Regaining Some Control Over ODS RTF Pagination When Using Proc Report
Gary E. Moore, Moore Computing Services, Inc., Little Rock, Arkansas

ABSTRACT
When creating RTF files using SAS Proc Report, some of the control over pagination is lost because third party programs like Microsoft Word determine the pagination during rendering. Often, this pagination does not appear appropriate or esthetically pleasing.

We can regain some of that control using the break option with Proc Report and manually setting page break points in the data. This requires an examination of RTF pagination while answering some questions about the data and report. What elements factor into the determination of pagination? What types of pagination are possible? What type of pagination is required?

This paper examines these questions and provides a SAS Macro that will assist in manually setting break points for pagination.

INTRODUCTION
Because RTF pagination is done by third party programs like Microsoft Word at rendering, the default pagination can be unexpected and undesirable. It can be complicated even more with the customization of reports by the client. Some simple examples follow.

THE PROBLEM
Undesirable results can occur as in this example of a simple summarization table.

![Table Example](image-url)
THE VARIABLES

Report structure
The number of available lines for the body of the report will vary based on numerous factors that involve the structure of the report. Among these are: 1. The font size of the report. 2. The number of title (header) and footer lines in the report. 3. The size of the columns and the column label length.

Report Content
The number of observations displayable on a given page of the report depends on the report structure and the available lines for the body of the report. But it will also depend on how many groups in the data are separated by blank lines on a given page and on the number of observations and number of variables that have flowed (wrapped) values within the columns on a given page.

REPORT STRUCTURE

FONT SIZE
Larger font size will leave less room for the number of lines per page.

Output 2. Large font (40pt) effect on available lines

Smaller font size will leave more room for the number of lines per page.

Output 3. Small font (12pt) effect on available lines
TITLES (HEADERS) AND FOOTERS

The number of titles (headers) and footers will affect the available number of lines per page.

Output 4. Titles (headers) and footers effect on available lines (12pt font)

COLUMN SIZE AND COLUMN LABEL LENGTH

The size of the columns and the column label length will affect the available number of lines per page.

Output 5. Column size and Column label length effect on available lines (12pt font)
REPORT CONTENT

SEPARATING GROUPS

Separating groups in the data with blank lines will affect the number of observations on a given page.

Output 6. Separating groups (12pt) effect on number of observations per given page

FLOWED (WRAPPED) VALUES WITHIN COLUMNS

The number of observations and variables that have flowed (wrapped) values within the columns will affect the number of observations on a given page. Because of the different ways Proc Report handles flowed variables, finding the best number for the length of the columns that works for the entire report can be an iterative process. Proc Report tries to split words in logical places, like spaces and punctuation. When it does this, it can use more lines than expected, particularly when there are many short words that do not use the entire column width.

Output 7. Flowed (wrapped) values within columns (12pt) effect on number of observations per given page
REGAINING CONTROL

After examination of issues that affect the behavior of pagination of RTF files, we can create a plan to control the behavior. Our plan of action is the creation of a paging macro (NextPage in Appendix) that addresses these issues and creates manual break points to be used with Proc Report and the break option. Because the pagination is determined during rendering, as the file is loaded and processed by Microsoft Word, handling these issues can be an iterative process. After looking at all of the factors that go into determining lines and observations per page, we can make a good guess as to the number of lines available and the number of characters available in the columns for flowed (wrapped) variables. But, it can be an iterative process to determine the correct number of lines and the correct column length that works for the entire report.

MACRO NEXTPAGE

`%macro NextPage(indata=FINAL,byvars=,flowvars=,flowlens=,skipvars=,pagevars=,nlines=18,ngrps=,outdata=);`

PARAMETERS:
1. INDATA (Required) The input dataset name. Default = FINAL.
2. BYVARS (Optional) Sorting variables for report.
3. FLOWVARS (Optional) flowing variables for report.
4. FLOWLENS (Optional) column length needed for flowing variables.
5. SKIPVARS (Optional) skip variables for report; should be included in BYVARS.
6. PAGEVARS (Optional) page variables for report; should be included in BYVARS.
7. NLINES (Required) lines per page for body of report after counting for header, footnotes, and column labels.
   The default = 18.
8. OUTDATA (Required) The output dataset name for the report.
   The default = &INDATA.

FUTURE ENHANCEMENT

There are possibly better ways to estimate or control the flowing (wrapping) of text within columns (ref. MWSUG PO04-2011). There are some special features that we occasionally want to add to reports. It can be helpful to control the number of groups per page. Controlling the undesirable splitting of groups between pages (Widow Variables). When groups are split between pages adding a line that indicates it is continued from the previous page is nice.

CONCLUSION

The Microsoft Word pagination during rendering can create a loss of pagination control. But, after looking at the factors that determine this pagination, we can regain control by creating break points within the data prior to sending it to Proc Report.

REFERENCES

MWSUG PO04-2011 - Pagination in Clinical Trial PROC REPORT ODS - Chao Su, William Conover -
Nesug 121 - Practical Solutions for Pagination, Indentation, and Mixed-Text Alignment in ODS RTF Word Documents - Hui Song, PRA International, Horsham, PA
CONTACT INFORMATION

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

Name: Gary E. Moore
Enterprise: Moore Computing Services, Inc.
Address: 2006 Beckenham Cove
City, State ZIP: Little Rock, AR 72212
Work Phone: 501-225-8689
E-mail: gary@moorecsi.net
Web: www.moorecsi.net

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 that Estimates Pagination based on observations, levels, and flowing variables

Created By : Gary Moore
Moore Computing Services, Inc.
2006 Beckenham Cove
Little Rock, AR 72212
501-225-8689
www.moorecsi.net

Name : NextPage.sas

PARAMETERS:
1. INDATA --- (Required) The input dataset name.
   Default = FINAL.
2. BYVARS --- (Optional) Sorting variables for report
3. FLOWVARS --- (Optional) flowing variables for report
4. FLOWLENS --- (Optional) column length needed for flowing variables
5. SKIPVARS --- (Optional) skip variables for report; should be included in BYVARS
6. PAGEVARS --- (Optional) page variables for report; should be included in BYVARS
7. NLINES --- (Required) lines per page for body of report after counting for header, footnotes, and column labels
   The default = 18.
8. OUTDATA --- (Required) The output dataset name for the report
   The default = &INDATA.

*******************************************************************************

%macro NextPage(indata=FINAL,byvars=,flowvars=,flowlens=,skipvars=,pagevars=,nlines=18,ngrps=,outdata=);

options mprint;
run;

/* Number of FLOWVARS variables */
%let nflvs_=0;
%if %length(&flowvars) > 0 %then %let nflvs_=%sysfunc(countw(&flowvars));

/* Number of FLOWLENS variables */
%let nflls_=0;
%if %length(&flowlens) > 0 %then %let nflls_=%sysfunc(countw(&flowlens));

/* Number of SKIPVARS variables */
%let nskvs_=0;
%if %length(&skipvars) > 0 %then %let nskvs_=%sysfunc(countw(&skipvars));

/* Number of PAGEVARS variables */
%let npgvs_=0;
%if %length(&pagevars) > 0 %then %let npgvs_=%sysfunc(countw(&pagevars));

/* Dummy by variable */
data _indata;
   set &indata;
   byv_ = 1;
run;

%let byvars=byv_ &byvars;
proc sort data=_indata;
   by &byvars;
data _indata;
   set _indata;
   by &byvars;
   retain LINECNT PGBK 0;
run;

*******************************************************************************
/* Estimate number of lines needed for each flowed variable */
%if %length(&flowvars)>0 %then
  %do z_ = 1 %to &nflvs_
     _len&z_ = 1;
     if %scan(&flowvars,&z_) ne ' ' then
       _len&z_ = ceil(length(%scan(&flowvars,&z_))/%scan(&flowlens,&z_));
  %end;
/* Increment line count by 1 or by max of all needed flow variable lines */
LINECNT = LINECNT + max(
  %if %length(&flowvars)>0 %then
    %do z_ = 1 %to &nflvs_
       _len&z_ ,
    %end;
1);
/* Increment line count for each skip variable */
%if %length(&skipvars)>0 %then
  %do z_ = 1 %to &nskvs_
     if first.%scan(&skipvars,&z_) then LINECNT = LINECNT + 1;
  %end;
/* Increment page count by 1 each time we use up the lines per page and for each page variable */
if %if %length(&pagevars)>0 %then
  %do z_ = 1 %to &npgvs_
     first.%scan(&pagevars,&z_)or
  %end; 
LINECNT >= &nlines then 
  do;
    PGBK = PGBK+1;
    LINECNT=0;
  end;
run;
%if %length(&outdata)=0 %then %let outdata=&indata;
data &outdata;
  set _indata(drop=(linecnt _len:));
run;
%mend NextPage;