Paper CC06

# SAS® Can Do it for You!

## - Automate and Customize Printing of Word Files in Windows® Environment

Jay Zhou, Amylin Pharmaceuticals, Inc. San Diego, CA

**ABSTRACT**

During the data analysis and reporting stage of a clinical study, programmers often need to print clinical reports, such as tables, listings, and graphs, for review. This can be cumbersome and time consuming, if it involves hundreds of Word files, especially when printing a specific page for each file. To meet the need, a SAS macro has been developed to automate and customize printing of Word files in the Windows environment. This paper reviews the capabilities of the application, some common uses for it, and the programming techniques implemented.

**INTRODUCTION**

With the growing popularity of SAS ODS (Hamilton, 2003; Smoak, 2004) and other homegrown applications (Peszek, et al., 1999; Zhou, 2001 and 2002a; Qi and Zhang, 2003), more and more SAS programmers generate Word documents or outputs in RTF format directly from SAS. For a clinical trial study, the number of files of tables, figures, and listings can range from several dozens to over hundreds depending on the size of a study. A programmer is often asked to print all the files out for review, which can be burdensome if there is no efficient way to do it.

There are different ways to print Word files in the Windows environment. You can simply open a file in Word and print it directly from Word. But, this is not practical when dealing with hundreds of files. A better solution is to select all the files in Microsoft Explorer, select the **Print** command from the **File** drop-down menu or right-click the mouse and then click on **Print**. However, this can be very slow and will freeze the computer, especially if the files are large, because the selected files must be opened in Word before printing them one by one. To avoid this, Cowmeadow (2004) mentioned the method of using the Microsoft Binder by selecting, and then dragging and dropping all the files from Microsoft Explorer into the left panel of the Binder, then printing. This allows all the files to be printed together with one print command. But this solution is limited by the availability of Microsoft Binder and it does not provide the flexibility of printing only certain pages. Also, the entire process is not automated and the files are not necessarily in the desired order.

The point I want to make here is that none of the available techniques is efficient and flexible enough to do a mass printing with a large amount of Word files. The ideal solution of printing files is expected to be automatic, customized, flexible, and effective. This is especially true if you are asked to print the files with specific pages once they are generated. For instance, let's assume you want to QC the page layout, column displays, formats, titles, and footnotes for all the listings against the mock listings. To do this effectively, you would like to just compare the mock listings with the first page of the production listings. So, ideally only the first page of each listing should be automatically printed out once the listings are generated. If the job is huge, you can run a batch job before leaving for the day and have the hard paper copies next morning to work with.

SAS provides us a unique solution via Dynamic Data Exchange (DDE) to meet this challenge. With the DDE as a bridge, SAS can directly communicate WordBasic command with Microsoft Word to ask it to do certain things. Since there are many references available in the discussion of what DDE is and how to access it from Base SAS code (Zhou, 2002b), there is no need to repeat them here. This paper will introduce a SAS macro and merely paraphrase the techniques of how SAS can control the printing process in Word.

**DESIGN OF THE MACRO %PRINT**

**WORDBASIC CODE**

When printing a Word file manually with any customized need, you can click **Print** on the **File** drop-down

menu to open the **Print** dialog box and define the settings. This manual action is supported and automated with VBA (Visual Basic for Applications), which is the language supporting Word 97 or above. Since SAS is not able to communicate any VBA statements via DDE with Word, you have to use WordBasic to substitute VBA. There are two WordBasic commands that can control Word to print files:

**FilePrintDefault** prints the active document using the current settings in the **Print** and **Print Setup** (Word version 6.0) dialog boxes (from **File** menu) and on the **Print** tab in the Options dialog box (from **Tools** menu). Since this command only allows you to print the active document with the current settings, it will not give users the flexibility to customize the printing process. Therefore, this option is not desirable.

**FilePrint** prints all or part of the active document or a document you specify. The arguments for the **FilePrint** statement correspond to the options in the **Print** dialog box (from **File** menu). Since this command gives the freedom of customizing the printing process, let's discuss it in more detail. Here is the syntax of **FilePrint**:

**FilePrint** [.Background = number] [, .AppendPrFile = number] [, .Range = number] [, .PrToFileName = text] [, .From = text] [, .To = text] [, .Type = number] [, .NumCopies = number] [, .Pages = text] [, .Order = number] [, .PrintToFile = number] [, .Collate = number] [, .FileName = text] [, .OutputPrinter = text]

The arguments and their explanations, which can be found from Word 95 WordBasic Reference (Microsoft Press, 1995), are listed below:

| Argument | Explanation |
|---|---|
| .Background | If 1, the macro continues while Word prints the document. On the Macintosh, this argument is not available and generates an error. |
| .AppendPrFile | If you print to a file, specifies whether to overwrite or append to the file if it already exists: 0 for Overwrite, 1 for Append. |
| .Range | The page range:<br>0 Prints the entire document<br>1 Prints the selection<br>2 Prints the current page<br>3 Prints the range of pages specified by .From and .To<br>4 Prints the range of pages specified by .Pages |
| .PrToFileName | If you print to a file, the path and filename of the file to print to. This argument is not available on the Macintosh unless QuickDraw GX is installed. |
| .From | The starting page number when .Range is 3. |
| .To | The ending page number when .Range is 3. |
| .Type | The item to print:<br>0 Document<br>1 Summary Information<br>2 Annotations<br>3 Styles<br>4 AutoText Entries<br>5 Key Assignments |
| .NumCopies | The number of copies to print. |
| .Pages | The page numbers and page ranges you want to print, separated by commas. For example, "2, 6-10" prints page 2 and pages 6 through 10. |
| .Order | Further delimits the range of pages to print:<br>0 Prints all pages in the range.<br>1 Prints only odd pages in the range.<br>2 Prints only even pages in the range. |
| .PrintToFile | If 1, sends printer instructions to a file. Make sure to specify a filename with PrToFileName. This argument is not available on the Macintosh unless QuickDraw GX is installed. |
| .Collate | If 1, organizes pages when printing multiple copies of the document. |
| .FileName | The path and filename of the document to print. If omitted, Word prints the active document. |
| .OutputPrinter | On the Macintosh, if QuickDraw GX is installed, specifies the printer to print to. If the specified printer is not valid, Word prints to the printer selected in the Print dialog box or to the default desktop printer. |

For the purpose of automating the printing process, the arguments pertinent to you are .Range, .Pages, .From, .To, .NumCopies, and .Filename. These arguments can be macrotized in order for users to customize the printing.

#### SAS MACRO PARAMETERS

To let SAS control this printing process, the macro should be flexible enough to meet the basic needs of selecting the files for printing and defining the number of copies and page ranges. With these needs in mind, there are five keyword parameters designed in the macro to provide customized specifications that allow programmatic control of the printing process in Word. The full macro specification is as follows with the defaults shown:

```
%print(dir=, file=*.doc, page=1, copy=1, forder=);
```

The meaning and usage of the macro parameters are thus:

**`dir`**: the only required parameter. It specifies the path to the location where the file(s) reside(s), e.g., `dir=c:\sugi\output`.

**`file`**: specifying the name(s) of file(s) to be printed. It can handle more than one file. For instance, to print out both the table (`t_demog.rtf`) and listing (`l_demog.doc`) files for the demographic data, the parameter can be defined as `file=t_demog.rtf l_demog.doc` by separating the names with a blank space. The wild card file name is accepted, e.g., `file=t*.rtf` will print out all the RTF files prefixed with 't'. The extension `.rtf` is required to be specified if a file is in RTF format, otherwise, the macro will assume the file is in Word document with the extension `.doc`. By default, the value for this parameter is `*.doc` which will print out the Word document files for a given directory.

**`Page`**: defining the specific page(s) to be printed, e.g., `page=all` to print all pages or `page=3-10` to print pages from 3 to 10. By default, the first page will be printed.

**`copy`**: specifying the number of copies to be printed. By default, one copy will be printed.

**`forder`**: defining the path and name of the external file or SAS data set that contains the order of the files.

#### ALGORITHM

The following algorithm is designed for the macro to meet the basic needs:

Get the list of files to be printed based on the definition given by the **`file`** parameter;
Put them in order if the **`forder`** parameter is specified;
Iterate the printing process from the first file to last file;
  Open Word when Word is not active;
  Instruct Word to print the targeted page(s) or the whole file based on the definition given by the **`page`** parameter.
End;

## IMPLEMENTATION

Based on the algorithm mentioned in the previous paragraph, this section will discuss the SAS code for implementing the design of the major part of the macro in detail.

#### GET THE LIST OF FILES FOR PRINTING

By default, the **`%print`** macro will print all the files with the file extension `.doc` for the given directory defined by the **`dir`** parameter. There are different ways to get the file names along with the file extensions from a given directory. To meet the needs, you can simply use the SCL (SAS Component Language) functions in a data step to create a data set, `_file`, which only contains two variables, `file`

(file names with file extension) and `filename` (without the file extension), for future use (see the code below). This code will be executed only when the **`dir`** parameter is defined with the wildcard value, e.g., `*.rtf` or `*.doc`.

```
data _file(keep=file filename);
   length file $50 filename $40;
   rc=filename('dir',"&dir");
   dirid=dopen('dir');
   do i=1 to dnum(dirid);
      file=dread(dirid,i);
      filename=upcase(scan(file, 1, '.'));
      output;
   end;
   rc=dclose(dirid);
run;
```

**KEEP THE FILES IN ORDER**

The printing order is the file order given by the **`file`** parameter. But in the case where all the files (e.g., `file=*.rtf`) for a given directory are to be printed, the names of the files are in alphabetic order when collected from the directory. To print the files in the desirable order, the **`forder`** parameter must be defined to give a text file or spreadsheet which contains the file names with the specified order. At Amylin, we use the file containing titles and footnotes for tables and listings to meet this purpose. The macro will read in this external file and merge with the data set, `_file`, obtained from the previous step.

Once the `_file` data set is sorted in the desired order, the following code will be employed to create two macro variables with the `CALL SYMPUT` routine for the files and the total number of the files to be printed.

```
data _null_;
   set _file end=eof;
   call symput('file'||compress(_n_), trim(file));
   if eof then call symput('total',(compress(_n_)));
run;
```

**ITERATE THE PRINTING PROCESS**

Because SAS will not directly print Word files, the macro must rely on Word to do the job. To communicate with Word from SAS, the following filename statement is used to establish the linkage between SAS and Word via DDE:

```
filename word DDE 'Winword|System' lrecl=1000;
```

In order for a client/server communication link to be established, both SAS and Word must be running. Therefore, for the first iteration, it is necessary to programmatically launch Word from a SAS session. There are a few techniques available to launch Word from SAS. To open Word, the **`%print`** uses the following statement:

```
%let rc=%sysfunc(system(start winword));
```

Prior to executing this statement, there are two system options needed: `NOXWAIT` and `NOXSYNC`. The `NOXWAIT` option specifies that the DOS Command Prompt window disappears without your having to type EXIT when the process is finished, while the `NOXSYNC` specifies that the process should execute asynchronously. That is, control is returned immediately to the SAS System and the command continues executing without interfering with your SAS session.

Now, it's time to communicate with Word via DDE by sending WordBasic command using a `DATA _NULL_` step:

```
%do i=1 %to &total;
   data _null_;
      file word; ❶
      put " [FilePrint"
          ".FileName = ""&dir\&&file&i"", "
          ".Background=1, "
          ".Range=&range, "
          ".Pages=""&page"", "
          ".NumCopies=&copy]" ; ❷

   run;
%end;
```

❶ The `FILE` statement specifies the fileref `word` defined in the previous doublet-style DDE `filename` statement for `PUT` statements in the current `DATA` step. In other words, the WordBasic commands sent by the `PUT` statements will control Word. ❷ This `PUT` statement will send the **`FilePrint`** command to Word to instruct it by the arguments. The argument values are controlled by the macro variables, except the `&&file&i` and `&range` macro variables, specified by the macro parameters in the call to the **`%print`** macro. The `&&file&i` macro variable is created by the `CALL SYMPUT` routine previously described, while the `&range` is derived based on the definitions of the `&page`. Note that the `&dir\&&file&i` is quoted with double-quotes, which is necessary because a single set of double-quotes must be kept to quote the resolved macro value after it is compiled.

Though not presented here, in order to make the macro more robust, the macro variables defined by the parameters in the **`%print`** macro call are processed and error handling is also implemented. For instance, there are many ways to define the **`page`** parameter. The macro variable defined by this parameter must be converted to the format that Word can understand. For the error handling, for example, the process of a macro call will be aborted if the **`dir`** parameter is not specified or if there are no files in the given directory.

**SAMPLE MACRO CALLS**

Assuming the **`%print`** macro is available in the SAS autocall library or compiled by the `%include` statement, the following calls to the macro are made to demonstrate its usage:

To print two copies of all the pages for all the Word `.doc` files in the `c:\pharmasug` directory with the default printer:

```
%print(dir=c:\pharmasug, page=all, copy=2)
```

To print the first and last pages of all the RTF files prefixing with '`t_`' in the `c:\pharmasug` directory with the order defined by the `c:\pharmasug\title.txt` file:

```
%print(dir   =c:\pharmasug,
       file  =t_*.rtf,
       page  =first last,
       forder=c:\pharmasug\title.txt)
```

To print the first 5 pages of the files specified by the **`file`** parameter in the `c:\pharmasug` directory:

```
%print(dir =c:\pharmasug,
       file=l_demog.rtf l_ae.rtf t_ae.doc t_lab.doc,
       page=1-5)
```

## CONCLUSION

With this macro utility tool, the entire process of printing a large number of Word files in Windows is automated. Customization of printing by selecting the pages to be printed makes the process more cost effective and timesaving. You can relax or do other important things while letting SAS do the job for you.

## REFERENCE


Cowmeadow, Mary (2004), The Big Button Approach for Quickly Running and Printing Large Numbers of Tables. *Proceedings of the PharmaSUG 2004 Conference*, paper AD17.

Hamilton P. (2003), ODS to RTF: Tips and Tricks. *Proceedings of the Twenty-eighth Annual SAS User Group International Conference*, paper 24-28.

Microsoft Press (1995), Word 95 WordBasic Reference. *http://www.microsoft.com/downloads/details.aspx?familyid=1a24b2a7-31ae-4b7c-a377-45a8e2c70ab2&displaylang=en.*

Peszek, I., Song, C., and Kuznetsova, O. (1999), Producing Tabular Reports in SAS System in the Form of MS Word Tables. *Proceedings of the PharmaSUG 1999 Conference*, 298-293.

Qi, E. and Zhang, L. (2003), %RTFTable – A Powerful SAS® Tool to Produce Rich Text Format Tables. *Proceedings of the PharmaSUG 2003 Conference*, paper TT038.

Smoak, C. G. (2004), Creating Word Tables using PROC REPORT and ODS RTF. Proceedings of the PharmaSUG 2004 Conference, paper TT02.

Zhou, Jay (2001), From SAS ASCII File to Word Document – A SAS Macro Approach. *Proceedings of the PharmaSUG 2001 Conference*, pp. 49-52.

Zhou, Jay (2002a), Output Generating System – A Tool for Creating Tables and Listings in Word. *Proceedings of the 27th Annual SAS Users Group International Conference*, paper 127-27.

Zhou, Jay (2002b), %Qexcel Tackles the Challenges of Converting Data to SAS from Excel. *Proceedings of the PharmaSUG 2002 Conference*, pp. 65-70.


## ACKNOWLEDGEMENTS


The author would like to thank Larry Shen, Maurice Gloster, and Mi Kim for reviewing this manuscript.




## CONTACT INFORMATION

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


Jay Zhou
Amylin Pharmaceuticals, Inc.
9360 Towne Centre Drive
San Diego, CA 92121
Email: zhou_jay@yahoo.com