When displaying data for quick review, the most commonly used method is the PRINT procedure. Unfortunately, when observations are too long to be displayed on a single line, PROC PRINT wraps the output lines in a manner such that the variables for one observation seem scattered throughout a page.

A more useful report is one in which each observation appears as a rectangular block. In this sense, the report is made up of "objects"—each block of data, an observation, is an object. We could try to produce such a layout using PROC REPORT or a DATA step, but the programming required is more than elementary.

A simple approach that can generate the desired report layout is to use the FORMS procedure. Although PROC FORMS is most frequently used to produce special forms such as mailing labels, it can generate other types of formatted output.

The basics of PROC FORMS involve defining the layout of the blocks—the number of blocks across and down a page, the width and length of each block, spacing between blocks, contents of each line, etc. The FORMS procedure is fully documented in the SAS® Procedures Guide: Version 6 manual.

Let's assume that we want a layout with two side-by-side blocks across the page, each block containing five lines. We will specify the options ACROSS=2, BETWEEN=2, LINES=5. The blocks are each 39 columns wide and are separated by 2 columns. The number of blocks per page automatically will be determined by the PAGESIZE system option. The LINESIZE system option is assumed to have the value 80.

To display selected variables for each observation, we use LINE statements. Since we have 5 lines in each block, we can specify up to 5 LINE statements. Line numbers can be skipped to produce blank lines as can be done by skipping title line numbers when defining TITLE statements. In our example, we will display the value for one variable per line.

Unlike other reporting procedures, PROC FORMS cannot display variable names (or labels), although values can be formatted. We could display our variables without names, but it might be difficult to identify specific variables. Therefore, we must create additional variables that contain the variable names and/or labels.

The code in Figure 1 creates name variables for all character and numeric variables to be displayed. The VNAME routine obtains the name of the variable represented by the first parameter, assigning that name as the value of the character variable represents by the second parameter. In this example, CLABEL1 is assigned the value "COMPANY", CLABEL2 is set to "PERSON", etc. The "::" is attached to each name purely for aesthetics.

```sas
DATA WITHNAME;
SET ORIGINAL;
ARRAY CVARS COMPANY PERSON ITEM;
ARRAY CLABEL $ 9 CLABEL1-CLABEL3;
DO OVER CVARS;
   CALL VNAME (CVARS, CLABEL);
   CLABEL=TRIM (CLABEL) "::";
END;
ARRAY NVARS DATE PRICE;
ARRAY NLABEL $ 9 NLABEL1-MLABEL2;
DO OVER NVARS;
   CALL VNAME (NVARS, NLABEL);
   NLABEL=TRIM (MLABEL) "::";
END;
RUN;
```

Figure 1

If variable labels are desired, the LABEL routine would be used in place of the VNAME routine. Since variable labels can be much longer than names, some care must be taken when designing the output blocks, possibly restricting the report to blocks that span the entire width of the page.

In the PROC FORMS step (Figure 2), we display each name variable immediately preceding its corresponding value. The PACK option is added to each LINE statement to remove excess blanks between the name and the value.

```sas
PROC FORMS DATA=WPHDMB
   ACROSS=2 BETWEEN=2 LINES=5 WIDTH=39;
LINE 1 CLABEL1 COMPANY / PACK;
LINE 2 CLABLE2 PERSON / PACK;
LINE 3 NLABEL1 DATE / PACK;
LINE 4 CLABLE3 ITEM / PACK;
LINE 5 NLABEL2 PRICE / PACK;
FORMAT PRICE DOLLAR12.2 DATE DDMMYY8.5;
TITLE 'DUMP OF DATA SET WORK.WITHNAME';
RUN;
```

Figure 2

The resulting output is shown in Figure 3A.

The same data displayed with variable labels obtained with the LABEL routine are shown in Figure 3B. Only one block is printed across the page as indicated earlier.

This general technique can be used to produce a multitude of layouts, for instance, placing multiple variables on a single LINE statement or creating specialized display lines by creating custom variables through DATA step character string manipulation functions.
To contact the author:

Warren Repole Jr.
CSC Professional Services Group
1301 Piccard Drive, 2nd floor
Rockville, MD 20850
voice: 301/258-5300
fax: 301/258-6897

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.