting:

```
put '[Save()]';

put '[Save.As("c:\Test_Excel.xls")]';

put '[Quit()]';
```

Acknowledgements:

There are many great resources to learn more about using DDE with SAS. I would suggest a “google” search with the keywords “sas dde excel” or go to the SAS website and under customer support search “dde excel”.

Programs/Comments:

All of the programs are available for download at
http://www.ccsi.com/~jshoffman/DDE_Examples.zip

For comments or questions please feel free to contact me via email at
jshoffman@dbsicorp.com