Paper 147-2007

7 Bulletproof Steps – The Quickest Way to Programmatically Generate a Dynamic Excel Graph from SAS®

Choon-Chern Lim, Mayo Clinic, Rochester, MN

ABSTRACT

If you are thinking, "Jeezz… not another way to create an Excel graph…", I would highly encourage you to continue reading this paper. While there are several proven techniques that allow you to programmatically create a dynamic Excel graph based on N observations from a SAS data set, they may not necessarily represent the optimal technique to achieve the best possible result. A fairly complex implementation could result more maintenance nightmares down the road.

This paper introduces an innovative way to tackle the same task in seven simple steps. This technique does not require any Visual Basic coding and you can rest assured that it will not give you any nightmares, so you can sleep soundly every night.

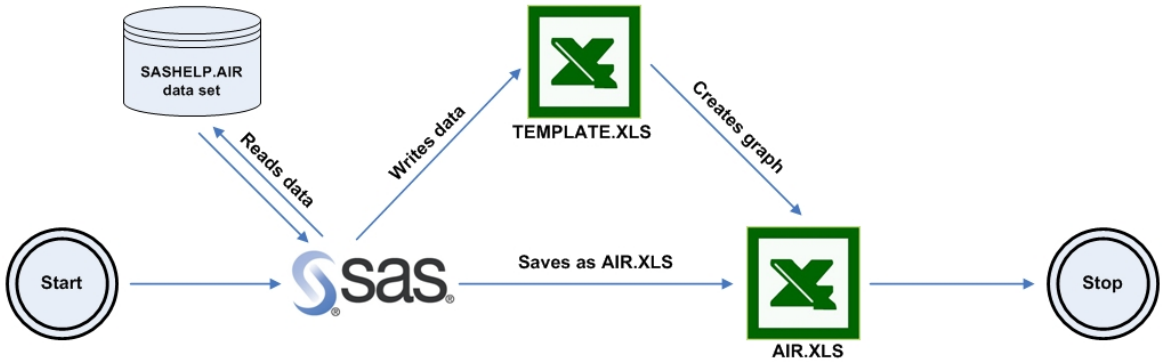
INTRODUCTION

In the SUGI 31 poster presentation, I presented paper 154-31 entitled "Step-by-Step in Using SAS® DDE to Create an Excel Graph Based on N Observations from a SAS Data Set" (2nd Best Paper Award). While the presented technique worked well for me at that time, I quickly realized that there is too much complexity in the overall process and it requires the use of the Visual Basics language to programmatically generate a dynamic Excel graph.

This may produce a maintenance nightmare to some SAS programmers. It makes future Excel graph modifications difficult as the unfortunate SAS programmer would vaguely understand the complex Visual Basics codes. Furthermore, the embedded Excel macro causes an annoyance because the MS Excel software would constantly prompt the user to either disable or enable the macro every time the Excel file is opened.

After spending months researching a better solution, I have finally crafted a better technique to overcome this concern. The goal of this paper is to present a simpler yet efficient way to produce the same Excel graph based on dynamic input data.

GENERAL OVERVIEW



In this paper, we use SAS to export the data from **SASHELP.AIR** data set into an Excel template file called **TEMPLATE.XLS** using Dynamic Data Exchange (DDE). The Excel template file generates an Excel graph based on the input data. SAS saves the Excel file as **AIR.XLS** and discards any changes made to the template file so that this template file can be reused again in the future.

THE SEVEN BULLETPROOF STEPS

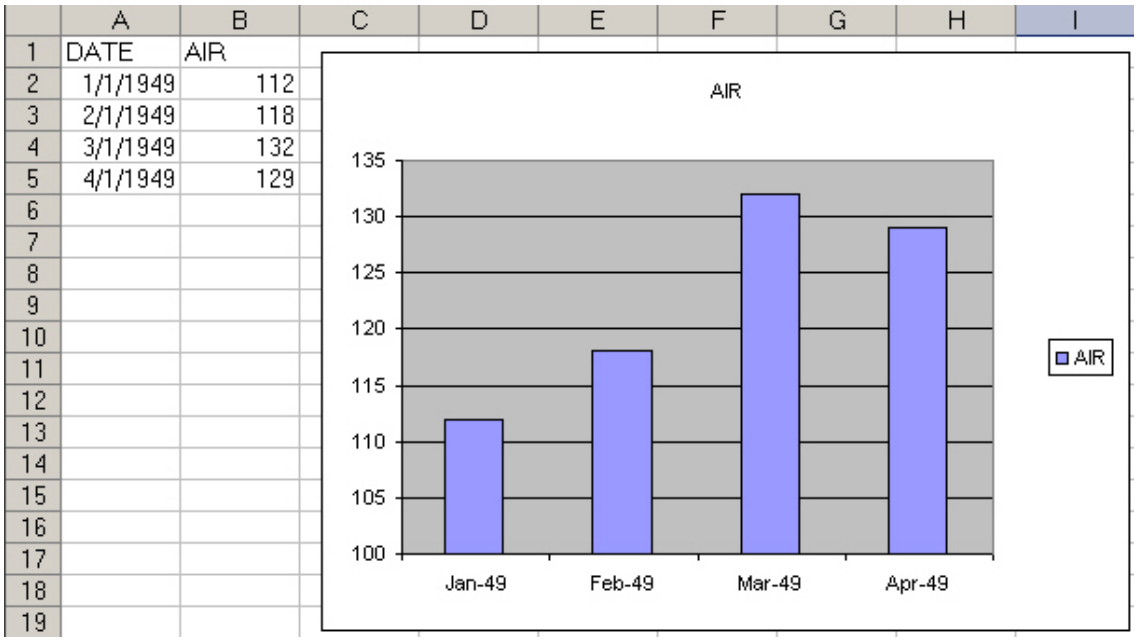
1. The first step is to create an Excel template file. To do so, we will run the below SAS codes to export four data rows from **SASHELP.AIR** data set into an Excel file. We also specify an Excel sheet name to be **GRAPHSHEET** so that we can easily reference this particular Excel sheet later on.

```
data sample;
      set sashelp.air(obs=4);
run;

proc export data=sample outfile="C:\sgf2007\template.xls" dbms=excel;
      sheet=GRAPHSHEET;
run;
```

The content of the generated Excel file should look as the following:-

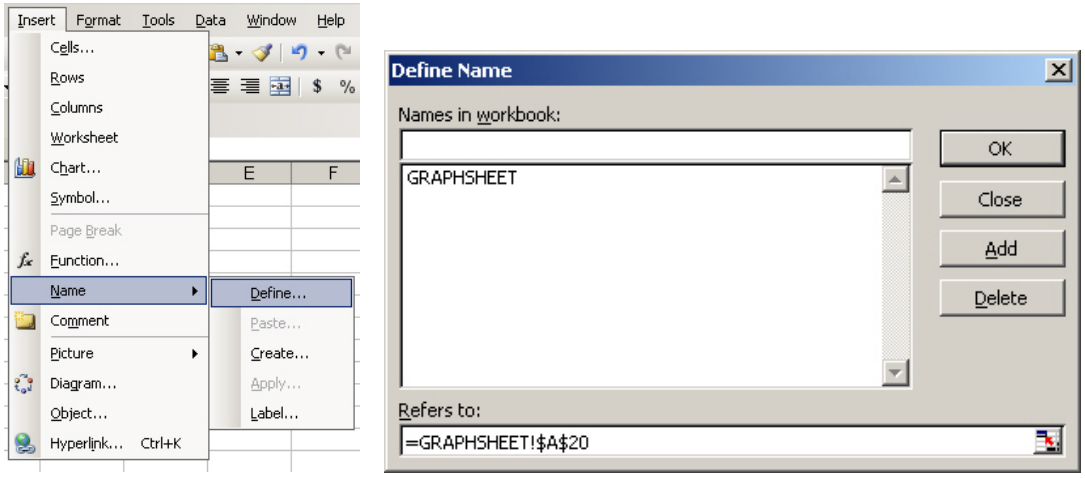
	A	B
1	DATE	AIR
2	1/1/1949	112
3	2/1/1949	118
4	3/1/1949	132
5	4/1/1949	129

2. Create a graph by highlighting all data (A1 to B5).
Then, select **Insert --> Chart --> Finish**.



3. Once the graph is created, we want to define a named Excel formula for each data column. Before we perform this step, please ensure that you have clicked anywhere outside the graph to deselect the graph first. Then, select **Insert --> Name --> Define**.

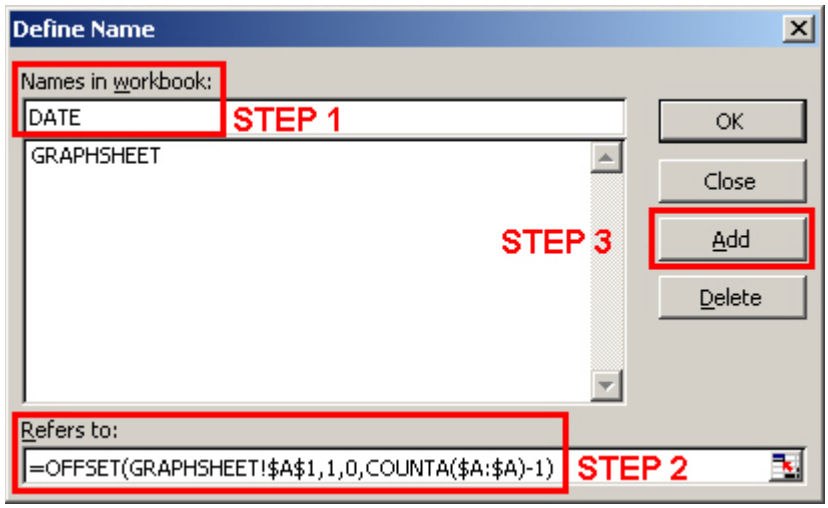
   It is necessary to deselect the graph first so that the **Insert --> Name --> Define** menu item will appear in the menu bar. Once selected, a **Define Name** dialog box will appear.



4. In the **Define Name** dialog box, enter the following information:-

   Names in workbook: DATE
   Refers to: =OFFSET(GRAPHSHEET!$A$1,1,0,COUNTA($A:$A)-1)

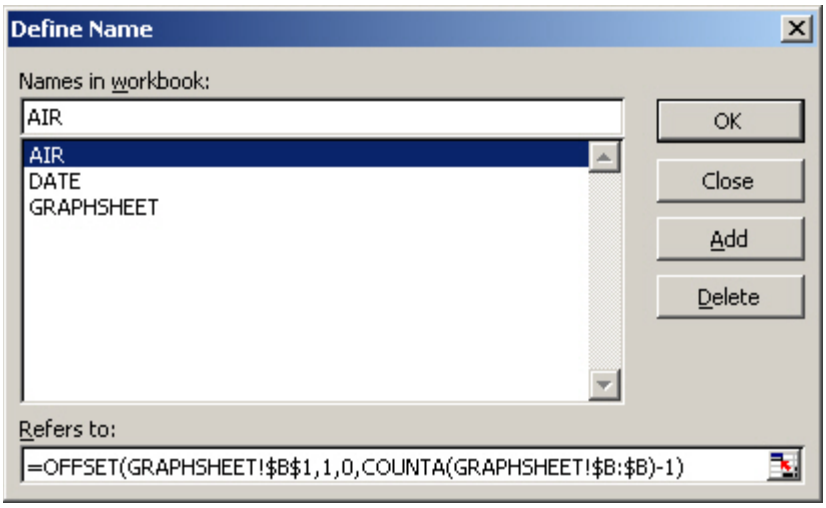
   Click **Add** button.



   Repeat the same for named formula AIR.

   Names in workbook: AIR
   Refers to: =OFFSET(GRAPHSHEET!$B$1,1,0,COUNTA($B:$B)-1)

The **Define Name** dialog box should look as the following:-



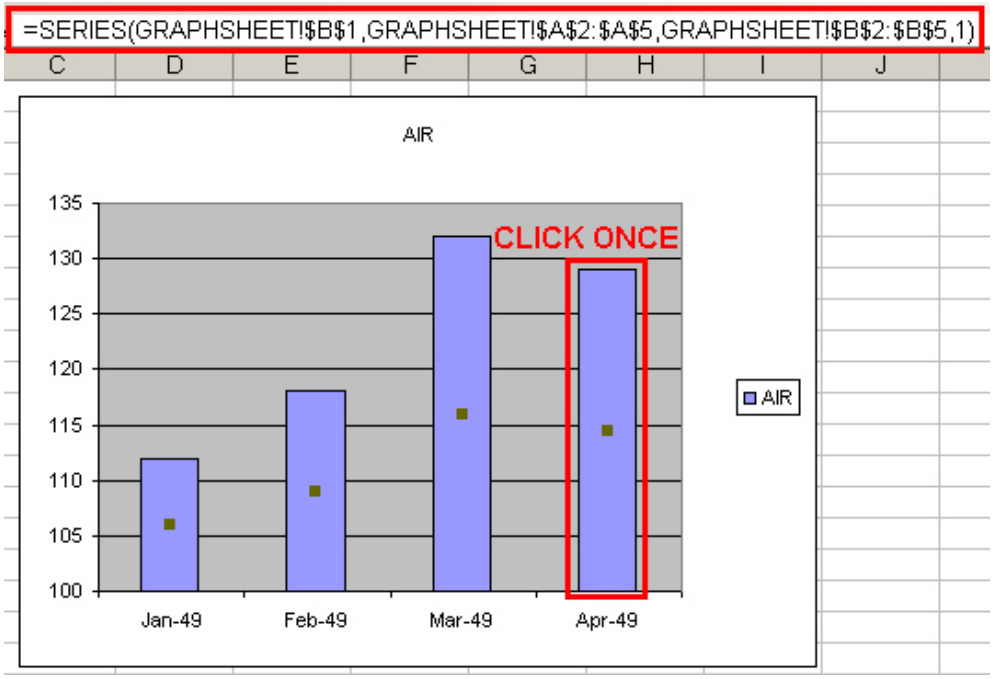
If the Excel graph requires more than two variables, then repeat these steps for column C, D, E, etc.

Finally, click **OK** button to close the **Define Name** dialog box.

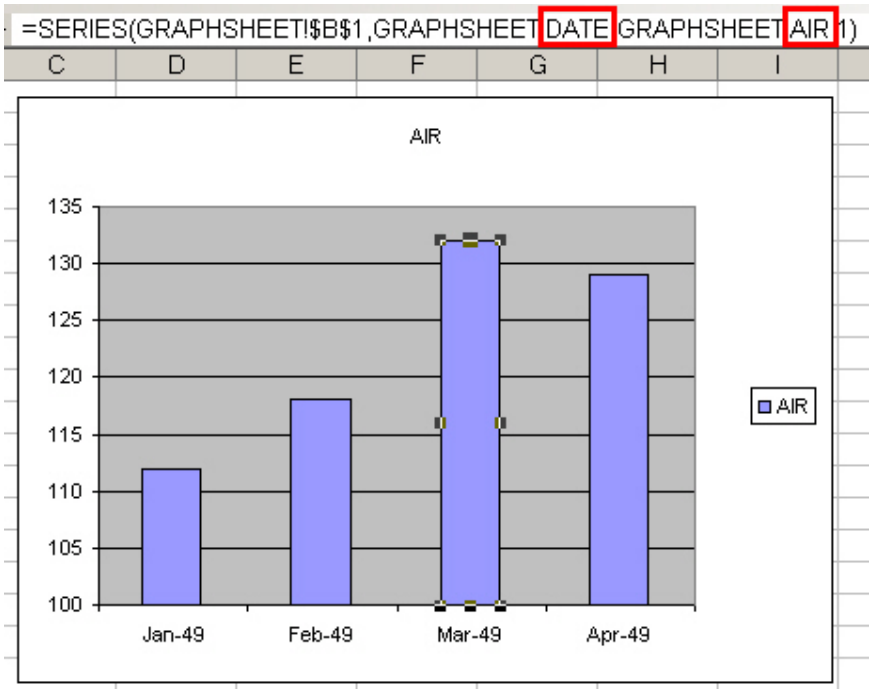
Each of this formula specifies a range that begins from one row below the respective column to one less than the populated rows in that column. When the data rows are added or removed, these ranges will automatically grow or shrink.

5. Now, we need to replace the hard coded data range for the generated graph with the named Excel formulas we have just created. This modification enables the graph to be automatically updated when the data rows are added or removed in the respective columns.

Click on any bar column of the graph once. You should see the following in the formula bar:-

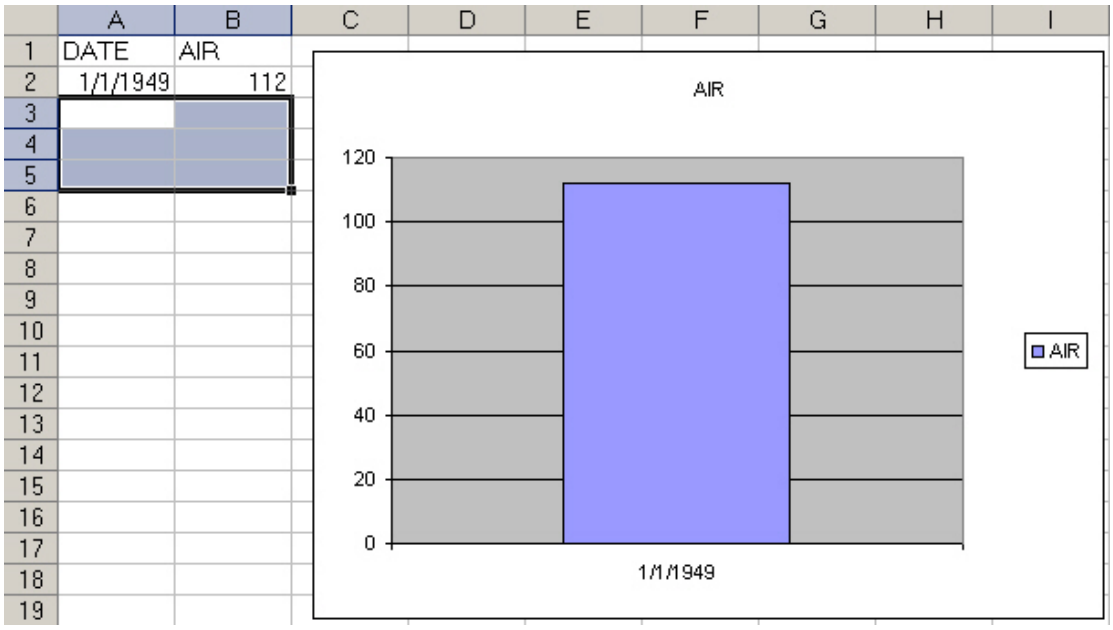


Substitute **$A$2:$A$5** with **DATE** and **$B$2:$B$5** with **AIR**. The modified new formula should look as the following:-

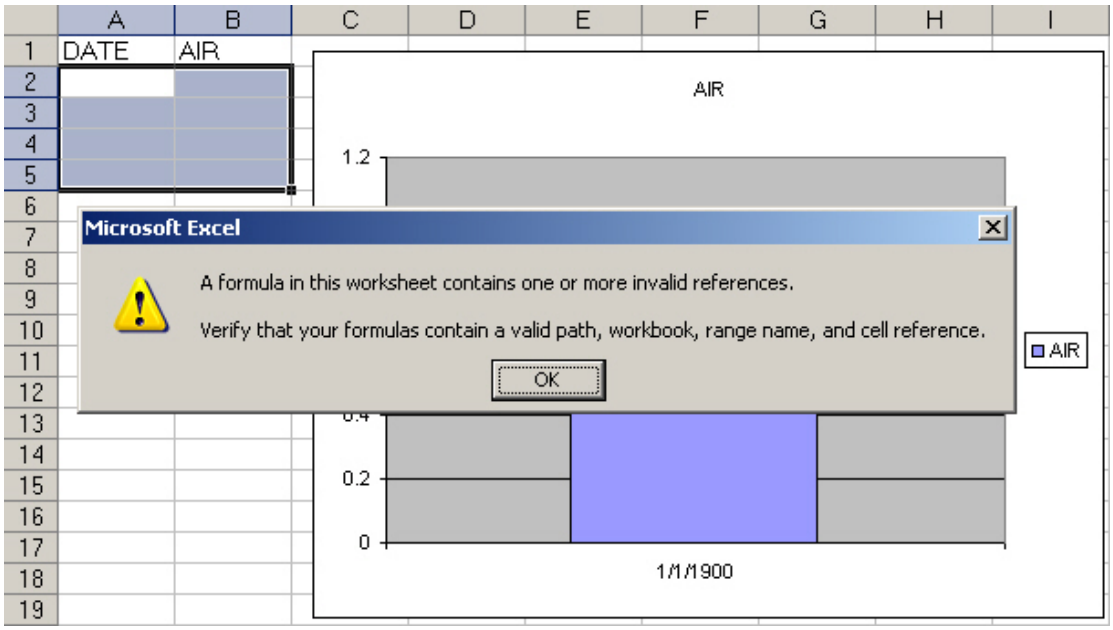


Then, select **ENTER** key to update the formula.

6. Delete all sample data except the first data line. Then, save this Excel file as **TEMPLATE.XLS** and close it.



If you attempt to delete all sample data, you will get the following error:-



7. Finally, run the SAS codes below. These SAS codes use DDE to write **DATE** and **AIR** data from **SASHELP.AIR** data set into column A and B of **TEMPLATE.XLS**. Then, this modified Excel file is saved as **AIR.XLS**. The changes made to **TEMPLATE.XLS** are discarded so that it can be reused again the next time.

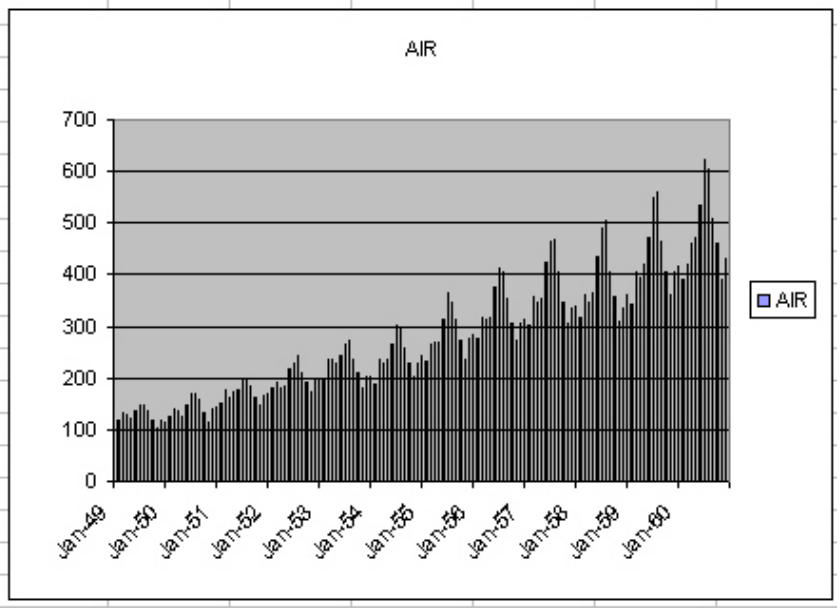
```
options mrecall mprint;

%macro createExcelGraph(lib=, dsn=);

      /************************************************************
       *** Excel application path.
       ************************************************************/
      %let xlsCmdPath = C:\Program Files\Microsoft Office\OFFICE11\excel.exe;

      /************************************************************
       *** Excel template file path.
       ************************************************************/
      %let xlsTemplatePath = C:\sgf2007\template.xls;

      /************************************************************
       *** Excel output file path.
       ************************************************************/
      %let xlsOutputPath = C:\sgf2007\air.xls;

      /************************************************************
       *** Get number of rows in data set and store the count in
       *** macro variable "total". The dictionary.tables is a special
       *** read-only table that contains plenty information about
       *** SAS data sets.
       ************************************************************/
      proc sql noprint;
            select nlobs into :total
            from dictionary.tables
            where libname = upcase("&lib.") and memname = upcase("&dsn.");
      quit;
```

```
      options noxsync noxwait;

      /*************************************************************
       *** Open the template file in the Excel application.
       *************************************************************/
      x "'&xlsCmdPath.' &xlsTemplatePath.";

      /*************************************************************
       *** Wait 5 seconds for Excel application to start.
       *************************************************************/
      data _null_;
            rc = sleep(5);
      run;

      /*************************************************************
       *** Determine the Excel cell range to write the data.
       *** The data starts on second row since first row contains
       *** variable names.
       *************************************************************/
      %let fromCell = r2c1;
      %let toCell = r%eval(&total. + 1)c2;

      /*************************************************************
       *** Create a reference to that specific cell range in the
       *** Excel sheet called GRAPHSHEET.
       *************************************************************/
      filename ddedata dde "excel|GRAPHSHEET!&fromCell.:&toCell." notab;

      data _null_;
            file ddedata;
            set &lib..&dsn.;
            put date mmddyy10. '09'x air '09'x;
      run;

      /*************************************************************
       *** Create another reference to the Excel application.
       *************************************************************/
      filename ddecmds dde "excel|system";

      /*************************************************************
       *** Save the Excel output file and close Excel application.
       *** The false/true error wrapper suppresses the file
       *** overwrite popup confirmation window if the specified
       *** output file has already exists.
       *************************************************************/
      data _null_;
            file ddecmds;
            put '[error(false)]';
            put "[save.as(""&xlsOutputPath."")]";
            put '[error(true)]';
            put '[file.close()]';
            put '[quit()]';
      run;

%mend createExcelGraph;

%createExcelGraph(lib=sashelp, dsn=air);
```

TESTS AND RESULTS

When the SAS code is executed, **AIR.XLS** is created and will look similar as the following:-.

	A	B
1	DATE	AIR
2	1/1/1949	112
3	2/1/1949	118
4	3/1/1949	132
5	4/1/1949	129
6	5/1/1949	121
7	6/1/1949	135
8	7/1/1949	148
9	8/1/1949	148
10	9/1/1949	136
11	10/1/1949	119
12	11/1/1949	104
13	12/1/1949	118
14	1/1/1950	115
15	2/1/1950	126
16	3/1/1950	141
17	4/1/1950	135
18	5/1/1950	125
19	6/1/1950	149



CONCLUSION

In order to accommodate constant requirement changes in the rapidly growing business, it is essential for us to simplify our implementations yet still sufficiently meet the customer requirements. While there are many ways to programmatically generate a dynamic Excel graph from SAS, this paper presents a simpler technique to achieve the same result.

REFERENCES

Choon-Chern Lim. 2006. Step-by-Step in Using SAS® DDE to Create an Excel Graph Based on N Observations from a SAS Data Set. Available http://www2.sas.com/proceedings/sugi31/154-31.pdf.

Microsoft Corporation. 2006. Using named ranges to create dynamic charts in Excel. Available http://office.microsoft.com/en-us/assistance/HA011098011033.aspx.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Choon-Chern Lim
Mayo Clinic
200 First Street SW
Rochester, MN 55905
Email: limc@mayo.edu