

## SAS® & VBSCRIPT FOR GENERATION OF POWERPOINT PRESENTATION

Suhas R. Sanjee, MaxisIT Inc., Edison, NJ  
Christopher Tong, Merck and Co., Rahway, NJ

### Abstract

Some analysis reports are presented to the stake holders as PowerPoint presentations. The process of organizing tables and graphs in a PowerPoint presentation is laborious and it needs to be done several times during the trial. A technique is presented in this paper which uses SAS/Graph and Windows Scripting Host (WSH) to automatically generate a PowerPoint presentation whenever the data is updated. This way, the presentation can be updated with any changes in tables / graphs by a simple macro call in SAS. WSH is a simple, powerful, and comprehensive tool which facilitates the controlling of Microsoft applications like Word, PowerPoint, and Excel programmatically. The WSH allows VBScript programs to run on the Windows operating system as stand-alone applications. SAS and WSH can be used to automate routine tasks performed using MS Office. VBScript can be generated through a SAS program which can then be executed using WSH to perform various tasks in MS Office and other Windows based applications. The graphs generated during analysis are stored as images. Tables can be converted into SAS/Graph objects using the GPRINT procedure, which are then saved as images. A SAS macro is then used to generate a VBScript which in turn populates a PowerPoint presentation with the image files.

### Introduction

VBScript (Visual Basic Scripting Edition) is an active scripting language developed by Microsoft. It is a limited variation of Microsoft's Visual Basic programming language. VBScript is installed by default in every desktop release of the Windows Operating System. VBScript comes in handy as an automation tool and has been widely used by Windows administrators.

A technique to automatically generate PowerPoint presentations containing relevant tables and graphs is presented here. Graphs and tables are generated by using SAS procedures GPRINT, GSLIDE, GPLOT and TABULATE. These are then converted to image files which are programmatically inserted into a PowerPoint presentation using VBScript. More details on producing graphs, converting tables to graph objects and juxtaposing graphs and tables can be found in our previous work [Sanjee et al., 2007].

### Data Overview

This technique can be used on data collected from any clinical/pre-clinical trial. The only prerequisite is that the results have to be stored as image files. Graphs generated from any software can be automatically populated in a PowerPoint presentation. The graphs can be arranged in any desired order, e.g., alphabetical order of their filenames, date of creation, and etc. This technique was successfully implemented on data resulting from longitudinal animal studies.

## SAS Programming Overview

The steps involved in the proposed technique are discussed from a SAS programming perspective in this section.

### Initialization

The SAS system needs to be initialized to produce the desired graphs and tables. This is done with statements OPTIONS and GOPTIONS.

The OPTIONS statement specifies one or more SAS system options to be changed. The OPTIONS statement can appear anywhere in your SAS program except within datalines. The change that is made by the OPTIONS statement remains in effect for the rest of the job, session, SAS process, or until you issue another OPTIONS statement to change the options again. You can specify the SAS system options through the OPTIONS statement, through the OPTIONS window, at SAS invocation, and at the initiation of a SAS process.

```
OPTIONS <option(s)>;
```

FORMCHAR= option is used here to produce solid border lines around output for PROC TABULATE.

```
FORMCHAR = <formatting-characters>
```

Formatting characters are used to construct tabular output outlines and dividers for various procedures, such as the CALENDAR, FREQ, and TABULATE procedures. If you omit formatting characters as an option in the procedure, the default specifications given in the FORMCHAR= system option are used.

The GOPTIONS statement can appear anywhere in your SAS program. Options specified in a GOPTIONS statement remain in effect until you use another GOPTIONS statement to change them or you end the session [*SAS/GRAPH User's Guide*]. Syntax is as follows:

```
GOPTIONS <options-list>;
```

DEVICE= option is used here to specify the format of the stored graphs.

```
DEVICE=drivername
```

tells SAS/GRAPH what graphics device you are using. If you are running SAS/GRAPH interactively and *no* device name has been specified, you are prompted to enter a device name. SAS Graphs can be exported to JPEG, CGM, BMP and other image formats by specifying an appropriate DEVICE.

**Fig 1: Initialize SAS System and SAS/GRAPH**

```
*Specify appropriate option for FORMCHAR to produce solid borders in PROC  
TABULATE output;  
options formchar='82838485868788898A8B8C2B3D7C2D2F5C3C3E2A'x;  
  
*Set GOPTIONS to export SAS graph objects to BMP files;  
GOPTIONS DEVICE=SASBMP;
```

## Generation of graphs

Graphs that are used for the presentation should first be stored as image files. This can be done by using appropriate graph options followed by the invocation of SAS/ Graph procedures such as GPLOT, GCHART, and etc. The list of folders containing the graphs is used as input to the SAS macro to automatically generate the PowerPoint presentation.

Results used for presentation can be produced using any of the statistical packages like SAS, R, Excel, Matlab, and etc. The only limitation is that these graphs should have been stored as image files.

## Convert tables to Graph objects

The SAS macro is programmed to insert tables and graphs into PowerPoint as images. Hence, tables that need to be included in the presentation should be converted to graph objects so that they can be stored as images.

Tables generated using PROC TABULATE are converted to SAS/Graph objects using PROC GPRINT. More details on converting tables to graphs can be found in [1].

## Export SAS/ Graph objects to images

Image files of the desired resolution and format can be created from SAS/ Graph objects by using SAS/GRAPH device drivers along with FILENAME and GOPTIONS statements.

- The FILENAME statement defines a fileref to be associated with one or more graphics stream files.
- The GOPTIONS statement specifies graphics options related to the creation of the graphics stream files.

## SAS® Macro for Generation of PowerPoint Presentation

This section will focus on the steps involved to programmatically generate the contents of VBScript using SAS which in turn creates a PowerPoint presentation with desired tables and graphs.

**Step 1:** Read the Excel file containing the folder names under which the image files of tables and graphs are stored.

**Step 2:** A list of all the image files under each folder mentioned in the Excel file is generated along with the path using the X statement and DIR command as illustrated below.

**Fig 2: Creating File List**

```
*Generate DOS command strings;  
%let cmd="dir "c:\study1\folder1\*.tiff" > c:\study1\pptlist\folder1.txt" /b';  
X "&cmd";
```

The DOS command string “cmd” is dynamically generated for each folder in the Excel file. The output of DIR by default contains filename and other details like date, time, and etc. The option “/b” in the DIR command is used to suppress these details and generate a list of files in bare format. Execution of each of these commands results in a text file folder<n>.txt. All these text

files are then converted to SAS datasets and concatenated to get a comprehensive list of all the image files that are to be included in the PowerPoint presentation. This SAS dataset can now be sorted to include files in the required order.

**Step 3:** A PowerPoint presentation is created by assigning appropriate values to object properties as shown in Fig 3. “Filelist” is the name of the SAS dataset containing all the image files generated as described in Step 2. Each image file is then inserted as a PowerPoint slide through the AddSlide subroutine in VBScript. The PowerPoint presentation is then saved in desired location.

**Fig 3: Generation of VBScript**

```
DATA _NULL_;
  set filelist end=eof;
  file "&protopath/studyfiles/&pptname..vbs" lrecl=2048;

  *Initialization - Create the presentation file;
  if _N_=1 then do;
    put "%str(Set objPPT = CreateObject%)" @ ;
    put '"PowerPoint.Application"' @; put "%str(%))";
    put "%str(objPPT.Visible = True)";
    put "%str(Set objPresentation = objPPT.Presentations.Add)";
    put "%str(strComputer = %".%")";
  end;

  *Add a slide;
  put "%str(Set objSlide = objPresentation.Slides.Add%(1, 2%))";
  put "%str(Set objShapes = objSlide.Shapes)";
  put "%str(objShapes.AddPicture %)" name
    "%str(%", False, True, 20, 20, 670, 505)";

  *Save the presentation and quit powerpoint;
  if eof=1 then do;
    put "%bquote(objPresentation.SaveAs("&pathwin\studyfiles\&pptname"))";
    put "objPresentation.Close";
    put "objPPT.Quit";
  end;
run;
```

The macro outputs a VBScript file which generates a PowerPoint presentation when executed.

### **Conclusion and Future Directions:**

This technique has been successfully implemented on the Windows platform. The possibility of using this technique in UNIX has to be investigated. The same technique can be used to automate any routine task performed using MS Office and other Windows based applications.

### **Acknowledgements:**

We would like to thank Amy Gillespie, Merck and Co. and the management of MaxisIT Inc. for all the encouragement and support.

## References:

1. S. Sanjee, S. Zhang, "Juxtaposition of Tables and Graphs Using SAS® /GRAPH Procedures", SAS Conference Proceedings: NESUG 2007 held at Baltimore, MD November 11-14 2007.

Author Name: Suhas R. Sanjee  
Company : MaxisIT Inc.  
Address : 1199 Amboy Ave  
City: Edison State: NJ ZIP: 08830  
Email:sanjee@maxisit.com

Author Name: Christopher Tong  
Company : Merck & Co., Inc.  
Address : RY33-300, 126 E Lincoln Ave  
City: Rahway State: NJ ZIP: 07065  
Email:christopher\_tong@merck.com