ABSTRACT.

Techniques for improving printouts of SAS data sets using macros are discussed in this report.

Data sets with large fields such as a 200 character field or an array of 200 character fields, "wrap around" when using PROC PRINT, making reading of the data very difficult. This paper presents a macro which greatly simplifies the printing of data in column format when used with PUT statements in a DATA step.

DATA SET EXAMPLE

The macro presented in this paper can create from 1 to 9 columns, each containing 1 to 999 variables. The examples included in this report are based on a SAS data set that contains 10 variables, 2 of which are character variables of length 200 defining a problem for a problem report, and 3 of which are character variables of length 200 documenting the solution to the problem. The following is the PROC CONTENTS of this data set.

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>06/23/84</td>
<td>884</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>07/11/84</td>
<td>844</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>07/11/84</td>
<td>844</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>11/16/83</td>
<td>543</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>

PROC PRINT

Using PROC PRINT to display this data set is a natural first choice. As can be seen, this output is for all practical purposes, unreadable. Even though this printout has been limited to 48 characters in width for this report, a 132 width printout is really no better. The 200 character variables have been truncated and the variables are grouped on the page in such a manner that reading of a single observation is very difficult.

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>DATE</th>
<th>DEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VAIR</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>VM USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>GENERAL USERS</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>USER X</td>
<td>299</td>
<td>1</td>
</tr>
</tbody>
</table>
A very acceptable solution to the readability problem is to use PROC FSEDIT or PROC FSBROWSE. These PROCs are contained in SAS/FSP. Setting the OPTIONS statement set to NOFS results in the screens being printed to the SAS LISTING file. This technique allows a screen to be defined which customizes printouts in a very acceptable manner. A screen may be defined with little effort, and any number of screens may be defined. An example of a screen layout follows. When the data is presented in this format, a completely readable format is presented.

Problem Tracking System

Problem reported by: VAIR
ZIP 884 001 3

Problem: The audience is falling asleep!!!!!!!

Solution: 1) Coffee Break
2) NODZ
3) Hard Chairs
4) Loud Speaker
5) Good Jokes
6) Bad Jokes
7) Discuss Salaries
8) Pass out popcorn
9) Let them sleep, they may of had a rough night.

SAS MACRO TO PRINT AN ARRAY OF VARIABLES
IN COLUMN FORMAT

The use of FSEDIT to format printouts does have some limitations.

1. Not all installations have purchased SAS/FSP and therefore may not have FSEDIT available for use.
2. FSEDIT produces one observation per page.
3. It is difficult to produce screens wider than 80 characters or longer than 24 or 32 lines. The physical size of a screen is limited by the size of the terminal in use.
4. The output produced by FSEDIT is one screen per page, which often results in an observation being split over several pages, especially when the number of comments or text areas are large. The resulting LISTING file requires editing to resolve this.
5. The data is not compressed on the printout. Unused lines in the variables are printed with the pad characters specified in FSEDIT (usually the underscore).

A macro has been written by the author to provide enhanced printout in several forms. This macro groups arrays of character fields into columns for inclusion in reports. The number of groups of data defining items to be included in columns such as PROBLEM, SOLUTION, STATUS, COMMENTS, etc. may range from one to nine. Up to 999 character variables may comprise each group. The macro variable NWORDS defines the maximum number of words per column, and the macro variable WORDCHR defines the maximum length for the words. Again for this report, an example of a data set containing 2 groups (PROBLEM and SOLUTION) is used. The first group (PROBLEM and PROBLEM2) consists of two 200 character variables, and the second group (SOLUTION, SOLUTN1, and SOLUTN2) contains three 200 character variables.

An ARRAYS macro is called to define the necessary arrays.

A TEXT macro has one parameter which indicates the column number. This macro parses the words in the variables.

%text(1)

A PRINT macro may be called with or without any parameters. This macro actually builds the printout. A variable named LINECNT is generated to indicate the number of lines printed. If no parameter is entered, all columns are printed. If COL= is specified, then only that column is printed. If PRINT=NOPRINT is specified, then no printout occurs, but the variable LINECNT is generated to indicate the number of lines that would have been printed.

%print
%print(col=1) -or-
%print(col=1,print=noprint)

The macro BUILD builds the following code for two columns.

%arrays
%text(1)
%text(2)

The COLUMN macro consists of a TEXT macro which builds each column of text and a PRINT macro which prints each column of text. An ARRAYS macro is also called to define the necessary arrays. The macro COLUMN builds the following code for two columns. This macro may also be called with the PRINT=NOPRINT option to return only the line count. The second time it is called it must be called with the option PRINT=PRINT.

%arrays
%text(1)
%text(2)
%print

COLUMN, A SAS MACRO

******************************************************************************
 * ROUTINE TO enhance SAS printouts
 * GARY VAIR
 * INFORMATION CENTER
 * IBM BOULDER
 * 4/11/84
******************************************************************************

OPTIONS MPRINT:
*OPTIONS MPRINT SYMBOLGEN;
*OPTIONS MACROGEN SYMBOLGEN MLOGIC;
*OPTIONS NOSOURCE;
OPTIONS QQUOTE;
******************************************************************************

* Build a macro for building the necessary *
* variables and arrays

757
%let nwords = 150;
/* maximum number of words per column */
%let wordchr = 20;
/* maximum number of letters per word */
%macro arrays;
LENGTH DUMMY $200;
/* blank characters for printout*/
LENGTH WORD $&wordchr;
%do i = 1 %to &nocols;
%if &i eq 1 %then %do;
%let leneol = LENGTH:
%let eolarray = ARRAY COL;
%let prtprint = ARRAY PRINTPOS;
%let j = 1 %to &wordchr;
%let txtarr&j = ARRAY TEXT&j (T&j);
%end;
%end;
%LET LEN COL = &LENCOL COL&I $ &&LENGTH&I;
%let col array = &colarray COL&i;
%let prtparray = &prtparray PRPOS&i;
%do j = 1 %to &nvar;
%let txtarr&i = &&txtarr&j &&var&i&j;
%end;
%end;
%macro text(a);
I&a=l; DO OVER TEXT&a; K=I' DO UNTIL (WORD = ' '); WORD = SCAN (TEXT&a,K,' '); K=K+I' IF WORD NE ' , THEN
DO; COLTEXT&a.=WORD; I&a = I&a + I; END' END ' END;
NO_WORD&a = I&a;
END;
%end;
%macro columns(print=);
%if %upcase(&print) ne PRINT %then %arrays;
%do a = 1 %to &nocols;
%text(&a)
%end; %print(print=&print);
%mend columns;

%if &a eq &cstart
%then %let dountil = (I&a GE NO_WORD&a;
%else %let dountil = I&a GE NO_WORD&a;
%end;
DO UNTIL (EXITST1);
%DO UNTIL ( &dountil);
/* the above statement fails by exiting before it
should. If a logic test is made directly in the
DO UNTIL statement and if too many variables
exist. Creating the test variable at the end
of the DO loop fixes the problem */
%do a = &cstart %to &end;
* BUILD COLUMN &a;
COL&a = ' ';
IF I&a = NO_WORD&a THEN
DO;
NO_CHARS = 0;
PASS = 'FIRST';
DO UNTIL (EXITST2);
* DO UNTIL (NO_CHARS = PRTL&i) | (I&a = NO_WORD&a);
/* the above statement fails by exiting before it
should. A logic test is made directly in the
DO UNTIL statement and if too many variables
exist. Creating the test variable at the end
of the DO loop fixes the problem */
NO_CHARS = NO_CHARS + LENGTH (COLTEXT&a);
IF NO_CHARS <= PRTL&i THEN DO;
ADD DOUBLE BLANK AFTER PERIODS;
IF (INDEX (COL&a, '.') EQ 0) THEN DO;
COL&a = TRIM(COL&a); COLTEXT&a = TRIM(COLTEXT&a);
NO_CHARS = NO_CHARS + 2;
END;
ELSE DO;
COL&a = (TRIM(COL&a)); COLTEXT&a = (TRIM(COLTEXT&a));
NO_CHARS = NO_CHARS + 1;
END;
* STRIP OFF LEADING BLANKS ON FIRST PASS;
IF PASS = 'FIRST' THEN
COL&a = STRIP (COL&a); COLTEXT&a = TRIM (COLTEXT&a);
NO_CHARS = NO_CHARS + 1;
END;
EXITST2 = (NO_CHARS = PRTL&i) | (I&a = NO_WORD&a);
END;
%end;
%end;
* BUILD THE PRINTOUT
*----------------------------------------------
%macro print(col=print=);
LINECNT=0;
%if &col eq %then %do;
%let cstart = 1;
%let cend = &nocols;
%end;
%else %do;
%let cstart = &col;
%let cend = &cend;
%end;
%do a = &cstart %to &cend;
I&a = 1;
PRTL&i = &length&a;
PRPOS&i = &prtpos&i;
/* start print column */
USING THE COLUMN MACRO

TWO COLUMNS SIDE BY SIDE NEXT TO OTHER VARIABLES

%let nocols = 2; /* number of columns */
%let prtpos1 = 55; /* START PRINT COLUMN a */
%let prtpos2 = 45; /* START PRINT COLUMN b */
%let length1 = 37; /* LENGTH OF PRINT COLUMN a */
%let length2 = 37; /* LENGTH OF PRINT COLUMN b */
%let nvar = 3; /* maximum number of variables to print in */
%let var1 = PROBLEM; /* variable name to print in column a first */
%let var2 = DUMMY; /* unused variables MUST be set to DUMMY */
%let var3 = SOLUTION; /* variable name to print in column b first */
%let var4 = SOLUTN1; /* variable name to print in column b second */
%let var5 = SOLUTN2; /* variable name to print in column b third */

DATA _NULL;
SET DATA.SASPRINT; FILE PRINT HEADER = H LINESLEFT = L;
%build
%print(col=1,print=noprint)
IF LINECNT + 3 GT L THEN PUT _PAGE_ @;
%print(col=2,print=noprint)
/* will it fit along with spaces and headings? */
LINEUSED = LINEUSED + LINECNT;
IF LINEUSED + 2 GT L THEN PUT _PAGE_ @;
PUT @1 NAME @22 DEPT @26 BUILDING @30 FLOOR @38 DATE @;
PUT @55 'PROBLEM'@;
%print(col=1)
PUT // @55 'SOLUTION'@;
%print(col=2)
PUT //;
RETURN;
H: PUT @1 'NAME' @22 'DEPT/BUILDING' @38 'DATE' @;
RETURN;

Gary G. Vair
Dept 844/001-3
Information Center
IPD Boulder
IBM Corporation
P.O. Box 1900
Boulder, Colorado 80301-9191
(303) 447-3122
SAMPLE PRINTOUTS

NAME | DEPT/BUILDING | DATE | PROBLEM |
--- | --- | --- | --- |
VAIR | 884 001 3 | 06/23/84 | The audience is falling asleep!!!!! |

PROBLEM: The audience is falling asleep!!!!!

SOLUTION: 1) Coffee Break 2) NODIZ 3) Hard Chairs 4) Loud Speaker 5) Good Jokes 6) Bad Jokes 7) Discuss Salaries 8) Pass out popcorn 9) Let them sleep, they may of had a rough night.

VM USERS | 844 001 3 | 07/11/84 | Teach a new user how to do their work using the VM/SP (Virtual Machine/System Product) system and a video display terminal. We need to present the material in a way that will interest the user, enable the user to be successful, and sometimes even make the user smile. |

PROBLEM: Teach a new user how to do their work using the VM/SP (Virtual Machine/System Product) system and a video display terminal. We need to present the material in a way that will interest the user, enable the user to be successful, and sometimes even make the user smile.

SOLUTION: Instruct the user on the methods of getting started, editing files, managing their files, printing files, communicating with other people who use computers, asking questions, using Script to format files, and creating and using EXECs. An introduction to Data Processing using SAS/FSP is the next logical follow-on course.

GENERAL USERS | 844 001 3 | 07/11/84 | How do I as a user control my data, process the data, summarize the data, and report the data? |

PROBLEM: How do I as a user control my data, process the data, summarize the data, and report the data?

SOLUTION: SAS is a computer software system for data analysis. Since its beginnings in 1966, the goal of SAS has been to provide data analysts one system to meet all their computing needs. When your computing needs are met, you are free to concentrate on results rather than on the mechanics of getting them. Instead of learning programming languages, several statistical packages, and utility programs, you only need to learn SAS.

USER X | 543 299 1 | 11/16/83 | How do I create letters and documents? |

PROBLEM: How do I create letters and documents?

SOLUTION: SCRIPT/VS formats text for printing on terminals, impact printers, or nonimpact printers. SCRIPT/VS provides flexible composition for printing on a computer printer as an alternative to independent typesetting machines or sending typesetting jobs to an outside vendor. SCRIPT/VS reads input data containing text and control information, formats the data into pages, and produces formatted output on a suitable output device.
NAME: VAIR
DEPT/BUILDING: 884 001 3
DATE: 06/23/84

PROBLEM: The audience is falling asleep!!!!!
SOLUTION: 1) Coffee Break 2) NODOZ 3) Hard Chairs 4) Loud Speaker 5) Good Jokes 6) Bad Jokes 7) Discuss Salaries 8) Pass out popcorn 9) Let them sleep, they may of had a rough night.

VM USERS

DATE: 07/11/84

PROBLEM: Teach a new user how to do their work using the VM/SP (Virtual Machine/System Product) system and a video display terminal. We need to present the material in a way that will interest the user, enable the user to be successful, and sometimes even make the user smile.
SOLUTION: Instruct the user on the methods of getting started, editing files, managing their files, printing files, communicating with other people who use computers, asking questions, using Script to format files, and creating and using EXECs. An introduction to Data Processing Using SAS/FSP is the next logical follow-on course.

GENERAL USERS

DATE: 07/11/84

PROBLEM: How do I as a user control my data, process the data, summarize the data, and report the data?
SOLUTION: SAS is a computer software system for data analysis. Since its beginnings in 1966, the goal of SAS has been to provide data analysts one system to meet all their computing needs. When your computing needs are met, you are free to concentrate on results rather than on the mechanics of getting them. Instead of learning programming languages, several statistical packages, and utility programs, you only need to learn SAS.

SAMPLE ENHANCEMENT NUMBER 3

NAME: VAIR
DEPT/BUILDING: 884 001 3
DATE: 06/23/84

PROBLEM: The audience is falling asleep!!!!!
SOLUTION: 1) Coffee Break 2) NODOZ 3) Hard Chairs 4) Loud Speaker 5) Good Jokes 6) Bad Jokes 7) Discuss Salaries 8) Pass out popcorn 9) Let them sleep, they may of had a rough night.

VM USERS

DATE: 07/11/84

PROBLEM: Teach a new user how to do their work using the VM/SP (Virtual Machine/System Product) system and a video display terminal. We need to present the material in a way that will interest the user, enable the user to be successful, and sometimes even make the user smile.
SOLUTION: Instruct the user on the methods of getting started, editing files, managing their files, printing files, communicating with other people who use computers, asking questions, using Script to format files, and creating and using EXECs. An introduction to Data Processing Using SAS/FSP is the next logical follow-on course.

GENERAL USERS

DATE: 07/11/84

PROBLEM: How do I as a user control my data, process the data, summarize the data, and report the data?
SOLUTION: SAS is a computer software system for data analysis. Since its beginnings in 1966, the goal of SAS has been to provide data analysts one system to meet all their computing needs. When your computing needs are met, you are free to concentrate on results rather than on the mechanics of getting them. Instead of learning programming languages, several statistical packages, and utility programs, you only need to learn SAS.

SAMPLE ENHANCEMENT NUMBER 4

NAME: USER X
DEPT/BUILDING: 543 299 1
DATE: 11/16/83

PROBLEM: How do I create letters and documents?
SOLUTION: SCRIPT/VS formats text for printing on terminals, impact printers, or nonimpact printers. SCRIPT/VS provides flexible composition for printing on a computer printer as an alternative to independent typesetting machines or sending typesetting jobs to an outside vendor. SCRIPT/VS reads input data containing text and control information, formats the data into pages, and produces formatted output on a suitable output device.