Paper 067-29

SAS® Data Views Simply Stated

James C. Stokes, D&B, Short Hills, NJ

ABSTRACT

What is a SAS Data View? How does it differ from a SAS Data File? What is a SAS data set? These are common questions often asked by novice and experienced SAS users alike.

Many of us are faced with the daunting task of processing large volumes of data stored as non SAS data sets (e.g. flat files, etc.). By using SAS Data Views, we can dramatically reduce the amount of computer resources needed to process large volumes of data and simplify the way we process the non SAS datasets. Given the scarcity of computer resources and associated costs, it is critical to our success that we utilize SAS Data Views.

For example, when processing large non SAS data sets, securing work space often proves to be a major stumbling block to the successful execution of a given job stream. By utilizing SAS data views, one no longer needs to secure large amounts of work space to ensure successful completion of the job. As a result, the primary obstacle to processing large amounts of data can be easily eliminated.

This paper will define what a SAS data view is, when one would use a SAS data view, how to create a SAS data view, and explore some sample code.

SAS DATA VIEW DEFINED

In defining what a SAS Data View is, it is important to first recognize the evolution of a SAS Data Set.

Prior to the Version 6 release of SAS, the term “SAS data set” referred to a proprietary way the SAS system stored data. It included both the descriptor information as well as the data values in a file formatted by the SAS System.

With the release of SAS Version 6, the definition of a SAS Data set was expanded to include both a SAS data file, previously known as a SAS data set, and the “new” SAS data view.

A SAS data file stores the descriptor information and data values together.

A SAS data view is a definition of a virtual data set. It contains the information required to access the data values It is stored separately from the data values.

There are three types of SAS data views:

- **DATA step views** are stored, compiled DATA step programs.
- **PROC SQL views** are stored query expressions that read data values from their underlying files, which can include SAS data files, SAS/ACCESS views, DATA step views, other PROC SQL views, or relational database data.
- **SAS/ACCESS views** (also called view descriptors) describe data that is stored in DBMS tables.

The sample code found in this paper will deal exclusively with DATA step views. (For more detailed information about PROC SQL views and SAS/Access views, please refer to SUGI proceedings and/or the SAS Technical Support web site.)

WHEN TO USE A SAS DATA VIEW

Instead of using multiple DATA steps to merge SAS data sets by common variables, you can construct a view that performs a multi-table join.

You can save disk space by storing a view definition, which stores only the instructions for where to find the data and how it is formatted, not the actual data.

Views can ensure that the input data sets are always current because data is derived from views at execution time.

Since views can select data from many sources, once a view is created, it can provide prepackaged information to the information community without the need for additional programming.

Views can reduce the impact of data design changes on users. For example, you can change a query that is stored in a view without changing the characteristics of the view's result.

With SAS/CONNECT software, a view can join together SAS data sets that reside on different host computers, presenting you with an integrated view of distributed company data.

You can use views in the following ways:

as input to other DATA steps or PROC steps

to migrate data to SAS data sets or to database management systems that are supported by SAS

in combination with other data sources using PROC SQL

as pre-assembled sets of data for users of SAS/ASSIST software, enabling them to perform data management, analysis, and reporting tasks regardless of how the data is stored.

How to create a data step view

The syntax for creating a SAS data view in a data step is as follows:

DATA *data-set-name* VIEW= *view-name*;

The code names a view that the DATA step uses to store the input DATA step view. Where *view-name* must match one of the data set names coded on the data statement. (Please note: If you specify additional data sets in the DATA statement, SAS creates these data sets when the view is processed in a subsequent DATA or PROC step.)

SAMPLE CODE

Suppose you have a large flat file named **bigfile.dat** stored in the following directory on UNIX: **/mydir/companyx/data.** The data from **bigfile.dat** is to be merged with a smaller permanent SAS data file named **small** stored in the same directory. Only the observations common to both files are to be written out to a new permanent SAS data file named **newsmall**

Here is the SAS code to accomplish this task without the benefit of using a SAS Data View:

```
Filename in '/mydir/companyx/data/bigfile.dat';
Data bigfile;
    Infile in;
    Input @1   key    $3.
          @4   sales1 9.
          @13 sales2 9.
          ;
libname inout '/mydir/companyx/data';
Data inout.newsmall;
    Merge inout.small(in=ins) bigfile(in=inb);
    By key;
    If ins;
Run;
```

There is nothing wrong with this code per se. However, the job often fails to run successfully because sufficient work space is not available at the time the job is executed.

An excellent way to eliminate the need for the large amount of work space is to make use of a SAS data view.

Here is the SAS code to accomplish the same task as above using a SAS data view:

```
Filename in '/mydir/companyx/data/bigfile.dat';
Data bigfile view=bigfile;
    Infile in;
    Input @1   key    $3.
          @4   sales1 9.
          @13 sales2 9.
          ;
libname inout '/mydir/companyx/data';
Data inout.newsmall;
    Merge inout.small(in=ins) bigfile(in=inb);
    By key;
    If ins;
Run;
```

The only difference in the syntax is the view option coded on the first data step where the temporary SAS data file "bigfile" is being created.

By creating a SAS data view, the temporary SAS data file "bigfile" is no longer being created in work space. Therefore, we can now successfully process this job stream because we have eliminated the need for a large amount of work space where "bigfile" would have been stored prior to being merged with "small".

CONCLUSION

Data step views are a great way to fully utilize the power of SAS Procedures and data step features without first having to convert non SAS data sets into SAS data files. One of the primary benefits of processing data in this manner is the elimination of data replication.

CONTACT INFORMATION

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

James C. Stokes
D&B
103 JFK Parkway
Short Hills, NJ 07078
Work Phone: (973) 921-5909
Fax: (866) 772-7168
Email: stokesj@dnb.com
Web: www.dnb.com