

VBScript-Driven Automation in SAS[®] : A Macro to Update Text in a Microsoft[®] Word Document at Preset Bookmarks

Shayala Williams Gibbs, Rho[®], Inc., Chapel Hill, NC

ABSTRACT

SAS can harness the power of Microsoft (MS) Visual Basic Scripting Edition (VBScript) to programmatically update MS Office documents. This paper presents a macro developed in SAS to automate updates to a MS Word document. This useful macro invokes VBScript to pass text directly from a SAS data set into a predefined bookmark in an existing Word document.

INTRODUCTION

SAS programmers are often called upon to compile data from one or many data sources and use the data to update reports. The reports we help produce often need to be updated over time or reproduced repeatedly for specific subsets of data. Compiling data in SAS and then manually updating a template report can risk data entry error. In years past, the Dynamic Data Exchange was a commonly used method to update MS Office documents using SAS, mitigating the human error aspect of updates to the reports. More recently, VBScript has been used to facilitate the interaction between a SAS program and a MS Office program in a MS Windows environment.

VBScript is a lighter version of MS Visual Basic. VBScript files are text files that end in a .vbs extension. VBScript files allow SAS programmers to perform commands in MS Office that cannot be directly accomplished in SAS. One of the advantages of VBScript is that these programs speak the same language as MS Office applications. The use of VBScript can greatly enhance the capabilities of using a SAS program to control Office applications. Sanjee, Ganesan, and Clark (PharmaSUG Proceedings, 2011) presented a useful white paper describing many of the VBScript commands used in the macro presented as a mechanism for automating updates to Word documents, including CreateObject, Open, GoTo, and TypeText. The syntax has been adapted to accomplish passing text from a SAS data set into a Word document at pre-defined bookmarks. Many more descriptions and samples of the functions and keywords that make up VBScript can be found in the VBScript Language Reference on the Microsoft Developer Network website.

The SAS macro presented in this paper will create a VBScript program to open an existing MS Word file with pre-defined bookmarks, insert the text from a SAS data set at each corresponding bookmark, and save the Word document with a new name. The program will then execute the VBScript program. Finally, the macro will remove the VBScript program after it is executed. The macro requires MS Word to be installed locally on the PC. This macro was tested in SAS version 9.3 and MS Office 2010.

STEPS TO UPDATE THE MS WORD FILE USING VBSCRIPT

1. Setup a template report in MS Word.
2. Insert bookmarks in the Word template report where text will be updated from SAS.
3. The bookmark names and corresponding text for insertion must be made into a SAS data set. The data set can be created in the program prior to invoking the macro or in another program that generated a permanent SAS data set.

THE MACRO CALL AND PARAMETER DEFINITIONS

The Macro VBScript_SAS_to_MSWord has six parameters, and all are required. The macro parameters defined are:

- Path – Must be a fully qualified path name pointing to the template MS Word document file
- TemplateFile – Name of the MS Word file with pre-set bookmarks for text insertion, valid Word file name (i.e. with .doc, .docx, or .rtf file extension)
- SavedFile – A valid MS Word filename. Will be saved in the location set in the PATH parameter name (i.e. with .doc, .docx, or .rtf file extension)

- InDataset – A SAS data set name containing a minimum of two variables: one with the bookmark name and another with text for insertion at the corresponding bookmark
- BookmarkVar – Name of character variable in the data set defined in the INDATASET parameter containing bookmark name
- TxtInsertVar – Name of character variable in the data set defined in the INDATASET parameter containing text to be inserted at corresponding bookmark

ACTIONS PERFORMED BY THE MACRO

The actions performed by the VBScript_SAS_to_MSWord are detailed step by step.

Read Dataset Containing Bookmarks and Text for Insertion

The first action of the macro is to read in the data set specified in the parameter InDataset.

```
data TempInsert; set
    &InDataset;
    BookMrk = strip(&BookmarkVar.);
    TxtInserted = strip(&TxtInsertVar.);
    keep BookMrk TxtInserted;
run;
```

This step creates a new data set in the working directory which contains only the bookmark variable and the insertion text variable, stripped of leading and trailing blanks. The purpose of this step is to clean up the data set that will pass through the VBScript, keeping only the two variables passed to the macro by the BookmarkVar and TxtInsertVar parameters.

Setup VBScript File

Next, the text file for the VBScript is defined. The file must have a .vbs extension.

```
filename script "&Path.\LivingDoc.vbs";
```

Once created, the VBScript file will be temporarily saved as LivingDoc.vbs in the same directory in which the template Word document and final updated Word document are saved.

Open an Existing Word File with Bookmarks Set

A DATA _NULL_ statement is used to develop the VBScript program in SAS. Before the program reads the first record from the bookmarks data set, Word automatically opens the existing file specified by the Path and TemplateFile parameters.

```
data _null_;
set TempInsert
end=eof; file script;
if _N_=1 then do;
    put "%str(Set objword = CreateObject%())" @ ;
    put "'Word.Application'" @; put "%str(%))";
    put "%str(objword.Visible = True)";
    put "%str(strComputer = %".%")";

    put "%str(Set objDoc = objWord.Documents.Open(%"&Path.\&TemplateFile.%"))"; put
"%str(Set objSelection = objWord.Selection)";
end;
```

Go to Bookmark and Insert Text

Next, build the VBScript that will command Word to go to the specified bookmark and insert the desired text contained in the input SAS data set. The program will repeat this action for each record in the SAS data set.

```
vbtypetext = 'objSelection.TypeText "' || strip(TxtInserted) || '"';

if missing(Bookmrk) ne 1 then
  put "%str(return=objSelection.GoTo%(-1,,%) " Bookmrk
    "%str(%")"); put vbtypetext;
```

A new variable VBTYPETEXT is created in the temporary TypeText data set which builds the VBScript syntax that will place the desired text at the corresponding bookmark. If the bookmark is not missing, the program will send the text to the bookmark location in the document.

Save the File with New Name; Close Word

The final lines of code written to the VBScript file save the updated Word document as a new file, close the file, and quit Word. These commands are written after last record in the input data set TypeText is read.

```
if eof=1 then do;
  put "%str(objDoc.SaveAs(%"&Path.\&SavedFile.%"))";
  put 'objDoc.Close';
  put
    'objWord.Quit'; end;
run;
```

The Word file is saved as a new document in the location and with the filename specified by the PATH and SAVEDFILE parameters.

Execute and Then Delete the VBScript

Finally, the VBScript built by our SAS program is ready to run. An X command kicks it off, sending DOS code to the command prompt. After the script runs, another X command deletes the temporary LivingDoc.vbs file that we created.

```
X "%str(cscript.exe %"&Path.\LivingDoc.vbs %")";

%if %sysfunc(fileexist(&Path.\LivingDoc.vbs)) %then
  %do; X "del %"&Path.\LivingDoc.vbs" ";
%end;
```

The file LivingDoc.vbs is an executable program. Simply clicking on the file while in the directory where it is temporarily saved would cause the program to open up the Word template document and save it again, perhaps inadvertently. For this reason, it is a good idea to delete the VBScript once the desired updates to the Word template are made and saved.

SAMPLE MACRO CALL

The macro call below will open an existing document "C:\MyDocuments\Study Report Template.rtf" and insert the text from the variable Tolinsert

```
%VBScript_SAS_to_MSWord (
  Path           = C:\MyDocuments,
  TemplateFile   = Study Report Template.rtf ,
  SavedFile      = New Study Report.rtf ,
  InDataset      = StudyData.ToBeInserted ,
  BookmarkVar    = BookMark,
  TxtInsertVar   = ToInsert )
```

CONCLUSION

The macro presented herein can be used to automate updates to text in a template MS Word document. It is also possible to insert graphics files into bookmarks in Office documents using similar methodology. In the future, the macro can be expanded to accomplish routine updates to both text and graphics in a template document.

REFERENCES

Grandy, Rose. 2013. "Improve Your ODS Experience with These Essential VBScript Tools." *Proceedings of the Midwest SAS Users Group 2013 Conference*. Available at <http://www.mwsug.org/proceedings/2013/RF/MWSUG-2013-RF08.pdf>.

Microsoft Corporation. "Microsoft Developer Network: VBScript Language Reference." June 23, 2014. Available at [http://msdn.microsoft.com/en-us/library/d1wf56tt\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/d1wf56tt(v=vs.84).aspx).

Sanjee, Suhas R., Ganesan, Rajavel, Clark, Jason. 2011. "Automated Generation of Clinical Study Reports." *Proceedings of the PharmaSUG 2011 Conference*. Available at <http://www.pharmasug.org/proceedings/2011/PO/PharmaSUG-2011-PO11.pdf>.

Stetz, Mark. 2000. "Using SAS® Software and Visual Basic for Applications to Automate Tasks in Microsoft Word: An Alternative to Dynamic Data Exchange." *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*. Available at <http://www2.sas.com/proceedings/sugi25/25/ad/25p021.pdf>.

ACKNOWLEDGMENTS

I would like to thank my colleagues Spencer Childress, Rita Slater, and Henry Bahnson for their assistance in coming up with the idea to pursue VBScript as a method to use SAS to update Word documents.

CONTACT INFORMATION

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

Name: Shayala Gibbs
Enterprise: Rho, Inc.
Address: 6330 Quadrangle Drive
City, State ZIP: Chapel Hill, NC 27517
Work Phone: 919-595-6860
E-mail: Shayala_Gibbs@rhoworld.com
Web: www.rhoworld.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.