

Using SAS DDE to Control Excel

George Zhu
Alberta Health, Government of Alberta

Edmonton SAS User Group Meeting
April 15, 2015

1 DDE: Dynamic Data Exchange

2 Examples and Demonstration

Demo 1: Import password protected Excel file

Demo 2: Read data from specific cells

Demo 3: Write date to specific cells

Demo 4: Format Excel reports

Demo 5: Run VBA macros

3 Troubleshooting

What is DDE

- DDE stands “Dynamic Data Exchange”
- DDE is communication protocol that lets PC-based (Microsoft) applications talk to each other
- Excel and Word are DDE-compatible applications
- The DDE mechanism in SAS enables SAS to control Excel or Word

Why DDE

Good for automation:

- Basically, what you can do with Excel/Word using keyboard and mouse can be replicated programmatically in SAS
- Only base SAS and MS Excel/Word are needed
- Most useful feature - customize the Excel worksheet or Word document programmatically
- DDE is far from obsolete - The backward compatibility requirement guarantees it is supported in later versions of Excel/Word
- You can run VBA through DDE

Why DDE is not popular

- DDE can be quite complicated and tedious to program
- Excel and Word use different programming languages for DDE
 - Excel: X4ML (Excel 4 Macro Language) - New features (like merge cells) after Excel 4 are not supported
 - Word: WordBASIC (a subset of MS QuickBASIC programming language)
 - Both are now replaced by VBA (Visual Basic for Applications), but VBA does not support DDE protocol

Examples and Demonstration

Get Excel/Word running

To use DDE,

- Server application (Excel/Word) needs to be fully up and running
- SAS gains control after Excel/Word launching

For SAS to gain control after launching Excel/Word, turn off the XWAIT and XSYNC options:

```
options noxwait noxsync;
```

Launch Excel/Word

There are several ways to get Excel or Word running from within SAS.

- Use “X” command

```
X "Start Excel";
```

- Use “%sysexec” statement

- Without opening an existing workbook

```
%sysexec "start excel";
```

- Opening an existing Excel workbook

```
%sysexec "examples.xlsx";
```

Let SAS wait for a few seconds until Excel is fully running:

```
data _null_;  
  rc=sleep(15); * wait for 15 seconds;  
run;
```


DDE syntax: Doublet vs Triplet style

DDE is specified through the "Filename" statement.

Two DDE formats:

- **Doublet:** - Main menu operations, such as file operations, worksheet selection, cell range selection, etc.

```
filename sas2xl dde "excel|system";
```

For doublet, it is always "excel|system"

- **Triplet:** - for accessing specific cell ranges

```
filename sas3xl dde "excel|[examples.xlsx]Sheet1!r1c1:r20c2"  
<DDE-options>;
```

Example 1: Import from password protected Excel file

A DDE Solution:

- Start Excel without opening the excel file
- Use open command to open the excel file. The syntax of the open command:

```
OPEN(file_text, update_links, read_only,  
      format, prot_pwd, write_res_pwd, ...)
```

- Use save.as command to save the excel file without the password. The syntax of the save.as command:

```
SAVE.AS(file_text, type_num, prot_pwd, backup,  
         write_res_pwd, read_only_rec)
```

- Import data from the password removed excel file
- Delete the password removed excel file

Example 1: Import from password protected Excel file

Make sure the file extension name matches `type_num` when saving a new file:

```
SAVE.AS(file_text, type_num, prot_pwd, backup,  
        write_res_pwd, read_only_rec)
```

- 51: .xlsx (Excel 2007-2013, without macro)
- 52: .xlsm (Excel 2007-2013, with or without macro)
- 56: .xls (Excel 97-2003 format)

Example 2: Read data from Excel in a specific range

Use DDE triplet-format to specific the range (notice the **notab** option):

```
filename exldata dde "excel|phds!r2c1:r7c7" notab;
```

Then use the infile statement with **delimiter='09'x** (Tab key)

```
infile exldata delimiter='09'x dsd;
```

Advantages of this data import method:

- Can handles mixed data type (numerical and character)
- Can import from irregular spreadsheet format

Example 3: Write data to specific cells

How?

- Use DDE triplet to specify output target places
- Need to use the “**notab**” option
- Use Tab key (ASCII value ‘09’x) to put data in next cell
- Remember SAS sends formatted value to Excel

Advantages of using DDE method:

- Only specific cell ranges are being updated
- Preserves custom format in the worksheet

Example 4: Formatting Excel Worksheet/Cells

Frequently used formatting operations:

- **Font Formatting:**

```
FORMAT.FONT(name_text, size_num, bold, italic,  
            underline, strike, color, outline, shadow)
```

- **Alignment**

```
ALIGNMENT(horiz_align, wrap, vert_align,  
          orientation, add_indent)
```

Some values for `horiz_align`, and `vert_align`:

- 1: General
- 2: Left
- 3: Center
- 4: Right

Example 4: Formatting Excel Worksheet/Cells

- Set column width

```
COLUMN.WIDTH (width_num, reference, standard,  
              type_num, standard_num)
```

The important parameter is `type_num`:

- 1: hide the column with `width=0`
 - 2: Unhide the column
 - 3: Set column width to best-fit
- Freeze panel

```
freeze.panes (true/false, col_split, row_split)
```

- `col_split`: Number of columns (from left) to be frozen
- `row_split`: Number of rows (from top) to be frozen

Example 4: Formatting Excel Worksheet/Cells

- Borders - for drawing tables

```
BORDER(outline, left, right, top, bottom, shade,  
       outline_color, left_color, right_color,  
       top_color, bottom_color)
```

Values for the line parameters:

- 0: No border
- 1: Thin line
- 2: Medium line
- 3: Dashed line
- 4: Dotted line
- 5: Thick line
- 6: Double line
- 7: Hairline

Example 5: Running VBA macros

Use the RUN command: Equivalent to click the Run button in the Macro dialog box:

```
put '[run("VAB_Macro_Name")]';
```

Make sure to enable VBA Macro function in Excel before running the SAS codes.

Demonstration:

- 1. Recode a macro for adding a title row
- 2. Remove the title row
- 3. From SAS run the DDE code to add the title row by revoking the recoded macro

DDE trouble shooting

- Common issue: **DDE session not ready**
 - Excel is not fully up and running when SAS sends control to Excel
 - Syntax error with X4ML statements
- Working wiht macro variables

Questions

Comments & Questions?