SAS Macros to Help Create Report Quality Tables in Word Documents

Brendan Bartley, Center for Biostatistics in AIDS Research (CBAR), Boston, MA
Heather Ribaudo, PhD, Center for Biostatistics in AIDS Research (CBAR), Boston, MA

Abstract
This paper describes macros to enhance RTF output for reports. The first macro allows the user to move tables within a document and have the table numbering change corresponding to their location in the document. It also sets up bookmarks for reference in a report. Another macro eliminates all but the first section breaks in a RTF document. The last macro gives a table its own style name so that its formatting does not change when text formatting in the document changes.

SAS Product: Base SAS, Skill Level: Any, Keywords: Macros, ODS, RTF, no_section, caption, bookmark, Wordstyle, Word field instructions, Word, \sect.

Introduction
The Output Delivery System (ODS) has enabled SAS users to output results to several destinations, with one of the most common being Rich Text Format (RTF) which is a destination compatible with Microsoft Word. As well as providing for high quality tabulation and plotting capacity, SAS has incorporated more advanced features for cross-referencing and indexing RTF output. Unfortunately, when one is writing detailed analysis reports that include large amounts of descriptive text either interspersed between tables or preceding groups of tables, the utility of these features is more limited. For example, within the Center for Biostatistics in AIDS Research (CBAR), analysis reports involve very detailed textual description of studies and data followed by a series of tables and figures – RTF files linked in to the Word document (such that they are easily updated should changes to the data be made and analysis programs re-run). Within these documents, the utility of the SAS numbering and cross-referencing features was limited – documents already utilized Word table of contents and headings for textual descriptions which conflicted with the cross-referencing capacities built into ODS. To improve the efficiency of our work, we sought ways to refine and automate the numbering and cross-referencing of tables and figures, as well as better controlling the layout of RTF output. After discussions with Wayne Hester from The SAS Institute and much experimentation, we developed a series of tools that have dramatically increased the efficiency of our work. These tools are described in this paper. They include macros to:

1. Incorporate Word field instructions within simple SAS macros that facilitate and automate labeling and numbering of RTF output, while also creating bookmark names for reference;
2. Remove section breaks that may otherwise wreak havoc on document formatting;
3. Apply distinct paragraph styles to the RTF output from SAS, so that its formatting differs from that of the report.
Caption Macro
Previously, all tables were labeled and numbered manually after insertion of the RTF output into WORD (via INSERT – REFERENCE – CAPTION). Although individually numbering the tables facilitated cross-referencing and lists of tables it was time-consuming and not without issue. With a better understanding of RTF syntax and Word field instructions our table numbering and labeling is now all done within SAS code. It’s not pretty, but it works!

\title{ "Table \field{\*ldinst{SEQ Table \* ARABIC }}: Text for your title" ;

Key Features:

- First, notice that this is a simple TITLE statement; the resolved text string will always be a title associated with any subsequent procedure output.
- Second, below is RTF syntax that introduces Word to field instructions (or field codes).
  \\field{\*ldinst{ ... }}
- Last, the code contains instructions for the field code.

SEQ Table \* ARABIC

These instructions retrieve a number (ARABIC) corresponding to a “Table” sequence (SEQ Table). The SEQ is a type of field and “Table” is a sequence identifier. This number does not appear initially in Word, but with these field codes in place, Word can produce a number coordinating with the sequence of tables in a document. This is how we can get table numbering to change according to its position in the RTF document. Now, we can create a listing of tables utilizing the Word cross-referencing facilities specifically designed to retrieve this information from a sequence of numbers.

Although the use of these sequence numbers also allows cross-referencing within the text of the Word document (i.e. INSERT – REFERENCE – CROSS-REFERENCE – Table), since our work practices were to link RTF files of tables within Word reports often needing to update them as data changes, such cross-referencing does not work (the invisible field code Word places at the sequence number being cross-referenced is lost when the linked RTF files are updated and the cross-reference fails). This can be worked around by placing hard coded bookmark tags around the table label and number.

\title{ {{\*kmkstart MYBOOKMARKNAME} 
{Table \field{\*ldinst{SEQ Table \* ARABIC }}}} 
{\*kmkend MYBOOKMARKNAME}: Text for your title} ;

Notice how the bookmarked text is simply the text string describing the output label (in this case Table) and the sequence number. When the bookmark is subsequently cross-referenced within the Word document, the referenced text will then simply look like Table 2 or Table 3. Something to note is that both references to “Table” in the code
above could be changed to “Figure”, or “Bumblejet”, or any word and then the referenced text would look like Figure 1 or Bumblejet 1. In our macro, changing the value of the parameter TYPE to “Figure” or “Bumblejet” will accomplish this. The default value is “Table”.

These basic commands can be further refined to accommodate SEQ field switch options on the sequence number field codes (we will discuss briefly, but see Word help system for more details on these options, also check http://msdn2.microsoft.com/en-us/library/aa163918(office.10).aspx for more information).

For Example:

1. First, the default SEQ field switch option is /n and is not needed in the code because it is the default; it inserts the next sequence number in your TITLE statement.
2. The /c SEQ field switch option repeats the closest preceding sequence number. This allows us to create subtext numbers for tables and titles. Here’s some sample code; notice the changes in bookmark names so they can be cross-referenced separately:

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC }}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}}
   
   title2 "{{\*\bkmkstart MYBOOKMARKNAMEa}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \c}}a
   {{\*\bkmkend MYBOOKMARKNAMEa}: Subtext for the subtitle}}
   
   title1 "{{\*\bkmkstart MYBOOKMARKNAMEb}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \c}}}b
   {{\*\bkmkend MYBOOKMARKNAMEb}: More subtext for a subtitle}}

If the tables created by the above code appear as the third sequenced set of tables in the report, this series of code will resolve to create:

Table 3: Text for your title
Table 3a: Subtext for the subtitle
<Procedure output>
Table 3b: More subtext for a subtitle
<Procedure output>

3. There is also a SEQ field switch, /r, which resets the sequence number to a specified number. In the example, we are resetting it back to 1.

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \r 1}}}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}"

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \r 1}}}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}"

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \r 1}}}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}"

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \r 1}}}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}"

   title1 "{{\*\bkmkstart MYBOOKMARKNAME}
   {Table \field{{\*\fldinst{SEQ Table \* ARABIC \r 1}}}}
   {{\*\bkmkend MYBOOKMARKNAME}: Text for your title}"
4. Lastly, there is a way to create table numbers that match up with heading numbers. Using the \s SEQ field switch option combined with the STYLEREF field code will create double level table numbering that cross-references a current heading, like heading 2. Let’s say the following statement is for a table with the heading number 2.1.

```
title1 "{{\*\bkmkstart MYBOOKMARKNAME}
  {Table \{\*\fldinst{STYLEREF 2 \s}\}}.
  \field{\*\fldinst{SEQ Table \"* ARABIC \s 2\}}}
{\"\bkmkend MYBOOKMARKNAME}: Text for your title}"
```

The number that needs to be inserted, and is highlighted in the code, is 2 because the heading number of 2.1 is a heading 2 number. Also notice the period after the field code for STYLEREF so that it maintains the same pattern as heading 2. So, the first title in this section would look like:

**Table 2.1.1: Text for your title**

<Procedure output>

Rather than it being necessary for users to remember this somewhat awkward code, a single unifying macro pulls everything together with user defined parameters to control the number of the title statement, the caption text, the caption justification, the type of sequence number, the bookmark name, a field specific switch parameter, as well as a parameter to set the “number” (likely a letter when used with the c switch value) to be used in conjunction with the field switch parameter.

```
%macro caption(tnum=1,captiontext=,justify=left,type=Table,bkmrk=,swtch=n,swnum=) ;
  %if %lowcase(&swtch)=n %then
    title&tnum j=&justify "{{\*\bkmkstart &bkmrk}
      {&type \field{\*\fldinst{SEQ &type \"* ARABIC \n }}}}"
    {\"\bkmkend &bkmrk}: &captiontext" ;;
  %if %lowcase(&swtch)=c %then
    title&tnum j=&justify "{{\*\bkmkstart &bkmrk}
      {&type \field{\*\fldinst{SEQ &type \"* ARABIC \c }}}&swnum
      {\"\bkmkend &bkmrk}: &captiontext}"
    ;
  %if %lowcase(&swtch)=r %then
    title&tnum j=&justify "{{\*\bkmkstart &bkmrk}
      {&type \field{\*\fldinst{SEQ &type \"* ARABIC \r &swnum }}}}"
    {\"\bkmkend &bkmrk}: &captiontext}"
    ;
  %if %lowcase(&swtch)=s %then
    title&tnum j=&justify "{{\*\bkmkstart &bkmrk}
      {&type \field{\*\fldinst{SEQ &type \"* ARABIC \s &swnum}}}
      {\"\bkmkend &bkmrk}: &captiontext}"
    ;
  %mend caption;
```

Here are some examples of how you could use the %caption macro. The first call is the simplest form you could use (because of defaults in the parameters), but would not allow the user to capitalize on its bookmarking capabilities. The next set of calls show some of
the other possibilities using SWTCH=c. Lastly, we look at how to use SWTCH=s.

Examples:

``` PROC FREQ CODE %caption(CAPTIONTEXT= Reasons study participants are no longer followed); %caption(TNUM=2,CAPTIONTEXT= Mothers,JUSTIFY= center,BKMRK= momfreq,SWTCH=c,SWNUM=a); < PROC FREQ CODE> %caption(CAPTIONTEXT= Children,JUSTIFY= right,BKMRK= chldfreq,SWTCH=c,SWNUM=b); <PROC FREQ CODE>
```

This is how it will appear in Word:

Table: Reasons study participants are no longer followed

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table a</td>
<td>Mothers</td>
</tr>
<tr>
<td>Table b</td>
<td>Children</td>
</tr>
</tbody>
</table>

Notice that the table numbers are missing and that the titles have been justified to the left (default), center, and right respectively just to show what it looks like. Also, in the code, we have used TNUM=2 to create a TITLE2 statement. To retrieve these table numbers after the document is opened in Word, you must highlight all the titles, just the section of the title where a number should be, or the whole document (CTRL+A). Once an area is selected, hit F9 or right click on the highlighted area and then select “!Update Field”. The above can be skipped if the tables are inserted into a RTF document. Word will automatically update the numbers. When the numbers are updated, the code above will look like this:

Table 1: Reasons study participants are no longer followed

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1a</td>
<td>Mothers</td>
</tr>
<tr>
<td>Table 1b</td>
<td>Children</td>
</tr>
</tbody>
</table>

NOTE: When you want to use the caption macro with PROC GPLOT or some other graphic procedure, you must use the NOGTITLE and BODYTITLE option in your ODS RTF statement. Otherwise, it will place the title in the figure and the Word field code will not resolve and your title will look very ugly.

Now, let’s say you need your table numbers to coordinate with headings in your report. For this, we need to use SWTCH=s. For the example, we will say that your heading number is 2.1. We will alter the previous code to show where you can run into some problems and add a PROC FREQ so that the example makes sense.

``` PROC FREQ CODE %caption(CAPTIONTEXT= Reasons study participants are no longer followed,SWTCH=s,SWNUM=2); < PROC FREQ CODE > %caption(CAPTIONTEXT= Mothers,JUSTIFY= center,BKMRK= momfreq,SWTCH=s,SWNUM=2); < PROC FREQ CODE > %caption(CAPTIONTEXT= Children,JUSTIFY= right,BKMRK= chldfreq,SWTCH=s,SWNUM=1); < PROC FREQ CODE > ```
After updating the fields or inserting the tables in another document, the Word document will appear like this:

Table 2.1.1: Reasons study participants are no longer followed
<PROC FREQ TABLE>

Table 2.1.2: Mothers
<PROC FREQ TABLE>

Table 2.3: Children
<PROC FREQ TABLE>

As you can see, we got the right numbering for the first 2 tables, but the third table does not have the 2.1 beginning. The reason for this is we only specified SWNUM=1. Also, notice that it did not change its sequence number to 1 and just continued to 3, the next number in the sequence. That is the %caption macro in a nutshell. Let’s move on to the %no_section macro.

No_Section Macro
In SAS, page breaks and section breaks are created between procedures and section breaks are created between multiple tables created in a procedure. The page breaks can be controlled by using the STARTPAGE= option in an ODS statement. The section breaks are much harder to control and can cause issues with paper layouts and resetting header and footer details of a document. We tried using VBA (Visual Basic for Applications) macros suggested in “Skinning a Cat…” by Elizabeth Axelrod and David Shamlin, but could not quite get it to work properly. So, we talked with Wayne Hester at the SAS Institute about finding a solution; he sent us the %no_section macro. This macro eliminates all section breaks in a document. We eventually altered it to eliminate all but the first section break. This first section break does not show up as a section break in the Word document, so it does not create any of the previously mentioned problems. Also, it seems to contain information about the formatting of margins for pages that are in landscape or have a multiple column page layout. Here is the no_section macro:

```sas
%macro no_section(source=xxx);
    /*
     * Reading and writing to the same file at the same time can be trouble
     * depending on where you are.
     */
    data temp ;
    length line $400;
    infile &source length=lg lrecl=1000 end=eof;
    input @1 line $varying400. lg;
    data _null_
        set temp ;
        retain flag  0 ;
        file  &source ls=400;
        /*
         * Trap for section data line only.
         * Don’t insert a space line for the first occurrence,
         * only between tables.
         */
```
/*
   Keep the first section information which seems to
define the characteristics of the document.
*/
if (index(line,"\sect") > 0) then do;
   if ( flag ) then
      line = "\par\";
   else do;
      flag = 1;
   end;
end;
put line;
run;
%mend no_section;

This macro is used in a similar way as the wordstyle macro below, so there will be an example of this macro with the wordstyle macro below.

Wordstyle Macro
This last macro came about by the nuisance of tables taking on the “Normal” Word style when inserted into a Word document. Since documents at CBAR contain many pages of text, they are often formatted to have double spaced lines with larger spaces between paragraphs. These are obviously formatting characteristics that are not compatible with table layouts.
The no_section macro showed us how to alter the RTF file in a macro and “Skinning the Cat...” showed us what to look for in the RTF file to create a style for tables separate from that of the text. By looking at the RTF code in a text finder, we found a line like this:

} {\stylesheet {\widctlpar\adjustright\fs20\egrid\snext0 Normal;} {\*cs10\additive Default Paragraph Font;}

The key word was ‘stylesheet’, which helped clue us in to where the RTF code needed to be altered. The default style for any Word document is Normal, so all the Wordstyle macro does is go into the file and change it to your own name. We made the default style name RTFOutput, but you may call it whatever you like. Here’s a look:

%macro wordstyle(source=,wordstyle=RTFOutput);
data temp;
   length line $400;
   infile &source length=lg lrecl=1000 end=eof;
   input @1 line $varying400. lg;

data _null_; set temp;
   file &source ls=400;
   if index(line,"Normal;\")>0 then line=tranwrd(line,"Normal;","&wordstyle;\")
      put line;
   run;
%mend wordstyle;
Here is a code example of how the wordstyle and no_section macros may be used in conjunction with the caption macro:

```
filename myoutput 'Z:\home\bbartley\mytables.rtf';
ods rtf file=myoutput bodytitle;
%caption(captiontext= Representation of my data.);
   proc report data=mydata;
   < code for table >
   run;
ods rtf close;

%no_section(source=myoutput);
%wordstyle(source=myoutput);
```

The No_section and Wordstyle macros must run after the RTF file is closed. They will not work if the file is open, so there must be an ODS RTF CLOSE statement before running the macros. The macros will eliminate the section breaks in your RTF document and give the RTF a Word style name, RTFOutput, which will be different from the Normal name that most documents will have and thus it can be formatted differently from the text of a report.

**Conclusion**

These macros are tools that aid in automatically making reports. They take care of a lot of things that would need to be done by hand or by learning how VBA macros work, which we did not find intuitive. You may find it helpful to learn more about RTF coding by trying other formatting in a Word document and then looking at the file in a text editor. We hope that you find these macros as useful as we have.

**References**

Wayne Hester, SAS Institute

**Acknowledgements**

Both authors would like to acknowledge the support of Wayne Hester from the SAS Institute, the programming staff at CBAR, and our family and friends.

**Contact Information**

Please feel free to send any comments or questions our way.

Brendan Bartley – bbartley@sdac.harvard.edu

Heather Ribaudo, PhD – ribaudo@sdac.harvard.edu
TRADEMARK CITATION
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.

Other brand and product names are registered trademarks or trademarks of their respective companies.
Helpful Word Review for this Paper

1. Directions for inserting a file into a Word document.
   a. Put cursor in Word document where you want to insert the file.
   b. Click Insert from the menu bar and scroll down to File and click again.
   c. Pop up box appears: Find the file you want to insert and click it ONCE.
   d. Move the cursor to the bottom right of the pop up box and click the Insert 
      and then click on Insert as Link.

2. Directions for selecting the whole Word document.
   a. Click Edit from the menu bar and scroll down to Select All.
   b. OR hit Ctrl+A

3. Directions for updating field codes once selected.
   a. Right click on the highlighted area, which will be the whole document if
      you followed #2 above, and then click !Update Field from the pop up box.
   b. Once the area you want is highlighted, hit F9.

4. Directions for inserting a bookmark into the text.
   a. Put cursor in Word document where you want to insert a bookmark.
   b. Click Insert from the menu bar and scroll down to Reference. Then slide
      the cursor to the right and click on Cross-reference.
   c. A pop up box appears and make Reference type = Bookmark.
   d. Click on a bookmark name you made in the %caption macro in the “For
      which bookmark” box and then click the Insert button. In my example on
      page 5, you would look for MOMFREQ or CHLDFREQ in the “For
      which bookmark” box.
   e. You will see the table number appear in your document like this: Table 1,
      or Figure 1.1, or Table 2a, or whatever number it is.
   f. If the table numbers change, these bookmark references will also change
      with them. So, if Table 1 became Table 3 and there was a bookmark for
      this table, then after updating the field the bookmark would also change to
      Table 3.