

The Power of PAGEOF

(A Valuable Page Numbering Macro)

Kelly Felty, PPD Inc., Morrisville, NC
Diane Nicholson, PPD Inc., Morrisville, NC

Abstract

PAGEOF is an innovative macro that counts the number of pages in a report and displays the resulting page count in one of several available styles. This macro's benefits are:

- the ease of displaying the page count in the Page X of N format
- it works with both the Paul Wehr Print Driver¹ and DATA _NULL_ with PUT statements

How It Works

PAGEOF makes two passes through the report. The first pass counts the number of pages in the report based on the number of control characters. These characters may vary from system to system and are easily modified in the macro itself. The second pass writes the page number in the selected style at the specified location.

Call It

The following is the PAGEOF macro call with the default settings:

```
%PAGEOF(f_name = ,
         ls      = 80,
         ps      = 66,
         style   = PAGEOF,
         justify = RIGHT,
         location = TOP,
         driver  = NULL);
```

f_name - Specifies the input/output filename.

ls - Determines the number of columns on the page. The default is 80.

ps - Determines the number of lines on the page. The default is 66.

style - The format of page numbering desired. The default is PAGEOF. Listed are the available formats.

```
N          - 10
NP         - (10)
P_N        - p. 1
P_NP       - (p. 1)
PAGEN     - Page 4
PAGENP    - (Page 4)
PAGEOF    - Page 1 of 20
PAGEOFP   - (Page 1 of 20)
```

justify - Determines the justification of the page number. The default is right justification. Other options include LEFT and CENTER justification, or the actual column offset.

location - Determines where the page number will print. The default is TOP, and other options are BOTTOM or #PAGE. The #PAGE option prints the page number in the location of the #PAGE string on the report.

driver - Determines whether standard data null with put statements processing is used or the Print Driver System¹. The default is standard data null with put statements. Options are NULL or RTF.

Here It Is

```
%macro pageof(f_name =,
              ls      = 80,
              ps      = 66,
              style   = PAGEOF,
              justify = RIGHT,
              location = TOP,
              driver  = NULL);
```

```

* -----
Rename the input table/listing file to a
temporary file
-----;
x "rename &f_name &sysjobid..listing";

filename _temp_ "&sysjobid..listing";
filename __listfl "&sysjobid..listing";

filename _table_ "&f_name";

%let sigspace='FEFF'x;

%* -----
Determine where to put the page number
-----;

%if %substr(%upcase(&justify),1,1) eq L
%then
  %let _at = 1;
%else %if %substr(%upcase(&justify),1,1)
eq C %then
  %let _at = ROUND((&ls + 1 - LEN) / 2);
%else
  %if %substr(%upcase(&justify),1,1) eq R

    %then %let _at = (&ls + 1 - LEN);
%else %if 2 <= %eval(&justify) < &ls
  %then %let _at = &justify;

options ls=&ls ps=&ps;

data _null_;
  %if %upcase(&style) eq PAGEOF or
  %upcase(&style) eq PAGEOFFP or
  %upcase(&driver) eq RTF %then %do;

  infile _temp_ noprint eof=eof1
    length=length1;
  input @1 line $varying200. length1;

  if (substr(line,1,1) eq '1') or
  index(upcase(line),\PAGE') or _n_=1
  then do;

    pages + 1;

    L_count = 0;
  end;

```

```

if (substr(line,1,1) eq '0') then
  L_count + 1;

  L_count + 1;

return;

eof1:
  put L_count=;

  call symput('_tpage',
    compress(put(pages,3.) ) );
%end;

infile __listfl noprint eof=eof2
  length=length2;

file _table_ noprint notitles;

do while(1);
  length cc $1 subline txt $200;
  input @1 line $varying200. length2;

  if upcase("&driver") = 'NULL' then
    cc = substr(line,1,1);

  if upcase("&driver") = 'NULL' then

    subline = substr(line,2);
  else if upcase("&driver") = 'RTF' then
    subline = substr(line,1);

  length3 = length(subline);

  if cc eq '1' or
  (index(upcase(subline),#\PAGE')
  and "&driver" = "RTF") then do;
    page + 1;

  * -----
  Output page break special code
  -----;
  put '0C'x;

  * -----
  Reset the line counter
  -----;
  linecnt = 0;
end;

```

```

else if cc eq '0' then do;
  put;

  linecnt + 1;
end;

subline = translate(subline, ' ', &sigspace);

* -----
  Determine the location of page number
  -----;

%if %upcase(&style) eq PAGEOF or
  %upcase(&style) eq PAGEOFFP
%then %do;
  if (%upcase("&location") eq "TOP" and
    cc eq '1') or
    (%upcase("&location") eq "BOTTOM"
    and linecnt eq (L_count - 1) ) or
    (index(upcase(subline), '#PAGE') )
  then do;
%end;

%else %do;
  if (%upcase("&location") eq "TOP" and
    cc eq '1') or
    %upcase("&location") eq "BOTTOM"
    or (index(upcase(subline), '#PAGE') )
  then do;
%end;

  %if %upcase(&style) eq PAGEOFFP
  %then %do;
    txt = 'Page ' !!
      compress(input(put(page,8.),$8.))
      !! ' of ' !!
      compress(put(pages,8.))||";
  %end;

  %else
  %if %upcase(&style) eq PAGEOF
  %then %do;
    txt = 'Page ' !!
      compress(input(put(page,8.),$8.))
      !! ' of ' !! compress(put(pages,8.));
  %end;

  %else
  %if %upcase(&style) eq PAGENP
  %then %do;
    txt = 'Page ' !!
      compress(input(
        put(page,8.),$8.))||";
  %end;

  %else
  %if %upcase(&style) eq PAGEN
  %then %do;
    txt = 'Page ' !!
      compress(input(
        put(page,8.),$8.));
  %end;

  %else
  %if %upcase(&style) eq P_NP
  %then %do;
    txt = '(p. ' !!
      compress(input(
        put(page,8.),$8.))||";
  %end;

  %else
  %if %upcase(&style) eq P_N
  %then %do;
    txt = ' p. ' !!
      compress(input(
        put(page,8.),$8.));
  %end;

  %else
  %if %upcase(&style) eq NP
  %then %do;
    txt = "("||compress(input(
      put(page,8.),$8.))||";
  %end;

  %else
  %if %upcase(&style) eq N
  %then %do;
    txt = "
      ||compress(input(
        put(page,8.),$8.));
  %end;

* -----
  Replace #Page in Print
  Driver or Data Null
  -----;

* Print Driver ;
if upcase("&driver") = 'RTF'
then do;
  rtfpos = index(upcase
    (subline), '#PAGE');
  txt = trim(left(txt))||substr(
    subline, rtfpos+5);

```

```

len = length(txt);

substr(subline,rtfpos,len) = txt;
end;

else do;
  len = length(txt);
  * Data Null ;
  substr(subline,&_at,len) = txt;
end;

* -----
  Space out '#PAGE'
  -----;
if (index(uppercase(subline),'#PAGE') )
then do;
  substr(subline,index(uppercase(
    subline),'#PAGE'),5) = '  ';
  subline = translate(subline,
    ' ',&sigspace);
end;

length3 = length(subline);
end;

put @1 subline $varying200. length3;

linecnt + 1;

end;

eof2:
  stop;
run;

x "delete/noconfirm &sysjobid..listing;*";

filename _temp_ clear;
filename __listfl clear;
filename _table_ clear;

%mend pageof;

```

Wrap Up

PAGEOF provides an easy way to apply various styles of page numbering to a report. PAGEOF is easily modified for different control characters and/or new styles of page numbering.

Acknowledgment

We would like to acknowledge Chuan-Feng Shih for writing this wonderful macro.

Authors

Diane Nicholson
E-mail: Diane.Nicholson@rtp.ppdi.com

Kelly Felty
E-mail: Kelly.Felty@rtp.ppdi.com

PPD PHARMACO
4025 Paramount Parkway
Morrisville, NC 27560
(919) 380-2000

¹ “%Print Drivers: Teaching SAS to Speak the Many Languages of Document Publication”, Proceedings of the 21st Annual SAS User’s Group Meeting, International.