
Mit MPRINT zum Programmgenerator

KSFE Online

5 April 2022

Hanspeter Schnitzer, Dipl.-Ing.



www.hpschnitzer.de



www.estimondo.com

Gründe für die Erzeugung von makrofreien SAS Programmen

- Auftraggeberwunsch
 - Programme sollen mitgeliefert werden
 - Aber ohne Know-How Transfer
 - Makros kompilieren
 - Wenn es aber gar keine SAS Programme gibt, die man weitergeben kann, weil z.B. mit einem auf Metadaten basierenden System gearbeitet wird?
- Behördenanforderungen
 - Programme sollen mitgeliefert werden
 - Programme müssen frei von jeglichem Makrocode sein

MPRINT Macro System Option

Specifies whether SAS statements generated by **macro execution** are traced for **debugging**.

Syntax

MPRINT | NOMPRINT

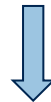
Details

The MPRINT option displays the text generated by macro execution. **Each SAS statement begins a new line**. Each line of MPRINT output is identified with the prefix `MPRINT(macro-name)` to identify the macro that generates the statement. Tokens that are separated by **multiple spaces** are printed with one intervening space.

You can direct MPRINT output to an **external file** by also using the MFILE option and assigning the fileref MPrint to that file. For more information, see [MFILE Macro System Option](#).

```
%macro test;  
  data demo;  
    set sashelp.demographics;  
    rename name      = country  
           isoname   = isocountry  
           pop       = population  
           popurban  = popurbanpct  
           gni       = income  
  ;  
run;  
  
proc sort data=demo;  
  by country;  
run;  
%mend;
```

```
options mprint;  
%test;  
options nomprint;
```



Logfenster:

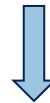
```
MPRINT(TEST):  data demo;  
MPRINT(TEST):  set sashelp.demographics;  
MPRINT(TEST):  rename name = country isoname = isocountry pop = population popurban =  
popurbanpct gni = income ;  
MPRINT(TEST):  run;
```

```
NOTE: There were 197 observations read from the data set SASHELP.DEMOGRAPHICS.  
NOTE: The data set WORK.DEMO has 197 observations and 18 variables.  
NOTE: DATA statement used (Total process time):  
      real time           0.03 seconds  
      cpu time            0.01 seconds
```

```
MPRINT(TEST):  proc sort data=demo;  
MPRINT(TEST):  by country;  
MPRINT(TEST):  run;
```

- Wie erwartet gehen Formatierungen verloren, alles ist linksbündig
- Die Übersicht lässt insgesamt zu wünschen übrig
- Längere Zeilen werden willkürlich umgebrochen
- Programmzeilen wechseln sich mit Notes etc. ab
- Bei komplexeren Programmstrukturen stoßen wir an Grenzen

```
filename mprint "c:\path\name.txt" lrecl=2048;  
options mfile mprint;  
%test;  
options nomfile nomprint;
```



```
Programmdatei name.txt:  
data demo;  
set sashelp.demographics;  
rename name = country isoname = isocountry pop = population ...  
run;  
proc sort data=demo;  
by country;  
run;
```

- Das Ergebnis ist deutlich verbessert
- Störende Informationen sind verschwunden
- Aber weiterhin gehen Formatierungen verloren, alles ist linksbündig
- Jeder SAS Befehl (hier RENAME) wird in einer Zeile dargestellt, also bis zum nächsten Semikolon (;)
- Leerzeichen werden komprimiert
- Zu weiteren Problemen werden wir später kommen



Es drängt sich auf, in einem nachfolgenden Schritt das Resultat von MPRINT aufzubessern

Doch zuvor schnell noch diese zusätzliche Information


Wie wir gesehen haben, wird der SAS Code lediglich bei der Ausführung von Makros ausgegeben. Wollen wir ganze SAS Programme inkl. der dort enthaltenen Makroausführungen ausgeben, können wir uns mit folgender Vorgehensweise behelfen.

```
%macro RunProc;
  filename mprint "c:\path\name.txt" lrecl=2048;
  options mfile mprint;
  %include "c:\path\name.sas"; /* original SAS program */
%mend;
%RunProc;
options nomfile nomprint;
```

Programmdatei vor Einrückung:

```
data demo;
set sashelp.demographics;
rename name = country isoname = isocountry pop = population ...
run;
proc sort data=demo;
by country;
run;
```

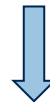
```
data _null_;
length InLine $2048;
retain Indent 0;
infile "c:\path\name.txt" DELIMITER='09'x DSD lrecl=2048; /* SAS code from MPRINT */
file "c:\path\name.sas" lrecl=2048; /* SAS code with changes */
input InLine $&;
if (upcase(substrn(InLine, 1, 4)) eq 'QUIT') or
    (upcase(substrn(InLine, 1, 3)) eq 'RUN') then do;
    Indent = 0;
    put InLine; put;
end;
else do;
    if Indent eq 1 then put ' ' InLine;
        else put InLine;
    if upcase(substrn(InLine, 1, 4)) IN ('DATA' 'PROC') then Indent = 1;
end;
run;
```



Programmdatei nach Einrückung :

```
data demo;  
  set sashelp.demographics;  
  rename name = country isoname = isocountry pop = population ...  
run;  
  
proc sort data=demo;  
  by country;  
run;
```

- Die Einrückung erhöht die Lesbarkeit deutlich
- Weitere Einrückungen auch in mehreren Ebenen sind denkbar
- Aber weiterhin wird jeder SAS Befehl in einer Zeile dargestellt




Wir müssen also weitere Schritte ergänzen

RENAME (1 von 2)

RENAME Statement vorher:

```
...  
rename name = country isoname = isocountry pop = population ...  
...
```

```
data _null_;  
length InLine word lword $2048;  
retain RENAME 0;  
  infile "c:\path\name.txt" DELIMITER='09'x DSD lrecl=2048; /* SAS code from MPRINT */  
  file "c:\path\name.sas" lrecl=2048; /* SAS code with changes */  
  input InLine $&;  
  if upcase(substrn(InLine, 1, 6)) eq ('RENAME') then RENAME = 1;  
  else put InLine;  
  
  if RENAME eq 1 then do;  
    lword = scan(InLine, 1, ' ;', 'Q'); put ' ' lword;  
    count = 1; cnt = 0;  
    do until(word=' ');  
      count+1; cnt+1;  
      word = scan(InLine, count, ' ;', 'Q');  
      if cnt eq 1 then lword = word;  
      else if cnt eq 2 then lword = catx(' ', lword, word);  
      else do;  
        lword = catx(' ', lword, word); put ' ' lword;  
        cnt = 0;  
      end;  
    end;  
    put ' ;'; RENAME = 0;  
  end;  
run;
```



RENAME Statement nachher:

```
...
  rename
    name = country
    isoname = isocountry
    pop = population
    popurban = popurbanpct
    gni = income
  ;
...
```

Ursprüngliche Variante:

```
...
  rename name      = country
         isoname   = isocountry
         pop       = population
         popurban  = popurbanpct
         gni       = income
  ;
...
```

- Die Aufteilung des RENAME Statements in mehrere Zeilen verbessert die Übersichtlichkeit nochmal deutlich
- Für jeden relevanten SAS Befehl können Lösungen gefunden werden
- z.B. bei PROC FORMAT oder bei PROC SQL

Ursprüngliche Programmzeilen:

```
proc format library=work;
value   $LBTEST   "HCG"   = "Choriogonadotropin Beta"
          "LBALL" = "Laboratory Data"
          "ALB"   = "Albumin (g/L)"
          "ALP"   = "Alkaline Phosphatase (U/L)";
run;
```



Nach MPRINT in Programmdatei:

```
proc format library=work;
value $LBTEST "HCG" = "Choriogonadotropin Beta" "LBALL" = "Laboratory Data" "ALB" = "Albumin ..."
run;
```

PROC FORMAT (2 von 2)

```
data _null_;
length InLine word lword $2048;
retain FMT 0;
infile "c:\path\name.txt" DELIMITER='09'x DSD lrecl=2048; /* SAS code from MPRINT */
file "c:\path\name.sas" lrecl=2048; /* SAS code with changes */
input InLine $&;
if upcase(substrn(InLine, 1, 11)) eq ('PROC FORMAT') then do;
    FMT = 1; put InLine;
end;
else if upcase(substrn(InLine, 1, 3)) eq ('RUN') then do;
    FMT = 0; put InLine;
end;
else if FMT eq 1 then do;
    lword = catx(' ', scan(InLine, 1, ' ', 'Q'), scan(InLine, 2, ' ', 'Q'));
    count = 2; cnt = 0; put ' ' lword;
    do until(word=' ');
        count+1; cnt+1; word = scan(InLine, count, ' ', 'Q');
        if cnt eq 1 then lword = word;
        else if cnt eq 2 then lword = catx(' ', lword, word);
        else do;
            lword = catx(' ', lword, word); cnt = 0; put ' ' lword;
        end;
    end;
end;
else do;
    put InLine;
end;
run;
```



Programmdatei nachher:

```
proc format library=work;
value $LBTEST
    "HCG" = "Choriogonadotropin Beta"
    "LBALL" = "Laboratory Data"
    "ALB" = "Albumin (g/L)"
    "ALP" = "Alkaline Phosphatase (U/L)";
run;
```

Ursprüngliche Programmzeilen:

```
proc sql;
  create table height_sum as
  select sex,
         age,
         count(*) as count,
         avg(height) as avg_height,
         min(height) as min_height,
         max(height) as max_height,
         std(height) as std_height
  from sashelp.class
  group by sex, age
         having avg(height) > 60;
quit;
```



Nach MPRINT in Programmdatei:

```
proc sql;
create table height_sum as select sex, age, count(*) as count, avg(height) as avg_height, ...
quit;
```

Lösungsvorschläge sind jederzeit willkommen

Über die bisher beschriebenen Formatierungsproblematiken hinaus gibt es weitere Dinge zu beachten.

SAS bietet drei verschiedene Methoden, um Kommentarzeilen in SAS Programme bzw. SAS Makros einzufügen:

```
*Comment1;
```

```
/*Comment2*/
```

```
%*Comment3;
```

Die Kommentare `Comment2` und `Comment3` werden bei MPRINT nicht mit ausgegeben. Sie eignen sich also für die Kommentierung der Makros, die wir schreiben.

Kommentar `Comment1` wird mit in die Logdatei geschrieben. Mit diesem Kommentartyp können wir also unser Programm kommentieren, welches wir erzeugen wollen.

Aber zur Erinnerung:

the prefix MPRINT (*macro-name*) to identify the macro that
multiple spaces are printed with one intervening space.

Ursprünglicher Kommentar:

```
*****;  
** Module Name:      TestMacro;  
** Type / Category:  Macro / Test;  
** Source File:      ..\Macros\Test.sas;  
** Software(Dev/Sys): SAS Version 9.4 / Windows 10;  
*****;
```



Nach MPRINT in Programmdatei:

```
*****;  
** Module Name: TestMacro;  
** Type / Category: Macro / Test;  
** Source File: ..\Macros\Test.sas;  
** Software(Dev/Sys): SAS Version 9.4 / Windows 10;  
*****;
```



Angepasster ursprünglicher Kommentar:

```
*****;  
** Module Name:_____TestMacro;  
** Type / Category:___Macro / Test;  
** Source File:_____..\Macros\Test.sas;  
** Software(Dev/Sys): SAS Version 9.4 / Windows 10;  
*****;
```

```
data _null_;  
length InLine $2048;  
infile "c:\path\name.txt" DELIMITER='09'x DSD lrecl=2048; /* SAS code from MPRINT */  
file "c:\path\name.sas" lrecl=2048; /* SAS code with changes */  
input InLine $&;  
InLine = translate(InLine, " ", "_");  
put InLine;  
run;
```



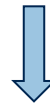
Nach MPRINT in Programmdatei:

```
*****;  
** Module Name:          TestMacro;  
** Type / Category:     Macro / Test;  
** Source File:         ..\Macros\Test.sas;  
** Software(Dev/Sys):  SAS Version 9.4 / Windows 10;  
*****;
```

Ursprüngliche Programmzeilen:

```
proc sql noprint;
  select min(height) into :MinHeight trimmed
  from sashelp.class;
quit;

data test;
  set sashelp.class;
  MinHeight = &MinHeight.;
run;
```



Nach MPRINT in Programmdatei:

```
proc sql noprint;
  select min(height) into :MinHeight trimmed from sashelp.class;
quit;
data test;
  set sashelp.class;
  MinHeight = 51.3;
run;
```

- Hier können wir uns schwerlich mit einem nachgelagerten Programm behelfen
- Wir benötigen eine andere Lösung

Makrovariablen (2 von 4)

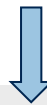
Angepasste ursprüngliche Programmzeilen:

```
proc sql noprint;  
  select min(height) into :MinHeight trimmed  
  from sashelp.class;  
quit;
```

```
option nomprint;
```

```
data test;  
  set sashelp.class;  
  MinHeight = &MinHeight. ;  
run;
```

```
%Print2Prog(c:\path\name.txt, data test);  
%Print2Prog(c:\path\name.txt, set sashelp.class);  
%Print2Prog(c:\path\name.txt, %nrstr(MinHeight = &MinHeight.));  
%Print2Prog(c:\path\name.txt, run);  
option mprint;
```



Nach MPRINT in Programmdatei:

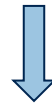
```
proc sql noprint;  
select min(height) into :MinHeight trimmed from sashelp.class;  
quit;
```

```
option nomprint;
```

```
data test;  
set sashelp.class;  
MinHeight = &MinHeight. ;  
run;
```

Ursprüngliche Programmzeilen:

```
%test2(file = c:\path\name.pdf);  
...  
ods pdf file = "&file.";  
proc report data = sashelp.class nowd headline headskip missing center ;  
  column name age;  
  define name / display;  
  define age / display;  
run;  
ods pdf close;
```



Nach MPRINT in Programmdatei:

```
ods pdf file = "c:\path\name.pdf";  
proc report data = sashelp.class nowd headline headskip missing center ;  
column name age;  
define name / display;  
define age / display;  
run;  
ods pdf close;
```

Nach MPRINT in Programmdatei:

```
ods pdf file = "c:\path\name.pdf";  
proc report data = sashelp.class nowd headline headskip missing center ;  
column name age;  
define name / display;  
define age / display;  
run;  
ods pdf close;
```

```
data _null_;  
length InLine $2048; retain Indent 0;  
infile "c:\path\name.txt" DELIMITER='09'x DSD lrecl=2048; /* SAS code from MPRINT */  
file "c:\path\name.sas" lrecl=2048; /* SAS code with changes */  
input InLine $&;  
InLine = tranwrd(InLine, "&file.", %nrstr("&file."));  
put InLine;  
run;
```



Programmdatei nachher:

```
ods pdf file = "&file.";  
proc report data = sashelp.class nowd headline headskip missing center ;  
column name age;  
define name / display;  
define age / display;  
run;  
ods pdf close;
```

```
%macro Print2Prog(iFile, iText);  
  data _null_;  
  length LineIn $2048;  
  file "&iFile." recfm=N mod;  
  LineIn = "&iText."; put LineIn '0d0a'x;  
run;  
%mend;
```


Gründe für die Erzeugung von SAS Programmen als Output von ausgeführten SAS Makros

- Auftraggeberwunsch
- Behördenanforderung
- Programmreview bzw. -debugging

Erzeugung von SAS Programmen als Output von ausgeführten SAS Makros

- MPRINT verwenden
- MPRINT ggf. zwischendurch ausschalten
- Makro ggf. in spezifischen Fällen anpassen (z.B. %Print2Prog)
- Überarbeitung der vom SAS Makro erzeugten SAS Programme

Fragen?



Schnitzer
Hanspeter Schnitzer Dipl.-Ing.
SAS Programming

www.hpschnitzer.de mail@hpschnitzer.de



estimondo 
The world of estimands.

Hanspeter Schnitzer
Co-founder & Senior Principal Statistical Programmer
hanspeter.schnitzer@estimondo.com
+49 177 745 9759

www.estimondo.com
Estimondo GmbH • Hintergasse 22 • D-61231 Bad Nauheim • Germany