

How to...

01/06/99

Write/Read UNIX Compressed SAS Data Sets Directly

It is now possible to write UNIX compressed SAS data sets directly to disk. You can then read back into SAS these specially formatted UNIX compressed SAS sets without first uncompressing them. This can save you a lot of disk space.

This handout describes how to write a UNIX compressed SAS data set and then how to read back in the compressed SAS data set that was created.

Note: SAS can only read UNIX SAS compressed data sets that have been created as UNIX compressed files following the instructions in this handout. That is, you cannot compress a SAS data set you have on disk somewhere and then read it in to SAS directly. Nor can you uncompress a SAS data set created following the instructions in this handout and then read it into SAS. These SAS data sets are written in a special format recognizable only when you read them using the statements described herein.

If you are unfamiliar with named pipes, they are described in Pub. #7-5, *Using Compressed Files*. Compressed raw data can be read and written directly and this is documented in Pub. #7-4, *Using SAS on UNIX*. Compressed SAS transport files can also be read and written and this is documented in How-to #28, *How to Read Compressed SAS Transport Files Directly in your SAS Program*.

Writing a UNIX Compressed SAS Data Set

Follow the steps below to write out a UNIX compressed SAS data set:

1. Create a named pipe using the X statement. The X statement allows you to issue UNIX commands within your SAS program. The X statement below is used to issue the UNIX `mknod` command which creates named pipes:

```
x 'mknod piper p';
```

2. Next, you need to issue a SAS LIBNAME statement to assign the libref to the named pipe:

```
libname SQUISH 'piper';
```

Recall that SQUISH in the statement above is the libref. For more information on librefs and the LIBNAME statement, refer to SSCC Pub. #7-4, *Using SAS on UNIX*.

3. Issue a FILENAME statement with the PIPE device to assign a fileref to the UNIX compress command:

```
filename nwrpipe pipe 'compress <piper  
> /aux/j/jhayes/cpsfeb98.Z &';
```

NWRPIPE in the statement above is the fileref. The FILENAME pipe forks another process and executes the UNIX command. The UNIX compress command is reading from the read end of the named pipe, then redirecting the output to the file. Be sure to place this in the background or the job will hang.

4. Next, include the following data step:

```
data _null_;  
  infile nwrpipe;  
run;
```

This data step will reference the filename pipe, then forks the process to execute the UNIX command.

5. Write a data step to write out the data. In the example below, an existing SAS data set (uncompressed) is read in and then written out:

```
libname temp5day  
  "/temp/fivedays/jhayes";  
data squish.cpsfeb98;  
  set temp5day.cpsfeb98;  
  ... more sas statements...  
run;
```

Note the libref used in the left side of the two level name for the SAS data set. It must match the libref you specified in step 2 above.

6. Remove the pipe:

```
x 'rm piper';
```

Following is a complete SAS program for the example described above:

```
x 'mknod piper p';
libname squish 'piper';
filename nwrpipe pipe 'compress <piper
> /aux/j/jhayes/cpsfeb98.Z &';

data _null_;
  infile nwrpipe;
run;

libname temp5day
  "/temp/fivedays/jhayes";
data squish.cpsfeb98;
  set temp5day.cpsfeb98;
  ... more sas statements...
run;

x 'rm piper';
```

Reading a UNIX Compressed SAS Data Set

Note: SAS can only read UNIX SAS compressed data sets directly that have been created following the instructions above.

Follow the steps below to read the UNIX compressed SAS data set created in the previous example:

1. Create a named pipe using the X statement:

```
x 'mknod piper p';
```

2. Issue a SAS LIBNAME statement to assign the libref to the named pipe:

```
libname unsquish 'piper';
```

3. Issue a FILENAME statement with the PIPE device to assign a fileref to the UNIX uncompress command:

```
filename nwrpipe pipe ' uncompress >
piper < /aux/j/jhayes/cpsfeb98.Z &';
```

The FILENAME pipe when referenced, will fork a process and then execute the UNIX command. The UNIX uncompress command gets its input from the compressed file, then writes the output to the write end of the named pipe. Be sure to issue the command in the background or the job will hang.

4. Next, include the following data step:

```
data _null_;
  infile nwrpipe;
run;
```

This data step will reference the filename pipe, then forks the process to execute the UNIX command.

5. Write a data step to read the data:

```
data cpsfeb98;
  set unsquish.cpsfeb98;
run;
```

6. Remove the pipe:

```
x 'rm piper';
```

7. Include the rest of your program to do your analysis.

Following is a complete SAS program for the example described above:

```
x 'mknod piper p';
libname unsquish 'piper';
filename nwrpipe pipe ' uncompress >
piper < /aux/j/jhayes/cpsfeb98.Z &';

data _null_;
  infile nwrpipe;
run;

data cpsfeb98;
  set unsquish.cpsfeb98;
run;

proc reg data=cpsfeb98;
  model learn=age female nonwhite
  usualhrs hsgrad col4grad tenure;
title 'February 1998 CPS - Job Tenure
supplement';
run;
```

A Note about Compression Tools

You are not restricted to the UNIX compress command for doing the compression. GZIP may also

be used and may result in better compression.

