

OSR – Output Macro System with Proc Report and Central Storage of Titles and Footnotes in Excel

Peter Gerber, Genzyme, Naarden, Netherlands

Abstract:

OSR is an Output macro system in SAS® code, which enables you to create tables and listing output in a standardized way using proc report and ODS (Output Delivery System). Figures can be done with **OSR** and graphic procedures like proc gplot. An important part of the **OSR** tool is a central repository of titles and footnotes in an Excel® spreadsheet. From here, title and footnote information can be transformed into a SAS dataset and printed automatically into the output. Additionally, page numbering, date and time information, information about the datasets used for programming of the output and standard lines can be printed into the output. The output produced is landscape, pagesize=66, linesize=132 and the font is SAS monospace.

Keywords:

Proc report, output, Excel, Macro system, ODS.

Introduction:

Among other work, statistical programmers in clinical research must produce tables, listings and figures (TLF's) for clinical study reports, interims analysis and other uses. SAS is the software most programmers are working with. There are many possibilities to create TLF's with SAS and often every company has its own system. It is possible to produce TLF's with put statements, proc tabulate, proc print or proc report. With this paper the author wants to show, how easy it is to create listing and post script output in a highly standardized way with a SAS macro system, called **OSR** (Output System with Proc Report), developed at Genzyme Europe B.V in Naarden, Netherlands.

OSR consists of several SAS macros and creates tables and listings with ODS (Output delivery system) and proc report. Additionally it is possible to create figures with **OSR** in combination with graphic procedures (for example proc gplot).

A second and important feature of **OSR** is, that titles and footnotes are stored centrally in an Excel spreadsheet, converted into a SAS dataset and printed with SAS statements into the output. **OSR** controls where titles and footnotes are printed (centered, left or right).

A line between the titles and the core part of the output and a line between the core part and the footnotes, both with the length of 132 characters, is printed automatically into the output (see example output page 5).

OSR runs under SAS version 8 and SAS version 9.

Additional macros calculate and print the page numbering (page xxxx of yyyy) of the listing output and also identify, which permanent datasets were used to create the output, which is also printed at the bottom of the output (these macros are not described in this paper).

The output produced is landscape, pagesize=66, linesize=132, SAS monospace and can be stored as listing file (.lst) or postscript file (.ps) or computer graphics metafile (.cgm)..

The listing file or the postscript file can be converted with Acrobat® distiller into a PDF file and the cgm file can be included into a Word® document..

With **OSR**, it is possible to create one output or several outputs within one SAS program. Also, one output can include several tables or listings with different titles or footnotes or only one table or listing.

OSR system components and example output:

A) EXAMPLE LISTING-OUTPUT-PROGRAM:

The code below is an example for an easy listing and shows, how easy it is to work with **OSR**. Between “ods listing file=...;” and “ods listing close;” it is possible to create several output with different titles or footnote information in one listing file. It is also possible, to create several output in different listing files. As you see, we need only a few additional code lines (see highlighted code) to produce a standardized output (see example output under H)). The program can be copied and adapted easily without including specific code for titles or footnotes or specific study information. When you want to produce a listing with the same layout in another study, copy the program and run it for your new study. The main work is then to fill in the program header.

```
%INCLUDE 'Z:\\Project\\setup\\setup.sas'; *---- include SAS code to define libraries;

proc sort data=stat.dem1 out=dem;          *---- simple data preparation step to provide
    by cohort patid ;                      code for listing;
run;

%let tablename=L-16-2-4-1-demog;          *----- TLF identifier;

%titlfoot;                                *----- main macro of OSR;

ods listing file="%tblpath.&tablename..lst"; *----- open ODS and define
                                           output path;

    *----- typical proc report program to create core part of the output;

proc report data=dem headskip headline missing spacing=4 split="$" nowd nocenter;
    column cohort patid birthdt age ethnic sex;
        define cohort          /order  "Cohort" width=8 center;
        define patid           /order  "Pat. ID" width=7 center spacing=1;
        define birthdt         /display "Date of birth" width=15 center;
        define age              /display "Age$(years)" width=10 center;
        define ethnic           /display "Ethnicity" width=12 center;
        define sex              /display "Sex" width=8 center;
        break after cohort/skip;
        %tlline; * macro to create a line between core part of the output and the
                    footnote part, macro contents a compute statement;
quit;

ods listing close;          *----- close ODS listing;
%pageno;                    *----- opens stored output and calculates page numbering
                             "Page xxxx of yyyy" and printed these information
                             into title 1 of the output ;
```

B) SETUP.SAS:

This program is necessary to include libraries and to initialize project and protocol. Further programs of **OSR** are included. Setup.sas is study specific and system specific and must be adapted according to your programming environment. An example code you can find under Appendix A.

C) TITLES AND FOOTNOTES EXCEL SPREADSHEET:

The programmer must store title and footnote information into an Excel spreadsheet. This Excel spreadsheet includes several columns to store the title and footnote information. This repository of titles and footnotes is a big help for programmers and statisticians because they have now a central place where they can find all titles and footnotes information. Additionally they have an overview which TLFs are produced.

An example of the content of this Excel spreadsheet you can find below:

PhUSE 2007

TLF flag (T-/L-/F-)	Program name	Title number	Title	Footnote number	Footnote	Responsible programme r
char	name	tnum	title	fnum	foot	resp
L-	16-2-4-1- demog	1	APPENDIX 16.2.4-1	1	Note: This is an example footnote.	Peter Gerber
L-	16-2-4-1- demog	2	LISTING OF PATIENT DEMOGRAPHIC S	2	Note: This is an example footnote	
L-	16-2-4-1- demog	3	WITH DUMMY DATA			
F-	testfigure	1	FIGURE 14.2.2	1	Note: this is a test footnote.	Peter Gerber
F-	testfigure	2	ALKALINE PHOSPHATASE VERSUS TIME			
F-	testfigure	3	SAFETY POPULATION			
F-	testfigure	4	PATIENT AAAA			
F-	testfigure	5				

1. From this Excel spreadsheet a csv file with a “#” delimiter is stored additionally. The reason is that we had difficulties to create a SAS dataset with titles and footnote information to use proc import and a csv file is easier to handle. “#” as delimiter is used, because other delimiters could be part of titles and footnotes text information. When necessary, it is possible to use another delimiter.
2. The csv file is included with an infile statement and stored as a permanent SAS dataset (see code in Appendix B: titlfoot.sas).
3. The titles and footnote information will be read from the dataset and printed into the output.
4. The first row shows the labels of the column names in row 2. To run the system, only row 2 in the worksheet is necessary.

D) TITLFOOT.SAS:

This macro helps to create listing files with ODS and proc report and is the main macro of the **OSR** system. The source code you can find under Appendix B.

E) TTLINE.SAS:

This macro is needed to print a 132 character long line between core part of the table or listing and the footnote part. In principle this macro is part of the proc report and includes a compute statement. The source code you can find under Appendix C.

F) EXAMPLE GRAPHIC OUTPUT PROGRAM:

The following program shows an example how to create a graphic output as cgm (computer graphics metafile) file. It is also possible to create a ps (post script) or eps (encapsulated post script) file. The highlighted text show you, that only 3 additional code lines are needed to produce standard figures like in I).

```
%let tablename=F-testfigure;                                *----- graphic identifier;

%INCLUDE 'Z:\XXXX\YYYY\CSR\setup\setup.sas'; *---- include SAS code to define
                                                libraries;

proc format;
  value visa
```

PhUSE 2007

```
0 = "-2" 5 = "0" 20 = "2" 40 = "4" 60 = "6" 80 = "8" 100 = "10" 120 =  
"12" 130 = "13" ;  
run;  
  
data lab;  
  set stat.lbl;  
  where lbloincc="AP" and patid eq 1001;  
run;  
  
proc sort data=lab;  
by patid visid;  
run;  
  
*----- graphic options -----;  
  
goptions reset=all rotate=landscape gunit=pct cback=white fby=HWPSL011  
  colors=(black blue green red orange) htitle=2.0 htext=2.0 ftitle=HWPSL011  
  ftext=HWPSL011 device=cgmoff97 gsfmode=replace gsfname=figure;  
  
axis1 label=(angle=90 h=2 'Alkaline phosphatase (U/L)')  
  minor=none  
  order=(0 to 240 by 10); *-----Y-axis;  
  
** x axis;  
axis2 split = '-' label=( h=2 'Week')  
  value=(h=2) minor=none order=(0 to 130 by 10)  
  offset = (0.5 cm); *-----X-axis;  
  
** symbol characteristics;  
  
symbol1 interpol = join color = black value=dot height=2 l=1;  
  
  %gtitles;  
  
** generate plot;  
  
filename figure "&figpath.&tablename..ps";  
  
proc gplot data=lab uniform;  
  by patid;  
  
  format visid visa. ;  
  plot lborresn*visid/  
    vaxis = axis1  
    haxis = axis2  
  ;  
run;  
  
quit;
```

G) GTITLES.SAS:

Gtitles.sas is the main macro of **OSR** to create graphic output (see example page 10). The source code you can find under D).

PhUSE 2007

H) EXAMPLE LISTING OUTPUT: this output should only show the principle layout (no pagesize=66, no original font size)

Genzyme Corporation
Protocol No. XXX-YYY-ZZZ

Page 1 of 2

APPENDIX 16.2.4-1
LISTING OF PATIENT DEMOGRAPHICS
WITH DUMMY DATA

Cohort	Pat. ID	Date of birth	Age (years)	Ethnicity	Sex
Cohort 1	2003	07DEC1986	18.2	CAUCASIAN	MALE
	2007	11JUN1979	25.7	CAUCASIAN	MALE
	3003	22APR1969	35.9	CAUCASIAN	MALE
	3015	23JUN1978	28.6	CAUCASIAN	MALE
	1001	24MAY1974	30.7	CAUCASIAN	MALE
	1002	04JUL1973	31.6	CAUCASIAN	MALE
	1004	23JAN1973	32.1	CAUCASIAN	FEMALE
	...				
	2001	14APR1979	25.9	CAUCASIAN	FEMALE
	2002	31JUL1979	25.6	CAUCASIAN	MALE
	2004	28NOV1981	23.3	CAUCASIAN	FEMALE
	2005	22OCT1983	21.4	CAUCASIAN	FEMALE
	2006	23DEC1980	24.2	CAUCASIAN	MALE
	2008	28OCT1979	25.3	CAUCASIAN	MALE
	3001	24JUL1976	28.6	CAUCASIAN	FEMALE
	3002	15DEC1971	33.2	CAUCASIAN	FEMALE
	3004	19JUN1980	24.8	CAUCASIAN	MALE
	3005	30MAR1975	30.0	CAUCASIAN	FEMALE
Cohort 2	1009	11MAY1979	26.1	CAUCASIAN	MALE
	1010	22OCT1962	42.7	CAUCASIAN	MALE

Note: This is an example footnote

Note: This is a second example footnote

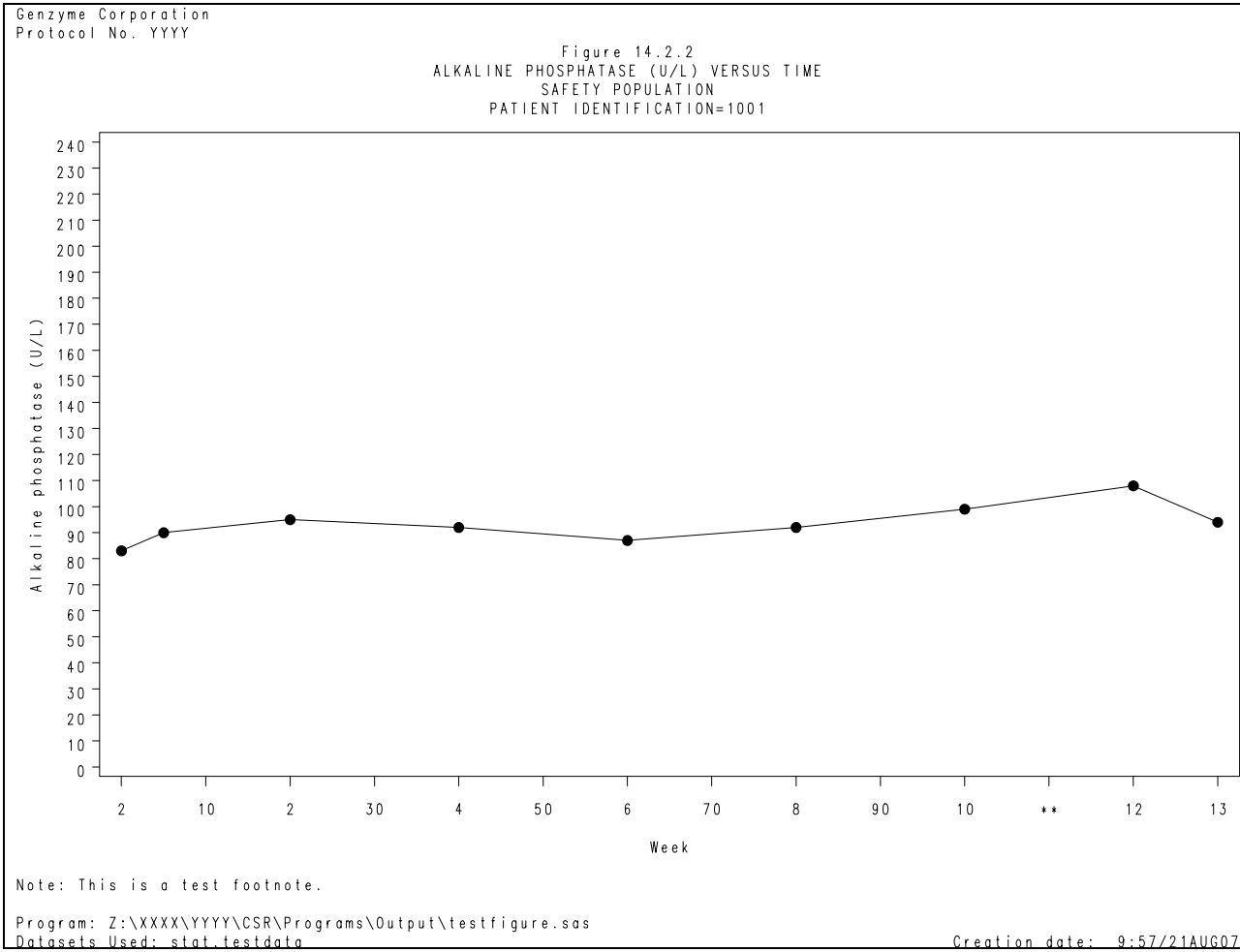
Program Name: A:\PROTOCOL\XXX-YYY-ZZZ\CSR\programs\output\L-16-2-4-1-DEMOG.sas

Creation Date: 13:00/10MAY07

PhUSE 2007

Datasets Used: STAT.Dem1

I) EXAMPLE GRAPHIC OUTPUT: this output should only show the principle layout (no original size).



K) CONCLUSION:

OSR is now the standard tool at Genzyme Europe B.V. to create TLF's for study reports and others. It is a validated tool and until now we had no problems with the system. It seems robust and our hope is that every programmer at Genzyme Europe, who is familiar with proc report can use it and produce TLF's with just a little training.

In the future, it is planned to adapt the system to create rtf files or pdf files with ODS.

The code provided in the paper can be used without permission and with little adaptation it should be possible to use **OSR** for the reader's purposes.

L) ACKNOWLEDGEMENTS:

The author would like to thank Tom Hide for his programming help, Daniel Boisvert for the discussions about programming issues and Dennis Fabrizio for his help to find errors in the programs and to review this paper.

M) APPENDIX:

A.

```
%let drv=A;          *----- drive -----;
%let Project=STUDY DRUG A;    *----- Project name-----;
%let Protocol=%str(XXX-YYY-ZZZ);    *-----Study name----;

%let Company=Genzyme Corporation;

options ps=66 ls=132 nodate nonumber validvarname=upcase missing=""
        nomacrogen nosymbolgen nomprint nosource2
        mstored sasstore=macrolib;

    /** Original data sets library **/
libname ssd      "&drv.:\&Project.\&Protocol.\CSR\data\Raw datasets\";

    /** Analysis data sets library **/
libname stat     "&drv.:\&Project.\&Protocol.\CSR\data\CRT Analysis\";

    /** Formats library **/
libname library  "&drv.:\&Project.\&Protocol.\CSR\formats\";

%let winsasin = %qupcase(%sysfunc(getoption(sysin)));
%let progname = %scan(%scan(&winsasin,-1,%str( \)),1,.);
%let systmdt  = %sysfunc(datetime(),tmdt.);
%let table    = %str(&drv.:\&Project.\&Protocol.\CSR\output\tables\&progname..TBL);

    /* Used in Tables for the output path name */
%let tblpath   = %str(&drv.:\&Project.\&Protocol.\CSR\output\tables\);

    /* Copy of TBLPATH. We need this as some programs may produce both Tables and
Listings, so we may need to restore TBLPATH */
%let tblpth    = &tblpath;

    /* Used in Listings for the output path name */
%let lstpath   = %str(&drv.:\&Project.\&Protocol.\CSR\output\listings\);

    /* Used in Figures for the output path name */
%LET figpath   = %STR(&drv.:\&Project.\&Protocol.\CSR\output\figures\);
```

PhUSE 2007

```
%let lastrun = %str(&programe - %sysfunc(datetime(),dttmp.));

%let pgmpath = %str(&drv.:\&Project.\&Protocol.\CSR\programs\output\);

%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\delds.sas";

*----- for creating output with proc report and central storage of titles and
footnotes -----;

%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\pageno.sas";
%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\tlline.sas";
%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\getds.sas";
%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\titlfootV2.sas";
%include "&drv.:\&Project.\&Protocol.\CSR\setup\Macros\gtitles.sas";

    /* Default TABLENAME */

%let tf=;

    /* Check if TABLENAME has been specified, and get its value if present */

proc sql noprint;
select value into :tf
    from sashelp.vmacro
    where scope="GLOBAL" and name="TABLENAME"
    ;
quit;

    /* Set Table2 to Tablename. This is used as the program name in DATAINST */

%let table2 =%upcase(&tf.);

    /* We want to change output path to LISTINGS for listings */

%macro chklist;
%let ll=%str(L-);
%if %bquote(&table2) ne %then %do;
    %let chk=%substr(%upcase(%str(&table2),1,2));
    %if %bquote(&chk)=&ll %then %let tblpath=%str(&lspath);
%end;
%mend;

%chklist;
```

B.

```
%macro titlfoot;

%local draft getds;

%LET linelen=%EVAL(%SYSFUNC(GETOPTION(LINESIZE)));

title; footnote; run;      *--- clear previous titles or footnotes, if any;

%macro ttl(t,text,tno);

    linelen=%eval(&linelen);
    piece1=SCAN(&text,1,'#');
    piece2=SCAN(&text,2,'#');
    piece3=SCAN(&text,3,'#');

    JL=''; JC=''; JR='';

    DO i=1 TO DIM(pieces);
        IF INDEX(pieces[i],'J=L') THEN JL=LEFT(SUBSTR(pieces[i],4));
        else IF INDEX(pieces[i],'J=C') THEN JC=LEFT(SUBSTR(pieces[i],4));
        else IF INDEX(pieces[i],'J=R') THEN JR=LEFT(SUBSTR(pieces[i],4));
```

PhUSE 2007

```

end;

tno=&tno;

if "&t"="T" then tf="TITLE";
else if "&t"="F" then tf="FOOTNOTE";

txt=repeat(" ",linelen);

txt=j1;

*---- move in left text;

if jc ne "" then do;
    len2=floor(length(trim(jc))/2); *---- get center position of string;
    pos2=floor(linelen/2)-len2; *---- get line length/2 minus string length/2
                                gives start position;
    substr(txt,pos2)=jc; *--- move in center text;
end;

*----- move in right text ;

if jr ne "" then do;
    len3=floor(length(trim(jr))); *----- get string length;
    pos3=linelen-len3; *----- get line length minus string length;
    substr(txt,pos3)=jr; *----- move in right string;
end;

ttl=trim(tf)||put(tno,z2.)||' '||trim(txt)||' ';

if "&t"="T" then ttl=translate(ttl,"#", "\");

if ttl ne "" then CALL EXECUTE (ttl); *----- produce Title and Footnote;

%mend ttl;

data METADATA; *----- read metadata from stored CSV file with # separator;
    drop info p1-p6;
    infile "&drv.:\&Project.\&Protocol.\CSR\Documents\titles.csv" dlm = '#'
    MISSOVER DSD lrecl=32767 firstobs=2 length=len;
    length _tnum _fnum 4 _char $2 _name $20 _title _foot $200;
    input @1 info $varying400. len;

    if info="" then delete;
    p1=index(info,"#");
    if p1 ne 1 then _char=upcase(substr(info,1,p1-1));
    else delete;

    info=substr(info,p1+1);
    p2=index(info,"#");
    if p2 ne 1 then _name=substr(info,1,p2-1);
    else _name="";

    info=substr(info,p2+1);
    p3=index(info,"#");
    if p3 ne 1 then _tnum=input(substr(info,1,p3-1),best.);
    else _tnum=.;

    info=substr(info,p3+1);
    p4=index(info,"#");
    if p4 ne 1 then _title=substr(info,1,p4-1);
    else _title="";

    info=substr(info,p4+1);
    p5=index(info,"#");
    if p5 ne 1 then _fnum=input(substr(info,1,p5-1),best.);
    else _fnum=.;

    info=substr(info,p5+1);
    p6=index(info,"#");

```

PhUSE 2007

```

        if p6 ne 1 then _foot=substr(info,1,p6-1);
        else _foot="";
run;

proc sql;          *----- creates permant SAS dataset with titles and footnotes
                    data;
        create table stat.metadata (label="Titles and footnotes dataset") as select
            a._char  "TLF flag (T-/L-/F-)",
            a._name  "Program name",
                   a._tnum  "Title number",
            a._title "Title" ,
            a._fnum  "Footnote number",
            a._foot  "Footnote"
        from metadata as a
        order by a._char, a._name, a._tnum
        ;
quit;

%getds;      *----- includes macro to identify, which permanent datasets are used
              to create output (not described);

data _null_;   /* Get current time and date */
    tm=time();
    dt=date();
    call symput("tm",put(tm,time5.));
    call symput("dte",put(dt,date7.));
run;

data _null_;
    length ttl piece1-piece3 JL JC JR $200 tf $8;
    length text $200;
    retain tnum fnum 0;

ARRAY pieces [*] piece: ;
    set STAT.metadata(rename=( _title=text _foot=foot)) end=fin;
    where upcase(compress(_char||_name))=upcase("&tablename");

if _n_=1 then do;      *----- the first 2 titles should be standard;
    tno=1;
    ttl="#J=L  Genzyme Corporation  #J=R Page XXXX of YYYY";
    %ttl(T,ttl,tno);
    tno=2;
    ttl="#J=L  Protocol No. &Protocol.";
    %ttl(T,ttl,tno);
end;

tnum=max(_tnum,tnum);
fnum=max(_fnum,fnum);

IF LENGTH(text) GT &linelen THEN PUT "CHECK: Title > &linelen";

if text ne "" then do;      *----- only use title statement if we have a TITLE
                            value;
    text="#J=C "||trim(text);      *----- TITLES are centered;
    %ttl(T,text,tnum+3);
end;

IF foot ne "" THEN do;      *----- only use footnote statement  if we have a
                            footnote value;
    foot=translate(foot,"#","\");      *----- change backslash to hash symbol;
    CALL EXECUTE ( 'FOOTNOTE' || put(fnum,z2.) ||
                   ' "' || TRIM(foot) || '";' );
end;
output;

if fin;

tnum=tnum+4;
JL=repeat("_",131);
CALL EXECUTE ( 'TITLE' || put(tnum,z2.) ||

```

PhUSE 2007

```
      ' "' || LEFT(TRIM(jl)) || '";' );
    foot="#J=L Program Name: &pgmpath.&table2..sas #J=R Creation Date: &tm./&dte.";
*-- the last 2 footnotes should be standard footnotes for all TLFs;

    ftn=fnum+2;

    %ttl(F,foot,ftn);

    foot="#J=L Datasets Used: &getds";

    ftn=ftn+1;

    %ttl(F,foot,ftn);

%if &draft=1 %then %do;
    Foot="#J=C D R A F T      V E R S I O N";      *---- if program runs interactive,
                                                    TEXT will be printed into Footnote;
Ftn=ftn+1;

run;
%mend titlfoot;
```

C.

```
%macro ttline;      *-- code, which creates a 132 character long line between
                    core part of the output and the footnote part;

compute after _page_;
length str $132;
str=repeat("_",132);
    line @1 str $132.;
    * line " ";      *--- line is 132 characters long;
endcomp;

%mend ttline;
```

D.

```
%macro gtitles (pageof=);

*----- to initiate update of metadata -----;
data METADATA;
    drop info p1-p6;
    infile "&drv.:\\&Project.\\&Protocol.\\CSR\\Documents\\titles.csv" dlm = '#'
        MISSOVER DSD lrecl=32767 firstobs=2 length=len;
    length _tnum _fnum 4 _char $2 _name $20 _title _foot $200;
    input @1 info $varying400. len;

    if info="" then delete;

    p1=index(info,"#");
    if p1 ne 1 then _char=upcase(substr(info,1,p1-1));
    else delete;

    info=substr(info,p1+1);
    p2=index(info,"#");
    if p2 ne 1 then _name=substr(info,1,p2-1);
    else _name="";

    info=substr(info,p2+1);
    p3=index(info,"#");
    if p3 ne 1 then _tnum=input(substr(info,1,p3-1),best.);
    else _tnum=.;
```

```

        info=substr(info,p3+1);
        p4=index(info,"#");
        if p4 ne 1 then _title=substr(info,1,p4-1);
        else _title="";

        info=substr(info,p4+1);
        p5=index(info,"#");
        if p5 ne 1 then _fnum=input(substr(info,1,p5-1),best.);
        else _fnum=.;

        info=substr(info,p5+1);
        p6=index(info,"#");
        if p6 ne 1 then _foot=substr(info,1,p6-1);
        else _foot="";

run;

/* Combine sequence numbers back into the title data and output to STAT.METADATA */

proc sql;
create table stat.metadata (label="Titles and footnotes dataset") as select
    a._char  "TLF flag (T-/L-/F-)",
    a._name  "Program name",
    a._tnum  "Title number",
    a._title "Title" ,
    a._fnum  "Footnote number",
    a._foot  "Footnote"
from metadata as a
order by a._char, a._name, a._tnum
;

quit;

%global getds;

title " ";
footnote " ";

/* In case we have no titles */

%let n_titles=0;
%let n_foot=0;

data titles;
retain nttl nftr 0;
set stat.metadata end=fin;
where upcase(compress(_char||_name))=upcase("&tablename");
ttl="TTL"||left(put(_tnum,2.));
ftt="FTT"||left(put(_fnum,2.));
if _tnum ne "" then do;
    nttl+1;
    call symput(ttl,_title);
end;
if _fnum ne . then do;
    nftr+1;
    if _foot ne "" then call symput(ftt,_foot);
    else call symput(ftt," ");
end;
if fin then do;
    call symput("n_titles",nttl);
    call symput("n_foot",nftr);
end;
run;

%let n_titles=&n_titles;
%let n_foot=&n_foot;

data _null_;
dt=put(date(),date7.);
tm=put(time(),time5.);

```

PhUSE 2007

```
call symput("dt", trim(tm)||"/"||trim(dt));
run;

%getds; *----- includes macro to identify, which permanent datasets are used
          to create output (not described in this paper);

title1 j=1 "%trim(&company.)" j=r &pageof ;
title2 j=1 "Protocol No. %trim(&protocol.)";

%do i=1 %to &n_titles;
    title%eval(&i+2) j=c "%trim(%nrquote(&&t1&i))";
%end;

%do i=1 %to &n_foot;
    footnote%eval(&i) j=1 "%trim(%nrquote(&&ftt&i))";
%end;

footnote%eval(&n_foot+2) j=1 "Program Name: %trim(&pgmpath)%trim(&table2).sas"
    j=r "Creation date: &dt";
footnote%eval(&n_foot+3) j=1 "Datasets used: &getds";

%mend gtitles;
```

CONTACT INFORMATION

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

Peter Gerber
Genzyme Europe B.V.
Gooimeer 10
1411 DD Naarden
Work Phone: Tel: +31 (0)35 699 8660
Evening Phone: Tel: +31 (0) 35 678 1412
Email: Peter.Gerber@genzyme.com

SAS and all other SAS institute Inc. product or service names are registered trademarks Other branded and product names are trademarks of their respective companies.