Paper SBC- 130

Using Unnamed Pipes to Automate Loading a SAS Data Warehouse Using Only Base SAS®

Vicky Feldman, Florida Department of Corrections, Tallahassee, FL
Kristine L. Dougherty, Florida Department of Corrections, Tallahassee, FL

ABSTRACT

This paper presents examples that use unnamed pipes to load a SAS data warehouse from zipped mainframe text files with generational data group names. The process uses Winzip® version 10 with Command Line Support Add-On and Base SAS® version 9.1.3 on a Windows® XP Pro platform. The text files are not unzipped during this process. Header records read from the mainframe COBOL extract file are used to create dataset names that include parameter information. Finally, the zipped text files are moved to appropriate file folders on a server, and information on the parameter dates is used to rename the zipped files for archiving purposes. With tight state budgets, this has allowed the agency to improve efficiency with only existing licensed SAS components.

INTRODUCTION

The example presented here is used to process files of zipped text data produced from COBOL generational datasets sent to our file transfer server on a periodic basis. The names of these datasets have the default generational dataset names and are not easily modified. The Command Line Support Add-On feature of WinZip®10 is used in combination with the SAS PIPES command to read the data into a temporary file, eliminating the manual unzipping step that had been used in the past. As each file is received, it is sent to a directory used for holding unprocessed files. On a schedule, the zipped file is processed and read into SAS and used to build or update SAS datasets. After each zipped file is processed, it is sent to the associated import folder for archival purposes and given a filename modification that adds the parameter date. In addition, the file parameter date is used in naming the resulting SAS datasets.

BACKGROUND

The SAS PIPES command, Winzip® Command Line Support Add-On, and Windows Scheduled Tasks are used to automate the process of loading the data warehouse used by our research and data unit. Budget constraints and staff constraints make this automated process a necessity for timely file processing and for faster turnaround on data reports requested by our customers.

Quoting from “SAS Companion for the OS/2 Environment” –

“Unnamed pipes enables you to run a program outside the SAS System and redirect the program's input, output, and error messages to the SAS System. This capability enables you to capture data from a program that is external to the SAS System without creating an intermediate data file. “

Copyright 1999 by SAS® Institute Inc., Cary, NC, USA. All rights reserved.

EXAMPLE 1

Instead of a programmer manually unzipping and moving COBOL transport files every time a file is deposited on the transfer server location, an unnamed pipe is used to redirect the output from a DOS command. The syntax for using unnamed pipes with operating system commands is:

```
filename <fileref> pipe “<command>”;
```

where command is a DOS command. The PIPE option directs the output from the command to a virtual text file, referenced by fileref, that can be accessed in a SAS DATA step.

For example, suppose the directory “c:\” contains a file of type .ZIP. When the following command is executed in the DOS command window:

```
c:\winzip\wzunzip.exe -o -c c:\sesug_import.zip;
```

the output directed to the screen is:

```
X101 A
X102 B
X103 C
X104 D
```

By using the FILENAME command with the PIPE option and the WZUNZIP.EXE command, this output can be directed to a virtual file and read into a SAS dataset. Once the information is in a SAS data set, it can be processed using SAS commands.

```
filename rspgm pipe '"c:\winzip\wzunzip.exe" -o -c c:\sesug_import.zip';

data temp;
infile rspgm firstobs=9 missover lrecl=50 length=offs;

input
loc 2-4
dorm $14;

label
loc = "FACILITY ID"
dorm = "DORM ID";

run;
```

EXAMPLE 2

The use of pipes can be combined with the SAS macro variable language to pull a parameter date from the header row of an imported file.

An example of the header record with the parameter date follows:

```
SESUG.TEXT 20080509 17:53:24
X101 A
X102 B
X103 C
X104 D
```

If the SAS macro variable named “dtepar” is defined to store the parameter date (parmdte) from the zipped file, the above SAS code could be modified to store this variable for use later in the SAS program:

```
data temp;
infile rspgm firstobs=9 missover lrecl=50 length=offs;
if _n_ = 1 then do;

input @40 parmdte yymmdd8.;
dte2=put(parmdte, yymmdd6.);
call symput('dtepar',dte2);
drop parmdte dte2;
delete;
end;
```

```
else input
loc 2-4
dorm $14;
label
loc = "FACILITY ID"
dorm = "DORM ID";

*see detailed note following code;
if loc  = . then delete;      *removes header record;
if _n_  = 1 then delete;      *removes blank lines;
if offs = 0 then delete;      *removes winzip added line;

run;
```

*Note - In addition to header records created by the COBOL program, we have found that when reading the zipped files, additional header, footer or blank records may appear. These lines of code ensure that the COBOL header record, and any records that are either completely blank or lacking key fields, are not included in the dataset.

EXAMPLE 3

To use this stored variable in the SAS dataset name, continue with the above SAS code and use the variable "dtepar" as part of the SAS dataset name as shown:

```
data sesug.final_&dtepar ;
set temp;
run;
```

EXAMPLE 4

To use this stored variable in the archived import filename, use the command "X copy":

```
X copy c:\sesug_import.zip c:\sesug_import_&dtepar..zip;
run;
```

CONCLUSION

Importing data into a SAS data warehouse can be automated using simple tools. Base SAS® and Winzip® version 10 with Command Line Support Add-On can be used on a Windows® XP Pro platform employing Windows Scheduled Tasks to allow the load process to be run at convenient times and without an operator. System managers can monitor results through automatic emails as well as inform users of the file update status. Although there are other products available, this method allows an agency with a tight budget to have more flexibility using some basic software products.

REFERENCES

SAS Companion for the Microsoft Windows Environment, Version 8 By SAS (Chapter 12).

Using Unnamed Pipes to Simplify Access to External Files. John Wenston, Pfizer Inc., New York, NY. Proceedings of NESUG, 1997.

ACKNOWLEDGMENTS

We would like to thank Richard (Keith) Hancock and Marie A. Bailey for their assistance.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Vicky Feldman
Florida Department of Corrections
2601 Blair Stone Rd
Tallahassee, FL 32399-2500
Work Phone: 850-488-1801
Fax: 850-488-1967
feldman.vicky@mail.dc.state.fl.us

Kristine Dougherty
Florida Department of Corrections
2601 Blair Stone Rd
Tallahassee, FL 32399-2500
Work Phone: 850-488-1801
Fax: 850-488-1967
dougherty.kristing@mail.dc.state.fl.us