

Dynamically Build a PROC FORMAT for Subsetting Large Datasets

Warren E. Stinson
 Director, Analytical Processes
 Dun & Bradstreet, Inc.

How do you pass through a vary large master file extracting specific records by account number without coding the Boolean logic or doing a SAS merge on the file?

This paper addresses a method to dynamically build a PROC FORMAT and utilize that format with the DATA step to extract the requested records. This method for subsetting the file can be used independently of changes to the master list of account numbers being constantly changed by the user.

To dynamically build the PROC FORMAT, the following steps need to take place. The first DATA step will read the account number file and write our a SAS file. (Note: Most files used in my facility are "raw" data files.) A PROC SORT is then built to sort the records and remove duplicate account numbers. The second DATA step reads the sorted file and writes the file to a temporary data set in the form of a PROC FORMAT value statement. The PROC FORMAT is then called into the job stream using the %INCLUDE statement. The format is referenced further down the job stream in another DATA step in conjunction with a PUT statement.

EXAMPLE:

```
DATA TRANS;
  INFILE TRANS;
  INPUT @1 ACCOUNT 9.;

PROC SORT DATA=TRANS NODUPS;
  BY ACCOUNT;

DATA _NULL_;
  SET TRANS END=EOF;
  FILE TEMP;
  IF _N_ = 1 THEN PUT @1 'VALUE MATCH';
  IF EOF = 0 THEN PUT @1 ACCOUNT Z9. ",";
  ELSE
  PUT @1 ACCOUNT Z9. "= '1'" /
    @1 "OTHER = '0'";

PROC FORMAT; %INCLUDE TEMP;

DATA _NULL_;
  INFILE MASTER;
  INPUT @1 ACCOUNT 9.
    @15 FIELDS $CHAR19.;
  FILE OUTFILE;
  IF PUT(ACCOUNT,ACCOUNT.) = '1' THEN
  PUT @1 ACCOUNT Z9.
    @10 FIELDS $CHAR19.;
  ELSE RETURN;
```