

ANOTHER SORTING TECHNIQUE FOR LARGE, TROUBLESOME DATASETS

David Hopkins, Economic Research Service, USDA, Washington, DC

This is a nice sorting technique for large files, if you have to physically sort the files (no indices) and if you are having problems, like running out of work space or disk space. This usually occurs because you have a large number of observations or large number of variables, or both. Sorting problems usually occur when you run out of work space or disk space because of the large number of variables or observations that need to be moved around. I first saw this type of sorting on the SAS-L discussion group. I have rewritten the sort so it would apply to a larger group of users.

In order to use this technique, your large SAS file must be stored as a disk or random access file. This technique will not work on tape files. And, you have to have some free disk space to copy the new, sorted file without destroying the original file. It works on all platforms, DOS, OS/2, CMS and MVS etc.

Basically, this technique will be sorting a much smaller file instead of the larger file. The smaller, sorted file will be used to pull the larger file together into a sorted form.

First, identify your sort variable(s) in your large file. You are going to make a smaller file of just your sort variable(s). For example, let's say you have SAS file called LARGE. It has about 10,000 observations and 200 variables. You want to sort the file on one of the variables, ID. First, you will create a smaller file, INDXFIL. In the file, INDXFIL, you will have two variables, ID

and RECNUM. RECNUM will take on the value of observation numbers. We will use it later to grab the records of the larger file in order.

```
data indxfil ;
  set large (keep = id);
  recnum = _n_ ;
run;
```

Next, sort the smaller file. This should be easier than sorting the large file.

```
proc sort data = indxfil;
  by id;
run;
```

Now use the sorted, smaller file to put the larger file in order, like this:

```
data sortfil;
  set indxfil;
  set large point = recnum ;
  drop recnum;
run;
```

In this final SAS data step, you have two SET statements. The first SET statement brings in a sorted record from the sorted, smaller file INDXFIL. The sorted record has the record number (in RECNUM) of the record in the large file that should be pulled into the new, sorted file. In the second SET statement, the POINT= sub-command of the SET statement will go out and grab the correct record that should be pulled into the new file, in sorted order. For each cycle of the DATA step, the two SET statements work together to pull in the records from the larger, unsorted file, so that the records are now in sorted order. This will go on until the entire large file is brought in, piece

by piece, in sorted order. The POINT= sub-command of the SET statement takes longer to process than the standard, sequential SET without the POINT= . This is one of the tradeoffs of using this technique. PLEASE REMEMBER: this only works with disk files-- NO TAPE FILES.