

# Reading Title and Footnote from RTF Output into SAS® utilizing Microsoft® Excel

Ajay Gupta, PPD Inc, Morrisville, NC

## ABSTRACT

Capturing titles and footnotes from Microsoft Word readable RTF documents into SAS datasets for the quality control/validation purposes has always been a tedious task for SAS programmers. Also, manually typing titles and footnotes from RTF documents into various places in SAS programs is clearly time consuming and prone to typing errors. In addition, SAS does not provide a built-in procedure to import titles and footnotes directly from a Microsoft Word (RTF formatted) document into SAS, but SAS does have built-in procedures for reading data from Microsoft Excel. In this paper a SAS macro %Read\_Title\_Footnote is introduced to read titles and footnotes from the Microsoft word readable documents using Excel/Word Basic commands via DDE.

## INTRODUCTION

In the pharmaceutical/CRO industry, tables and listings are often created in word (.rtf) documents using the Output Delivery System (ODS) destination. For quality control purposes, titles and footnotes from word (.rtf) tables are visually compared with the output generated on the same specifications from a second programmer or statistician. This method is time consuming and prone to human error. In order to replace human visual check methods programmatically, the SAS developer needs to convert the titles and footnotes from word (.rtf) documents into SAS datasets. However, there is no existing SAS procedure which will directly import titles and footnotes from RTF documents into SAS. Also, one of the time consuming and tedious jobs is to enter the titles and footnotes line-by-line from a table shell into SAS programs. Very often the titles and footnotes can be changed even after the table outputs are generated. The ongoing changes require SAS programmers to update their table programs many times and in varying places. This may cause errors in the final reports.

In this paper a SAS macro %Read\_Title\_Footnote is introduced to read titles and footnotes from the Microsoft word readable documents using Excel/Word Basic commands via DDE. In order to avoid any typing error caused while entering the text into SAS programs, a variation of this solution can be use to read titles and footnotes from word documents. Also, any further changes in the titles and footnotes can be easily incorporated into a SAS dataset for the final study report. This convenient and reliable solution will help SAS programmers/statisticians to have better control over the quality of reports and save significant time. The entire process involves SAS, Word, and Excel which is automated by a SAS macro called %Read\_Title\_Footnote. It will import the entire content from the title and footnote section of RTF tables, into SAS data.

## TECHNIQUE & MECHANISM

The general process to convert a RTF document (header/footer) into a SAS dataset is as follows:

1. Open the target file in Word and copy the header/footer section.
2. Open Excel and paste the content in Excel and save the Excel file.
3. Read the content into SAS with LIBNAME or PROC IMPORT.
4. Delete the Excel file from the production location.

To automate these steps, the DDE solution is leveraged to build the communication bridge between SAS and Word 2003 or Excel 2003. WordBasic commands (Microsoft Corporation, 1999) can be sent from SAS via DDE to enable SAS to take control of Word and Excel

## OPEN THE TARGET FILE IN WORD AND COPY THE CONTENT FROM THE HEADER/FOOTER SECTION

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 several techniques available to launch Word from SAS. The simplest one is the following statement:

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

The above command is dependent on the finishing of previous commands. Note, the SLEEP function can be used frequently in the SAS command/datastep that is dependent upon previous jobs finishing. This will help avoid errors from occurring due to delays in the execution of previous SAS commands/datasteps.

The syntax for the SLEEP function is given below:

```
DATA _null_;
  x=sleep(5);
RUN;
```

The above step will pause the SAS session for five seconds.

To communicate with Word from SAS, the following FILENAME statement is used to establish the linkage between SAS and Word via the DDE triplet:

```
FILENAME word DDE 'Winword|System';
```

Another technique (Roper 2000) to launch Word from SAS is given below:

```
%MACRO OpenWord;

  FILENAME word DDE 'Winword|System';

  DATA _NULL_;
    Length fid rc start stop time 8;
    fid = FOPEN('word','s');
    If (fid le 0) then do;
      rc = system('start winword');
      start = datetime();
      stop = start + 10;
      Do while (fid le 0);
        fid = FOPEN( 'word', 's' );
        time = DATETIME();
        If (time ge stop) then fid = 1;
      End;
    End;
    Rc=FCLOSE(fid);
  RUN;

%MEND OpenWord;
%OpenWord;
```

Macro %Openword will start an instance of Word if none is running yet, and will keep trying for at most ten seconds to establish a DDE conversation via the system-triplet word. The filename statement will remain same as given above.

Below is the screen shot of rtf file:

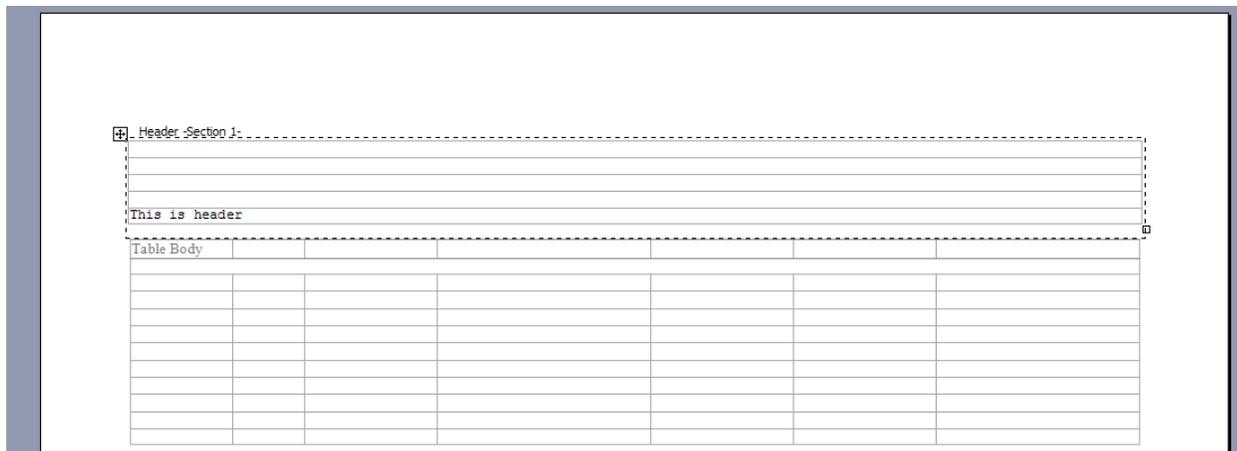


Fig 1: Example of rtf file.

The next step is to open the target file "&in" by sending the FileOpen WordBasic command to Word with a data \_null\_ step. Once the file is open in Word and the cursor moves to proper header section/page number, selects and copies the entire header by sending the EditSelectAll and EditCopy commands as shown in the following data step:

```

DATA _null_;
  FILE word;
  put '[FileOpen.Name = "' &in" ']' ;
  put "[EditGoTo.Destination= "' &page" ']]";
  put "[Viewheader]";
  put "[EditSelectAll]";
  put "[EditCopy]";
  put '[FileClose]';
  put '[FileExit]';
RUN;

```

It is necessary to programmatically close the document with the `FileClose` command. Further, the `FileExit` command will exit Microsoft Word.

Since, the cursor can only move either the header or footer. It is not possible to read both header and footer in the same run. So, the whole macro needs to be run twice separately to get the header/footer of an rtf document by pointing the cursor to either header/footer section respectively.

In order to copy the footer section, the word basic command "Viewheader" can be replaced by "Viewfooter" in the above dataset. This will allow the cursor to point towards the footer section.

#### OPEN EXCEL AND PASTE THE CONTENT IN EXCEL AND SAVE EXCEL FILE

The same approach as invoking Word is used to start a new Excel session:

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

To communicate with Excel from SAS, the following `FILENAME` statement is used to establish the linkage between SAS and Excel via the DDE triplet:

```
FILENAME excel DDE 'Excel|System';
```

Similarly, the `%OpenWord` macro given above can be easily modified to launch Excel from SAS.

The following command will paste and save the data in Excel file, and then close and exit the Excel session. The `Error(false)` macro function clears all error-checking and keeps Excel from displaying pop-up window.

```

DATA _null_;
  FILE excel;
  put '[Error(false)]';
  put '[Paste]';
  put '[SAVE.AS("' &location\content.xls" ')]';
  put '[Close]';
  put '[Quit]';
RUN;

```

After execution of the above dataset, the data will be stored in `Content.xls`. Please see the screenshot below.

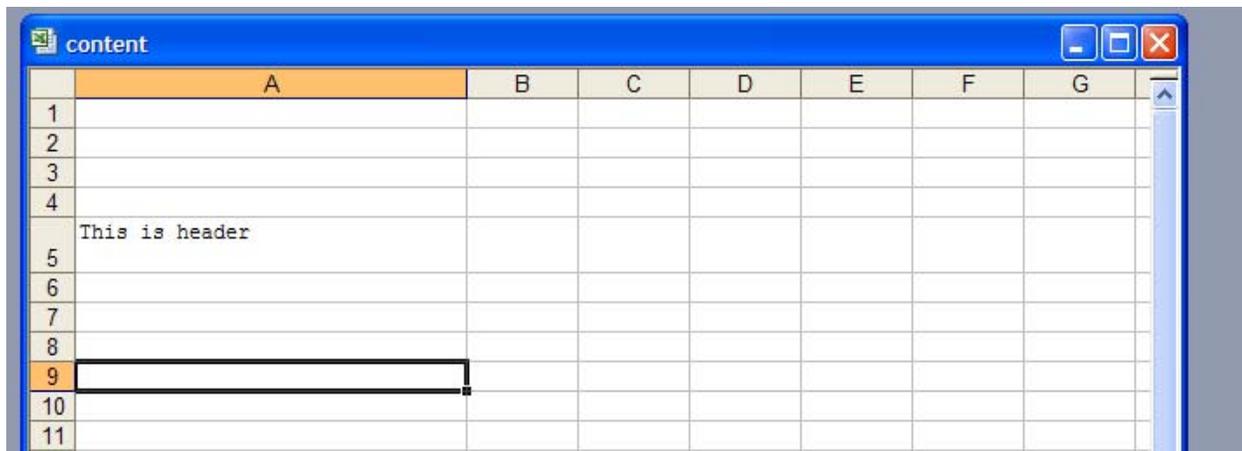


Fig 2: Screenshot of Excel File.

### READ THE CONTENT INTO SAS USING LIBNAME OR PROC IMPORT PROCEDURE

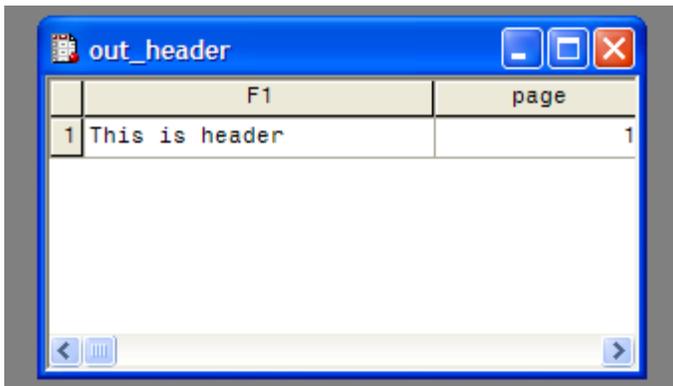
After the text or data is saved into an Excel file, it is time to read it into SAS by either using the LIBNAME statement or PROC IMPORT. If the SAS/ACCESS license is available, the following libname can be used:

```
LIBNAME xls excel "&location\Content.xls" header=no mixed=yes;
LIBNAME dat "&location";

DATA dat.&out.;
    set xls."Sheet1$"n;
    if missing(f1) then delete;
RUN;

LIBNAME xls clear;
```

Below is the screenshot of output dataset:



	F1	page
1	This is header	1

Fig 3: Output SAS dataset.

In the above dataset, "&location" is the production location and "&out" is the output dataset created and saved in library "dat". This dataset can be further manipulated in SAS and compared with the dataset created by other programmers/statisticians using PROC COMPARE.

### DELETE THE EXCEL FILE FROM THE PRODUCTION LOCATION

To avoid any overwriting of the data, Excel file can be deleted as follows:

```
FILENAME db "&location\Content.xls";
%let rc=%sysfunc(fdelete(db));
FILENAME db clear;
```

"&location" is the production location.

### %Read\_Title\_Footer

To facilitate and automate the above discussed steps from opening the Word file to reading the Excel file into SAS, a SAS macro called %Read\_Title\_Footer was developed using SAS v9.1.3 or above (see Appendix for details) and Microsoft Excel 2003. The user can easily extend the macro to fit other SAS versions.

There are only five keyword parameters:

**In:** Define the path and the name of the input Word file, e.g., C:\Demo\test.rtf.

**Out:** Define the SAS dataset name, for e.g., raw.

**Location:** Production Location, for e.g., C:\Demo.

**Section:** Get the header or footer section, for e.g., Viewheader or Viewfooter.

**Page:** Page number default is 1.

Below is the simple macro call to %Read\_Title\_Footer.

```
%Read_Title_Footer(in= C:\Demo\test.rtf, out=raw, location=C:\Demo, section=,page=);
```

## CONCLUSION

Macro %Read\_Title\_Footnote can capture the title and footnote from word readable RTF documents into a SAS dataset without any manual pre-process. The only drawback of this method is that the process is partially based on the DDE technology which may not be supported by Microsoft Corporation in future MS Office releases.

## REFERENCES

Gupta Ajay, Matthew Lesko. 2010. Importing Data from RTF Output into SAS® utilizing Microsoft® Access/Word in an intriguing, efficient way. Proceedings of the PharmaSUG 2010 Conference, paper CC06.

Huang, Daniel and Lois Lynn. 2008. Read Titles and Footnotes from a Table Shell into a SAS Dataset. Proceedings of the PharmaSUG 2008 Conference, paper PO07.

Zhou, Jay. 2009. Importing Data from Microsoft Word into SAS. Proceedings of the PharmaSUG 2009 Conference, Paper CC18.

Microsoft Corporation. 2000. Word 95 WordBasic Help File.

<http://www.microsoft.com/downloads/details.aspx?familyid=1a24b2a7-31ae-4b7c-a377-45a8e2c70ab2&displaylang=en>

Roper, C. A. 2000. Intelligently Launching Microsoft Excel from SAS, using SCL functions ported to Base SAS. Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, paper 97.

Viergever William, Koen Vyverman. 2003. Fancy MS Word Reports Made Easy: Harnessing the Power of Dynamic Data Exchange. Proceedings of the Twenty-Eight Annual SAS Users Group International Conference, paper 16-28.

## ACKNOWLEDGMENTS

Thanks to Karen Easterling and Anita George for there reviews and comments. Special Thanks to my wife Neha for motivating me to write this paper.

## CONTACT INFORMATION

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

Ajay Gupta, M.S.  
PPD Inc.  
3900 Paramount Parkway  
Morrisville, NC-27560  
Phone: (919)-456-6461  
Email: Ajay.Gupta@ppdi.com, Ajaykailasgupta@aol.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.

## APPENDIX

```
%Macro Read_Title_Footnote(in=, out=raw, location=, section=,page=1);

options noxwait noxsync ;
%let rc=%sysfunc(system(start winword));

data null ;
  x=sleep(5);
run;

filename word DDE 'Winword|System';

data null ;
  file word;
  put '[FileOpen.Name = "' &in" ']' ;
  put '[EditGoto.Destination="' &page" ']' ;
  put "[View&section]";
  put '[EditSelectAll]';
  put '[EditCopy]';
  put '[FileClose]';
  put '[FileExit]';
run;

data null ;
  x=sleep(5);
run;
%let rc=%sysfunc(system(start excel));

data null ;
  x=sleep(10);
run;

filename excel DDE 'Excel|System';

data null ;
  file excel;
  put '[Error(false)]';
  put '[Paste]';
  put '[SAVE.AS("' &location\content.xls" ')]';
  put '[Close]';
  put '[Quit]';
run;

data null ;
  x=sleep(7);
run;

libname xls excel "&location\content.xls" header=no mixed=yes;

data &out. &section;
  set xls."sheet1$"n;
  page=&page;
  if missing(f1) then delete;
run;

libname _all_ clear;

FILENAME db "&location\Content.xls";
%let rc=%sysfunc(fdelete(db));
FILENAME db clear;

%Mend Read_Title_Footnote;
```