

097-2007

## Creating Table of Contents in RTF Documents

Electra Small, MDRC, New York, NY

### ABSTRACT

With the ODS (Output Delivery System) RTF destination, SAS® users build tables that are opened directly in MS Word and other word-processing packages. A natural next step is to automate the creation of tables of contents and indexes for documents. The table of contents (TOC) and the index provide readers with road maps to find the tables of greatest interest to them. This paper reviews the new, version 9 methods for creating RTF (Rich Text Format) destination tables of contents. In addition, this paper offers two less-documented methods for creating customized tables of contents. One method, which is experimental in version 9, utilizes MS Word styles; the other method uses ODS to pass RTF commands directly to the word processor. These two methods require slightly more advanced programming skills but allow for greater TOC customization.

### INTRODUCTION

This is the second in a series of papers that chronicles a project that utilized SAS to create a book in a completely automated fashion. "The survey results book" had a title page, a table of contents, text and tables, and at the end a document index. In phase 1 of the project, Proc Tabulate was used to produce nicely formatted MS Word tables that documented the questions and responses from a telephone survey. This phase of the project was detailed in a paper presented at NESUG 2005. Phase 2 of the project, the focus of this paper, utilized SAS to add a table of contents to the results book. Phases 3 and 4 involved the use of SAS to construct a document index and a title page. These will be topics of future papers.

This paper illustrates three methods that can be used to create tables of contents in RTF destination documents. The first method utilizes the ODS RTF CONTENTS option which is new in version 9 and very easy to use. When this option is specified SAS builds a very detailed TOC in the RTF document. The programmer, however, has only a limited ability to control the wording of TOC entries and the number of TOC entries. The second method (experimental in version 9) allows the programmer to build a more general or condense TOC, in which the programmer controls the wording of the entries. With the second method SAS uses the power of MS Words styles; therefore, this method is only appropriate for those also using MS Word. The third method allows programmers to customize the TOC, general or detailed TOCs can be built, and the programmer controls the wording of all entries. This method, which uses ODS to embed RTF TOC commands in the document, requires more coding by the programmer, so it is not as easy as the other methods. This third method can also be used in SAS version 8.

### NEW IN VERSION 9: TABLE OF CONTENTS METHOD 1

Let's start with the new method that appears in Version 9; this method uses the ODS RTF CONTENTS=YES option and the ODS PROCLABEL statement. Consider the following example: there is a survey with two sections; the first section gathers respondents' personal information, and the second section inquires about the children in the respondents' homes. Each section of the survey has five questions. The aim is to create a table of contents, similar to the one below, that displays the initial page number referencing the tables in each section.

```
Table of Contents
Section 1: Personal Information... 2
Section 2: Children in the home... 4
```

Using Method 1, we would submit the following commands:

```
options nocenter nodate;
run;
ods rtf file='MyTOCdoc.RTF' startpage=no keepn CONTENTS=YES;
ods noproctitle;

title ; RUN;

ODS PROCLABEL='Section 1: Personal Information';
proc freq data=a; table q1-q5*groups /norow nopercnt nofreq; run;

ODS PROCLABEL='Section 2: Children in the home';
proc freq data=a; table q6-q10*groups /norow nopercnt nofreq; run;
ods rtf close;
```

The CONTENTS=YES option in the first ODS statement tells SAS to create a TOC preceding any output at the start of the RTF document. The ODS PROCLABEL statements are optional. The PROCLABEL statement creates customized label in the TOC that denotes the initial page of output from each procedure. In this example, we have two procedures (PROC FREQs); thus, we can supply two custom labels. Each PROCLABEL statement is, in effect, for only one procedure.

Notice that, in the first ODS statement, the STARTPAGE=NO and the KEEPNO options were used. The STARTPAGE=NO option requests that multiple tables be placed on each page. (The default is one table per page.) The KEEPNO option applies to situations in which half the table would be placed on one page and the other half would carry over to the next page. This option forces a page break in order to position the entire table on one page.

The table of contents generated will document the start page of each procedure; furthermore, if a procedure produces multiple tables, the TOC will document the start page of each table. When the RTF document is opened in MS Word, the title “**Table of Contents**” will be visible; however, the actual TOC table is not. Press **Ctrl+A** (to select all) and then **F9** (to generate the TOC) to view the actual TOC table.

As an aside, if the SAS-generated RTF document is “post-edit” (i.e., adding text or custom headers, page breaks, tables), these modifications may change the page locations of the tables. In this case, update the TOC by requesting a repagination directly in MS Word. To update a TOC after any document edits, press Ctrl+A (to select all) and then F9 (to update the TOC), selecting “entire table” when prompted. This step is also advisable if the page size or even printer selection is changed.

In the current example, we have two procedures that produce a total of 10 tables. The SAS-generated TOC in our RTF document will be:

*Table of Contents*

Section 1: Personal Information.....	1
Table q1 * Groups.....	1
Cross-Tabular Freq Table .....	1
Table q2 * Groups.....	1
Cross-Tabular Freq Table .....	1
Table q3 * Groups.....	2
Cross-Tabular Freq Table .....	2
Table q4 * Groups.....	2
Cross-Tabular Freq Table .....	2
Table q5 * Groups.....	2
Cross-Tabular Freq Table .....	2
Section 2: Children in the home.....	3
Table q6 * Groups.....	3
Cross-Tabular Freq Table .....	3
Table q7 * Groups.....	3
Cross-Tabular Freq Table .....	3
Table q8 * Groups.....	3
Cross-Tabular Freq Table .....	3
Table q9 * Groups.....	4
Cross-Tabular Freq Table .....	4
Table q10 * Groups.....	4
Cross-Tabular Freq Table .....	4

Note that when you generate the TOC (by pressing Ctrl+A and F9), you may not care for the styling (margins, font weight, size, type) of the TOC. MS Word offers a few TOC style templates, which can be accessed by **left-clicking in the middle of the TOC** (causing the TOC to become gray). Go to the menu, and select **Insert→Reference→Index and Tables**. Click on the **Table of Contents** tab; then select a **style** from the drop-down box (**initially showing From Template...**) at the bottom that appears to the right of Format. If the word “ERROR” appears in your TOC, this only means that the TOC needs to be updated (regenerated). As before, press **Ctrl+A and F9**; then select “**entire table**” to update the TOC.

One problem/inconvenience of the above TOC is the redundancy of the wording. (Imagine how many TOC entries are created as a result of producing over 100 tables.) Also, the repetition of the procedure names (e.g., Cross-Tabular Freq Table) is distracting, and the wording is too technical for some audiences. In this situation, you can utilize the features of PROC PRINT, REPORT, TABULATE, or FREQ by adding a CONTENTS= option to reword the procedure titles appearing in the TOC:

```
proc freq data=a;
table q6-q10*groups /norow nopercnt nofreq CONTENTS='Section 1 tables';
```

The statement above would change the existing procedure title, "Cross-Tabular Freq Table," to "Section 1 tables." To remove these titles altogether (demonstrated below); leave a blank inside the quotation marks:

```
proc freq data=a;
  table q6-q10*groups /norow nopercnt nofreq CONTENTS=' ' ;
```

Our final TOC would appear as follows:

*Table of Contents*

Section 1: Personal Information.....	1
Table q1 * Groups.....	1
Table q2 * Groups.....	1
Table q3 * Groups.....	1
Table q4 * Groups.....	2
Table q5 * Groups.....	2
Section 2: Children in the home.....	2
Table q6 * Groups.....	2
Table q7 * Groups.....	3
Table q8 * Groups.....	3
Table q9 * Groups.....	3
Table q10 * Groups.....	4

Another nice feature of this TOC is the hyperlinking of each TOC entry with its corresponding page. Clicking on any TOC entry will automatically redirect the cursor to that location in the document.

Note that the CONTENTS= procedural option works for crosstabs output produced by PROC FREQ (e.g., table gender\*race), but it does not work for one-way frequency tables produced by PROC FREQ (e.g., table gender race). Further, the CONTENTS= procedural option is only available on the FREQ, PRINT, REPORT, and TABULATE procedures. If the output includes results from one-way frequencies or one of the other procedures not listed above, there is also a PROC TEMPLATE solution that will allow the programmer to exercise the same level of control over the wording of the procedural title TOC entries.

PROC TEMPLATE is a very powerful and a very complex procedure. A full description of the use of PROC TEMPLATE to edit table templates is beyond the scope of this paper. But, briefly, each output-producing procedure has one or more table templates. These templates are the specifications that control how the output for each procedure will be laid out on the page (e.g., page headings, column headings, column order, column widths). To change the layout and look of the output pages, SAS allows programmers to edit the table templates (or the layout specifications) for its procedures. To edit a table template, the first thing the programmer needs to know is the name of the table template. One way to find out the name of the table template is to submit ODS TRACE ON /LISTING just before the procedure of interest. The table template that controls the look of the one-way frequency output is base.freq.onewayfreqs.

The table template attribute that controls the wording of the TOC entry is the contents\_label attribute. To specify custom wording for the TOC entries of each frequency procedures, set the contents\_label attribute to a macro variable and change the value of that macro variable just before each procedure. Further setting the macro variable to a blank (inside a macro quote function) will reduce the number of entries in the TOC. PROC TEMPLATE is used to declare a macro variable with the mvar statement (e.g., mvar cl;) and, as usual, the value of the macro variable can be changed with a %LET statement. The code below edits the one-way frequency table template in this manner:

```
ods path(prepend) sasuser.templat(update);

proc template;
  edit base.freq.onewayfreqs;
    mvar cl;
    contents_label=cl;
  end;
run;

ods rtf file='NESUG_TOctemplateedit.rtf' style=rtfcw startpage=no bodytitle keepn
contents=yes;
ods noproctitle;
title;
RUN;
```

```

%let cl=SOME LABEL OF MY CHOICE;
ODS PROCLABEL='Section 1: Personal Information';
proc freq data=a; table q1-q6;run;

%let cl=%str( );
ODS PROCLABEL='Section 2: Children in the home';
proc freq data=a; table q6-q10; run;
ods rtf close;

```

### *Table of Contents*

Section 1: Personal Information.....	1 <sup>□</sup>
Table q1.....	1
SOME LABEL OF MY CHOICE.....	1
Table q2.....	1
SOME LABEL OF MY CHOICE.....	1
Table q3.....	1
SOME LABEL OF MY CHOICE.....	1
Table q4.....	1
SOME LABEL OF MY CHOICE.....	1
Table q5.....	2
SOME LABEL OF MY CHOICE.....	2
Table q6.....	2
SOME LABEL OF MY CHOICE.....	2
Section 2: Children in the home.....	2
Table q6.....	2
Table q7.....	2
Table q8.....	2
Table q9.....	2
Table q10.....	3

The above TOC is very nice and easy to produce with SAS. There are times when a detailed TOC like this is desired. For this project, however, the level of detail is still a bit much, because with over 100 questions in the original survey the TOC would have over 100 TOC entries (one for each question) and multiple TOC entries for most of the pages of the document (in the TOC above there are over four entries for page 2 alone).

## CREATING INDEXES IN MS WORD

To understand the next 2 TOC methods, it is helpful to have some background knowledge about how TOCs are produced in a MS Word document.

One way to create a customized TOC is to highlight an area of text that is to comprise a TOC entry and mark it for inclusion by pressing **ALT+SHIFT+O** and then clicking **Mark.** You will notice that `{TC "Marked text" \f C \l "3"}` appears next to the highlighted area; these codes are TC field codes and RTF formatting commands. Repeat this step with each TOC entry text.

Another way to create a customized TOC is first intersperse section headers thought document. Next highlight a section header. From the main menu select **Format→Styles and Formatting...→Heading 1**. Repeat this step for every section header in the document. Notice as each section header is marked with the "Heading 1 style" the formatting attributes of the selected text will change, in most sites to Arial font, 16 pt, and bold.

After using either method for marking the TOC entries, position the cursor where you want your TOC to be located. From the menu, select **Insert→Reference→Index and Tables→Table of Contents→Options**. Notice that by default Heading Styles and outline levels are already checked. Also check the box marked **Table entry fields** (the RTF TC field codes). MS Word collects all the TOC entries that were marked and places these in the table of contents. This TOC table is also generated by fields or RTF commands that are hidden behind the TOC. To view them, press **ALT+F9** (this is a toggle). ALT+F9 reveals commands that appear as follows: `{TOC \o "1-3" \f \h \z \u}`. The command and parameters are explained later.

## EXPERIMENTAL IN VERISON 9: TABLE OF CONTENTS METHOD 2

The second TOC building method uses the ODS WORDSTYLE option that is experimental in version 9. This method allows programmers more control over the number of TOC entries and the wording of those entries. As demonstrated in the previous section, to build a TOC first the TOC entries must be marked and next the TOC table must be built. This method allows the programmer to mark the TOC entries from within the SAS program but the TOC tables must be built in MS Word. Further this method uses the MS Word Heading 1 style so this method works only with MS Word.

In the specifications for RTF documents style sheets are optional. However, if an RTF document includes the use of a style sheet file and custom style definitions it must follow RTF specifications for doing so. MS Word provides a standard style sheet file "Normal.dot." And within that style sheet file a number of custom styles are defined by default. Heading 1, Heading 2, and Heading 3 are declared and defined by default in normal.dot. Further as displayed in the previous section when MS Words builds a TOC it collects all entries marked with the Heading 1, 2 or 3 styles.

To take advantage of this MS Word feature from within SAS first declare a "nickname" for the Heading 1 style and next use that nickname to mark any heading text that is to become a part of the TOC. To declare a nickname for the Heading 1 style use the following syntax:

```
ODS RTF FILE="TOC2.RTF" startpage=no keepn NOTOC_DATA wordstyle='(\s1 Heading 1;);
```

Note that the NOTOC\_DATA option is also specified here. By default SAS version 9 marks all output tables with TC fields. The NOTOC\_DATA option prevents SAS from adding its default TOC commands to the document. Specifying this option is important, the point of this program being to create a custom TOC not the default TOC. In the syntax above the nickname s1 is given to the Heading 1 style. The \ before the s1 is an RTF trigger; it will let RTF software know that a RTF command will follow. However, SAS must first be alerted that RTF commands are included within this SAS program. There are 2 ways of alerting SAS that RTF commands are included in a program 1) ODS escape sequences or 2) RTF direct control words. The details and syntax of these 2 techniques are included in the appendix of this document. The ODS escape sequence technique will be demonstrated here and in the next section the RTF direct control word technique is demonstrated. The techniques are interchangeable. Therefore, either technique could be used in the examples that follow. Now that the \s1 nickname has been declared next declare the ODS escape sequences:

```
ods escapechar='^';
```

Now to mark text in the SAS output with the \s1 style there are several options. The TITLE statement, the ODS TEXT= statement, or the ODS PREPAGE= statement are generally used. To use the TITLE Statement the BODYTITLE option must be added first to the ODS RTF statement when the file is opened (ODS RTF FILE="TOC2.RTF" BODYTITLE ...). Next use the TITLE statement in the following way:

```
TITLE "^R/RTF'\s1' Section 1: Personal Information"; RUN;
```

(SAS FAQ #4018 demonstrates this use of the title statement to create a TOC. I found that this works in version 8 but not in version 9.)

The ODS RTF TEXT= is similar to a SAS Title statement. The Title statement places the specified text on the top of every page of output until it is cancelled or replaced. With the ODS RTF TEXT= statement the text is displayed on only one page in the place where the command is issued. If the ODS RTF TEXT= statement is specified between two procs the text is generally displayed at the bottom of the output from first proc (similar to a footnote). Unlike the TITLE statement the ODS TEXT= statement can be used without a proc. Use the ODS TEXT statement in the following way:

```
ODS RTF TEXT="^R/RTF'\s1' Section 1: Personal Information"; RUN;
```

The ODS RTF PREPAGE= statement is experimental in version 9. There is very little difference between the TEXT= statement and the PREPAGE= statement. With the ODS RTF PREPAGE= statement the text is displayed on only one page in the place where the command is issued. If the ODS RTF PREPAGE= statement is specified between two procs the text is generally displayed at the TOP of the output for the second proc (similar to a title). Like the TITLE statement the ODS PREPAGE= statement can not be used without a proc. Use the ODS PREPAGE statement in the following way:

```
ODS RTF PREPAGE="^R/RTF'\s1' Section 1: Personal Information"; RUN;
```

The full program using TOC building method 2 and the PREPAGE= statement to mark the TOC entries:

```

ODS RTF FILE="TOC2.RTF" startpage=no keepn NOTOC_DATA wordstyle='(\s1 Heading 1;);
ods noproctitle;
ods escapechar='^';
TITLE; RUN;

ODS RTF PREPAGE="^R/RTF'\s1' Section 1: Personal Information"; RUN;
proc freq data=a; table q1-q5*groups /norow nopercent nofreq; run;

ODS RFT PREPAGE="^R/RTF'\s1' Section 2: Children in the home"; RUN;
proc freq data=a; table q6-q10*groups /norow nopercent nofreq; run;
ods rtf close;

```

Now that the TOC entries are marked the next step is to build the TOC table. Open the document (TOT2.RTF) in MS Word. Place the cursor at the beginning of the document and select **Insert**→**Reference**→**Index and Tables**→**Table of Contents**→**OK**. The resulting TOC:

	1
Section 1: Personal Information.....	1
Section 2: Children in the home.....	2

TIP: This method can work in SAS version 8 with a few modifications. There are instructions on SAS' website regarding this. See the reference section of the paper for a link to this page.

### USING SAS TO PASS RTF COMMANDS: TABLE OF CONTENTS METHOD 3

Earlier it was revealed that the \tc RTF command is used to mark TOC entries and the \toc RTF command is used to generate the TOC table. The \tc and \toc commands are standard RTF commands so these commands should work in any RTF reader compliant software. From within SAS RTF commands can be embedded any where literal text strings are allowed (e.g., titles, footnotes, labels, character variable values). For more details about the syntax of using SAS programs to embed RTF commands in RTF documents refer to the appendix of the paper. The remainder of this section will demonstrate the use of this TOC building technique.

Step one is to alert SAS that some literal strings within the program contain embedded RTF commands. This is done by setting the PROTECTSPECIALCHARS attribute to off. Use Proc Template or in-line styles specified directly on a proc to accomplish this. There is currently no command to turn this attribute off for the whole document instead this attribute is turned off for each section of the output. What follows is sample code that turns the PROTECTSPECIALCHARS attribute to off for most sections of the output page.

```

proc template ;
define style RTFCW;
parent=styles.rtf;
style header from header /protectspecialchars=off;
style data from data /protectspecialchars=off;
style rowheader from rowheader / protectspecialchars=off;
style systemtitle from systemtitle /protectspecialchars=off;
style systemfooter from systemfooter /protectspecialchars=off;
style usertext from usertext /protectspecialchars=off           OutputWidth = 100%;
style byline from byline /protectspecialchars=off;
end; run;

```

Note that in the USERTXT sections of Proc Template OUTPUTWIDTH=100% was also specific. SAS determines the width of the output by the width of the table it has generated and then centers the table on the page. OUTPUTWIDTH=100% will allow the TOC table to span the entire width of the page.

As noted earlier the ODS RTF TEXT= statement is similar to a SAS Title statement. The Title statement places the specified text on the top of every page of output until it is cancelled or replaced. With the ODS RTF TEXT= statement the text is displayed on only one page in the place where the command is issued. Here it is used to create headers before the first table in each section and also to "mark" these headers as TOC entries using the \tc RTF command.

```

ODS RTF TEXT="{\tc Section 1: Personal Information}"; RUN;

```

Having marked the headers using the method above, the next step is building the TOC table at the top of the document. Just before the first table-producing procedure the following programming lines are added:

```
ODS RTF TEXT="Table of Contents"; run;

ODS RTF TEXT="{\field{\*\fldinst {\TOC \f \h} } }"; RUN;
```

The first ODS RTF TEXT= statement inserts a line in the destination document that reads "Table of Contents." The second ODS RTF TEXT= statement inserts a TOC field command at the top of the destination document. The \f switch requests that all text throughout the document marked with the \tc entry codes be collected and added to the TOC. The \h switch indicates that TOC entries should also be set up as hyperlinks. When opening this document in Word, you must press Ctrl-A+F9 to generate the TOC table.

The complete program follows. Notice that the new style template (RTFCW) created with the Proc Template command is referenced when the document is opened on the first ODS statement

```
run;
ODS RTF FILE='NESUG_TOCV9.RTF' STYLE=RTFCW startpage=no bodytitle keepn NOTOC_DATA;
ods noproctitle;

title; RUN;
ODS RTF TEXT="Table of Contents"; run;

ODS RTF TEXT="{\field{\*\fldinst {\TOC \f \h} } }"; RUN;

ODS RTF TEXT="{\tc Section 1: Personal Information}"; RUN;

proc freq data=a; table q1-q5*groups /norow nopercnt nofreq ; RUN;

ODS RTF TEXT="{\tc Section 2: Children in the Home}"; RUN;

proc freq data=a; table q6-q10*groups /norow nopercnt nofreq ; RUN;
ods rtf close;
```

Tip: This method works in SAS Version 8 with only 2 modifications. The Proc Template step is not needed and the NOTOC\_DATA option is not needed in SAS version 8.

The resulting TOC has page numbers and entries that are hyperlinked to the beginning of each section:

<b>Table of Contents</b>	
Section 1: Personal Information.....	1
Section 2: Children in the Home .....	2

Because SAS allows RTF commands to be embedded anywhere a literal text string is allowed this method is very flexible and can be used to develop condensed or very detailed TOC tables. For example to make the above TOC more detailed we could also mark the variable labels for inclusion in the TOC:

```
data a; set a;
label
q1="{\tc Married?}"
q2="{\tc Have your driver's license?}"
q3="{\tc Currently in School?}"
q4="{\tc Own your own home?}"
q5="{\tc Covered by a health Plan?}"
q6="{\tc Any pre-school children in the home?}"
q7="{\tc Any school aged children in the home?}"
q8="{\tc Any boys in your home?}"
q9="{\tc Any girls in your home?}"
q10="{\tc Have any of the children had health issues in the last year?}";
```

```

ODS RTF FILE="&progrname..RTF" startpage=no keepn NOTOC_DATA STYLE=RTFCW;
ods noproctitle; title; RUN;

ODS RTF TEXT="Table of Contents"; run;

ODS RTF TEXT="{\field{\*\fldinst {\TOC \f \h} } }"; RUN;

ODS RTF TEXT="{\tc Section 1: Personal Information}"; RUN;
proc freq data=a; table q1-q5*groups /norow nopercent nofreq; run;

ODS RTF TEXT="{\tc Section 2: Children in the home}"; RUN;
proc freq data=a; table q6-q10*groups /norow nopercent nofreq; run;
ods rtf close;

```

The more detailed TOC:

	1
Table of Contents	
Section 1: Personal Information.....	1
<b>Married?</b> .....	1
<b>Have your driver's license?</b> .....	1
<b>Currently in School?</b> .....	1
<b>Own your own home?</b> .....	2
<b>Covered by a health Plan?</b> .....	2
Section 2: Children in the home.....	2
<b>Any pre-school children in the home?</b> .....	2
<b>Any school aged children in the home?</b> .....	3
<b>Any boys in your home?</b> .....	3
<b>Any girls in your home?</b> .....	3
<b>Have any of the children had health issues in the last year?</b> .....	4

Remember the format of this TOC (e.g. the font sizes, font types, and bolding) and the other TOCs in this paper can be changed by selecting **Insert→Reference→Index and Tables**. Click on the **Table of Contents tab**; then select a **style** from the drop-down box (**initially showing From Template...**) at the bottom that appears to the right of Format. If the word "ERROR" appears in your TOC, this only means that the TOC needs to be updated (regenerated). As before, press **Ctrl+A and F9**; then select "**entire table**" to update the TOC.

## CONCLUSION

Regardless of which method used, there are three steps that must be taken to build a table of contents in a RTF document. First, intersperse the text (headings etc.) throughout the document that will eventually become entries in the TOC table. Next, as these headings are added throughout the document, they must also be marked with codes that indicate that this is the text that should become a part of the TOC. Third, create the table of contents. In RTF documents, this is also done with codes.

In Version 9, a new ODS statement option has been added: CONTENTS=YES. When this option specified SAS automatically handles the TOC three-step building process. Because it is automatic, SAS determines how many TOC entries to add to the TOC and the wording of those TOC entries. Using the ODS PROCLABLES statement, the CONTENT= procedure option, or even PROC TEMPLATE, the programmer can increase control over the number of TOC entries and the wording. Populate the TOC when the RTF document is opened by pressing Ctrl-A+F9.

The second method presented involves using the ODS RTF WORDSTYLE= option to assign a nickname to the Heading 1 style. The Heading 1 style is a default style available in MS Word. Next text is interspersed throughout the document, marked with the Heading 1 nickname (steps 1 and 2 of the TOC-building process). After the document is opened in MS Word the TOC is build using standard menu selections. Using this method, the programmer decides exactly how many entries become part of the TOC, and specifies the wording of these TOC entries.

The last method presented uses the ODS RTF TEXT statements to intersperse headings throughout your document. These statements include native RTF commands embedded with the headings that mark the text as TOC entries (steps 1 and 2 of the TOC-building process). Next, a TOC is built near the top of the document; again, this is accomplished by passing native RTF commands using the ODS RFT TEXT statement. Populate the TOC when the RTF document is opened by pressing Ctrl-A+F9. This method provides hyperlinks and page numbers, and, using this method, you decide exactly how many entries become a part of TOC and the wording of these TOC entries.



The Version 9 ODS CONTENTS=YES option creates a nicely formatted, detailed table of contents and is easy to use by programmers of all skill levels. Marking text with the Heading 1 style or with the \tc RTF command require additional programming effort but these methods allow for greater control to create customized TOCs. Further, either of these 2 methods can work in SAS version 8.

## REFERENCES

SAS Live Web Training Class: *Advanced Output Delivery System Topics Course*.

Support.sas.com FAQ on proc template and ODS:

<http://support.sas.com/faq/040/FAQ04018.html> TOC via \R/RTF" \s1 "  
<http://support.sas.com/rnd/base/topics/odsrtf/rtf901.html> Wordstyle= and Prepage=  
<http://support.sas.com/faq/041/FAQ04148.html> TOC using links - specifics  
<http://support.sas.com/faq/041/FAQ04154.html> TOC via bookmarks – more general info  
<http://support.sas.com/faq/032/FAQ03239.html> TOC Proc Template contents\_label option  
<http://support.sas.com/faq/032/FAQ03240.html> TOC contents=

RTF specification information:

<http://msdn.microsoft.com/library/?url=/library/en-us/dnrftspec/html/rftspec.asp>  
<http://www.oreilly.com/catalog/rtfpg/> Resource guidebook for sale

Microsoft's TOC and Index tutorials:

<http://office.microsoft.com/training/training.aspx?AssetID=RC011356771033>  
<http://office.microsoft.com/training/training.aspx?AssetID=RC011525811033>

Papers:

How to Do Horrible Things with Raw RTF Specifications and the SAS RTF ODA,  
<http://support.sas.com/rnd/papers/sugi27/rtfoda.pdf>

Skinning the Cat This Way and That: Using ODS to Create Word Documents that Work for You, Elizabeth Axelrod, David Shamlin, <http://www2.sas.com/proceedings/sugi29/084-29.pdf>

Markup 101: Markup Basics, Cynthia Zender, <http://support.sas.com/rnd/papers/sugi29/markup-basics.pdf>

Applying Microsoft Word Styles to ODS RTF, Output Lauren Haworth,  
<http://www2.sas.com/proceedings/sugi30/043-30.pdf>

## ACKNOWLEDGMENTS

Ihno Lee, Paulette Staum, Wayne Hester, Chevell Parker, and Cynthia Zender all helped by reviewing the paper, sharing code, recommending resources, proofreading, and ...

## CONTACT INFORMATION

Electra Small  
 Senior Technical Associate II  
 MDRC  
 212-340-8824 (Direct)  
 16 East 34th Street, 19th Floor  
 New York, NY 10016  
[Electra.Small@mdrc.org](mailto:Electra.Small@mdrc.org)

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

RTF documents are text documents. Open an RTF document with any text editor (e.g., Notepad); RTF commands are interspersed with the text, as shown in the left-hand column in the following table. Open this document with any RTF reader software (e.g., MS Word); formatted text as shown in the right-hand column is displayed.

Text with RTF control strings or commands	Text rendered in MS Word
{I was }{\b very }{\ul happy}{ to hear}.	I was <b>very</b> <u>happy</u> to hear.

The RTF command `\b` is responsible for formatting the word “very” into boldface type, and `\ul` causes the word “happy” to be underlined. Note that the RTF commands (or RTF control words) are translated into formatting when such documents are opened with RTF reader software. The focus of this paper concerns similar RTF commands that can signal RTF readers to mark index entries and build the document index. To use a SAS program to build an automated document with an index these steps are followed: 1) use SAS to build tables (via Proc Tabulate, Report, Freq, Means...) 2) use ODS to send the tables to the RTF destination 3) use SAS to pass along RTF commands (that you supply) to the RTF destination without interfering with your SAS program.

Notice in the table above {and \ are RTF evaluation triggers. These symbols don’t show in the display document they are triggers that request evaluation by the RTF software. But the {and \ symbols could also be part of literal text that might need to be displayed in the document. For example: TITLE “Printed from D:\Files”; The \ symbol in the text of the Title statement should be displayed, it is not the start of an RTF control word to be evaluated by the reader software. By default SAS signals the RTF reader software that all special characters (e.g. \ {) are to be displayed and not evaluated. In SAS’ terms it protects special characters. To embed RTF commands in a destination document SAS syntax must be used that signals to SAS that the text that follows is not to be protected in the destination document, otherwise {b very }{\ul happy} will be displayed literally as {b very }{\ul happy} in the document.

The destination is the RTF document being created through SAS. The following command, ODS RTF FILE="C:\Files\TEST.RTF", creates an RTF destination document, TEST.RTF, on the C: drive, in the Files directory. There are two ways to place, or “pass,” RTF commands into the TEST.RTF document.

Pass Technique 1 utilizes the ODS ESCAPECHAR statement and escape character sequences. Similar to the macro trigger characters (%,&), the escape character lets the SAS compiler know that what follows are RTF commands to be passed to the destination document. First, declare an ODS escape character: ODS ESCAPECHAR='^'; From this point forward, using the declared ODS escape character combined with any of the allowed escape sequence codes causes these codes to be passed to the destination document. Escape sequence code combinations are allowed wherever SAS allows a text string (e.g., titles, footnotes, labels, character variable values). The following example shows an ODS escape sequence. Example: TITLE '^R/RTF"\b" Very ^R/RTF"\b0\ul" Happy';. Notice the RTF commands are surrounded by quotes within the title quotes. This statement passes the `\b`, `\b0` (bold off), and `\ul` RTF formatting commands to the RTF document, and **Very** Happy becomes the title line in your output. The `^R` signals SAS that what follows is not to be protected from RTF evaluation. The `/RTF` signals that the text string that follows should be passed to the RTF document only. On occasion programmer use ODS to write to two or more types of destinations at the same time. If writing to RTF and PDF documents simultaneously the `/RTF"\b"` and `/RTF"\b0\ul"` commands are not passed to the PDF destination document. This is important since these commands are only understood by RTF readers. This technique uses a SAS raw text (^R) escape sequence, which is handy when only a few control words need to be unprotected. The syntax of this technique involves nesting single and double quotes; therefore, including any text with macro triggers or literal quote marks becomes a challenge.

Pass Technique 2 utilizes RTF control words directly. To use this technique PROTECTSPECIALCHARS=OFF must be specified as an inline style on the proc (print, tabulate, report) or as a style attribute in a PROC TEMPLATE statement before the proc. This tells SAS to pass special characters (<> &/), tags, and control words to the destination document without any translation, which allows the reader software to decide how to handle them. Once PROTECTSPECIALCHARS is turned off RTF control words can be used in your SAS program wherever a text string is allowed. Example: TITLE '\b Very \b0 \ul Happy'. This statement passes the `\b`, `\b0`, and `\ul` RTF formatting commands to the RTF document, and **Very** Happy becomes the title line in the output. Unfortunately, SAS does not currently have a way of setting the PROTECTSPECIALCHARS attribute to off for a whole document, instead commands must be issued to turn this attribute to off in each area of the output where RTF control words will be used (e.g. titles and footers, by lines, data, column headers, row headers...). Programmers who are not comfortable with PROC TEMPLATE or in-line styles find this cumbersome.