

# DCSUG Newsletter

Washington, DC SAS Users Group

## First Quarter Meeting


When: Tuesday, March 9, 1999  
Time: 8:30a.m. to noon  
Place: Bureau of Labor Statistics  
Postal Square Building  
Room G440  
First Street North East  
Washington DC



Agenda:	8:30 - 9:00	Registration and refreshments
	9:00 - 9:15	Announcements
	9:15 - 9:30	Michael Bretz, Bureau of the Census <b>Version 7 SAS/ACCESS Software Examples</b>
	9:30 - 9:45	Paul Gorrell, Westat, Inc. <b>NOBS and NLOBS: Let SAS Count the Ways</b>
	9:45 - 10:00	Doug McAllaster, US Army <b>Using Formats with Proc SQL to Summarize Data</b>
	10:00 - 10:15	Nancy Patton, GE Capital <b>In &amp; Out of CNTL with Proc Format</b>
	10:15 - 10:30	Break
	10:30 - 10:45	Michael Raithel, Westat Inc. <b>What Sort of Input Should You Input to a Sort?</b>
	10:45 - 11:00	Chris Roper, Qualex Consulting Services, Inc. <b>Launching Excel from SAS intelligently, using SCL functions ported to Base SAS</b>
	11:00 - 11:15	Howard Schreier, Department of Commerce <b>Using the FTP Access Method</b>
	11:00 - 11:30	Ian Whitlock, Westat, Inc. <b>The RESOLVE Function - What Is it Good For?</b>
	11:30 - Noon	User-to User Questions and Answers Moderators - DCSUG officers

**\*Speakers and presentation times subject to change**

Visitors must use the visitor's entrance on First Street NE. Directions for those taking Metro: Take the Red Line to Union Station. Exit the station via the Amtrak exit, but do not go up to the train level. Instead, after going up to and out through the farecard gates, turn to the left where there is an exit to the street (First Street NE). Directly across from the exit are the employee and visitor's entrances to the building. After checking in with security, take the main elevators down to the G level. **BRING PICTURE ID!**

 If you have special needs and are planning to attend this meeting, please contact any DCSUG officer no later than 3/4/99

SAS® is a registered trademark of the SAS Institute, Cary, NC



**\*\*\*\*\* If you are not currently a paid member of the DCSUG and are planning to attend the meeting, please Email Karen Dennis at dennisk1@westat.com or call 301-294-3876 by Mar 4<sup>th</sup> and leave your name. Security at the Bureau of Labor Statistics is very tight and having your name on a list of possible attendees will make your entrance to the building easier. \*\*\*\*\***

### President's Message

Peter Knapp

We are very fortunate to have DCSUG members who have presented at Local, Regional & International SAS User Group meetings. A variety of short Coder's Corner papers will be presented at the March meeting. There should be a topic that would be of interest to every SAS user in the Washington, DC area. I look forward to seeing you at this very exciting and informative meeting. I hope, after listening to the presenters, you will volunteer to share your SAS expertise at a future DCSUG meeting.

## Speaker Abstracts

### Version 7 SAS/ACCESS Software Examples

Michael Bretz, Bureau of the Census

SAS Version 7 provides many enhancements allowing transparent access to a DBMS via dynamic engines. Examples of these enhancements will be presented using the SAS/ACCESS to Oracle product.

*Michael Bretz has been using SAS for 11 years serving as an in-house SAS consultant at the Bureau of the Census. He provides technical support, systems design and application development on the use of SAS software products for Economic censuses and surveys. Mr. Bretz received a BA in Mathematics from Millersville University.*

### NOBS and NLOBS: Let SAS Count the Ways

Paul Gorrell, Westat Inc.

This talk briefly describes some of the ways SAS makes available for determining the number of observations in a data set. For example: (i) the NOBS= option in the SET statement; (ii) accessing dictionary tables or using COUNT(\*) with PROC SQL; (iii) the ATTRN function. Efficiency considerations will also be discussed.

*Paul Gorrell, a systems analyst at Westat, has been using SAS since 1982. Much of his work has involved the design and implementation of statistical-analysis systems supporting experimental research. Before joining Westat, he worked at BLS on data restructuring and imputation procedures for the Occupational Employment Survey.*

### Using Formats with PROC SQL to Summarize Data

LTC Doug McAllaster, US Army, Pentagon, Washington DC

The SAS System provides powerful tools for summarizing data. Many SAS programmers use PROC MEANS in conjunction with PROC FORMAT to summarize detail (legacy - transactional) level data up to an aggregate (analytical - summary) level using user-defined, custom formats. Then, analysts can work with summary data in modeling complex statistical relationships for business decision making.

PROC SQL is a recent, powerful addition to the SAS programmer's arsenal. SQL provides a universal, not SAS specific, syntax for data processing. Furthermore, SQL provides aggregation capability with its GROUP BY clause.

Thus, many SAS programmers want to replace their usages of PROC MEANS with PROC SQL. Unfortunately, PROC SQL does NOT support the global format statement. It seems the powerful summary capability of user-defined, custom formats is not available to the SQL programmer. Nevertheless, the knowledgeable SAS programmer CAN create summary data sets based on custom formats. This presentation shows how to correctly correlate the SELECT clause with the GROUP BY clause to summarize data with custom, user-defined formats. Thus, SAS programmers CAN replace their PROC MEANS with PROC SQL without losing the capability of the global format statement.

*Doug McAllaster serves the Army as an officer personnel analyst. He has been using SAS as his main tool for 8 years. Doug has contributed papers at several SUGI's, NESUGs, SESUGs, and DCSUG meetings.*

### IN & OUT of CNTL with PROC FORMAT

Nancy Patton; GE Capital; Arlington, VA

CNTLIN=

If you are given account numbers in a data set and you are asked to identify all these accounts in numerous flat files, how might you approach this?

Using the SAS system and the CNTLIN= option in PROC FORMAT, a format can be defined where all the account numbers you want identified are set to a value of 'KEEP'. By using the PUT function to apply that format to the account numbers as you read the flat files, you could identify the records with the accounts you are looking for and control their destiny! This example will make the look-up process simple to understand and easy to bring back to work!

CNTLOUT=

Have you ever hard-coded a 'dummy' data set to combine with your data in order to force all categories of a formatted CLASS variable in PROC SUMMARY be included in your output? You could have used CNTLOUT= option and created that data set quite easily! You'll use this technique whenever you need to create a 'dummy' data set to force uniform output from PROC SUMMARY.

*Nancy Patton has been using SAS for about 15 years. She is currently a Senior Risk Analyst with GE Capital in Stamford, CT. She lives in Arlington, VA and tele-commutes from her*

home to work each day. She hasn't been to Stamford since September!! Her areas of expertise include MACRO and PROC REPORT. Most of her programming is done in an MVS/TSO batch environment. Over the last 10 years, Nancy has been attended and presented at SUGI, NESUG and SESUG, and is designated as the SUGI Conference Chair at SUGI in the year 2000. She is happy to be part of DCSUG since moving to the area in May 1998!

## What Sort of Input Should You Input to a Sort?

Michael A. Raithel, Westat

Because the SAS System makes sorting so simple, some programmers sort an entire SAS data set without giving it a second thought. But, efficiency-minded programmers often pause to consider what portion of a SAS data set actually needs to be input to PROC SORT. What if only a specific subset of a SAS data set's variables are required in the final sorted data set? Is it really necessary for the entire data set be sorted, or is there a more efficient alternative? This paper discusses different methods of inputting a SAS data set to PROC SORT. It compares the efficiency of sorting an entire SAS data set with three methods of subsetting the data set's variables and inputting them to PROC SORT. This practical information can be used by SAS programmers to improve the efficiency of their sorts.

*Michael A. Raithel specializes in designing, creating and implementing SAS applications that run in the MVS operating system on mainframe computers. He has a solid background in computer performance and capacity planning, as well as in applications programming. Michael was Co-chair of the NESUG'95 Conference in Washington, DC, and has been a Section chair for SUGI, NESUG, and SESUG conferences. He has frequently been an invited speaker at SUGI and NESUG conferences, has spoken at SESUG, and was invited to present a one-day seminar to members of the VIEWS SAS Users group in the United Kingdom. Michael's foray into the PC SAS world, a SAS/IntrNet application for the US Customs Service, netted him a nomination for the 1999 Computerworld Smithsonian Award for Government & Non-Profit Organizations. Michael is the author of "Tuning SAS Applications in the MVS Environment" published by the SAS Institute.*

## Launching Excel from SAS intelligently, using SCL functions ported to Base SAS

Chris Roper, Qualex Consulting Services, Inc.

The SAS Companion for the Microsoft Windows Environment illustrates one technique for opening Microsoft Excel from within SAS using DDE. This technique has several disadvantages. First, it does not check whether Excel is already open, it can unintentionally open multiple Excel sessions and this can be problematic. Second, it does not know with

certainty when it has opened Excel, it relies on the SLEEP function to pause processing until a set amount of time has passed. Hopefully, the amount of time is sufficient for Excel to open successfully, but there is no guarantee. Finally, since this set amount of time must be predetermined and sufficient for the worst case scenario, it is always set higher than the minimum amount of time required for Excel to open, this is inefficient and wastes end-user's time. There is another technique using SCL functions that have been ported to Base SAS that is much more intelligent. With these functions it is possible to have SAS check if Excel is open. If Excel is not open, then open Excel, and continually check if Excel has finished opening successfully. As soon as Excel has finished opening, then proceed with the SAS processing. This technique will not unintentionally open multiple copies of Excel, and will not waste time by pausing processing for a set number of seconds.

*Christopher Roper is a Senior Systems Developer with Qualex Consulting Services, Inc. His areas of expertise are base SAS, macros, AF, Frame, and SCL. Chris has a BS degree in Business Administration with a concentration in Economics from Christopher Newport University. He has been a SAS user for over 11 years.*

## Using the FTP Access Method

Howard Schreier, Department of Commerce

The FTP access method allows SAS external file references to point directly to FTP (File Transfer Protocol) servers on the Internet. Two examples will be presented. One uses INPUT statements for streamlined data acquisition. The second uses PUT statements for automated maintenance of static Web content.

*Howard Schreier is an economist by training. He serves the U.S. Department of Commerce in an information management role. He has used SAS since 1981 and maintains the DCSUG Web site.*

## The RESOLVE Function - What Is It Good For?

Ian Whitlock; Westat, Inc.; Rockville, MD

There are four DATA step functions that communicate with the macro facility.

1. CALL SYMPUT
2. SYMGET
3. CALL EXECUTE
4. RESOLVE

CALL SYMPUT and CALL EXECUTE send data step information to the macro facility. They are both very powerful tools. Numerous SUGI papers have been written on the wonders of CALL

SYMPUT. The more powerful CALL EXECUTE is less well known, but again one can find SUGI papers documenting its use.

On the other hand, have you ever seen a paper on SYMGET or RESOLVE? These corresponding counterparts get information from the macro facility into the DATA step. One often sees SYMGET employed in macro code, but almost never in a place where the code would not be better written without it. Now RESOLVE is a stronger SYMGET, but I do not even remember one program that even uses RESOLVE, other than those provided by the SAS Institute for explanatory purposes. This paper presents and solves a problem where RESOLVE is crucial to an easy solution.

Ian Whitlock is a senior systems analyst at Westat working on survey research problems. He has been programming largely in SAS since 1981 using an interactive environment to develop programs for execution in a batch environment. He is skilled in problem solving involving SAS macro, PROC SQL, reporting procedures and the DATA step. In addition to programming, he teaches SAS courses to Westat employees. He is active on SAS-L, and has presented many different papers at national, regional, and local SAS User Group meetings.

## Questions and Answers

Moderated by DCSUG Officers

Bring your SAS programming or procedure questions to DCSUG and its members. During this open session, you can ask questions of and get answers from experienced SAS users. All are welcomed to participate



## Desktop Special Interest Group

by David Kiasi-Barnes

DCSUG's Desktop Special Interest Group will have its next meeting on April 28, 1999 at 7:30PM at Westat Inc., 1650 Research Blvd. in Rockville, Md. The topic will be, "**Using SAS Software to Analyze Sybase Performance on the Web**", given by Joe Mirabal of America Online and Zhengyu (George) Wang.

Our October meeting featured an in-depth view of SAS Enterprise Data Miner, given by Lauren Lieberman, systems engineer, of SAS in the Philadelphia area. Overall remarks were made by Ingela Welbrock, account manager of SAS Institute. The audience enjoyed the presentation very much remarking how thorough it was.

For the April meeting, we welcome any suggestions you may have for future meeting topics or speakers; if you would like to be added to our group's mailing list so you can get announcements of upcoming meeting events, please contact David Kiasi-Barnes at (301) 350 4752 or Yesvy Gustasp at (301) 589-4530. Also, if you need a ride from the Metro, please call either one of us. The meeting will be held in the Human Resources meeting room, room #112C, on the first floor. The phone number in the meeting room is (301) 294-3973 if you are lost and need to contact us.

To get to Westat, take I-270 8 miles from the Capital Beltway to exit #6B - Route 28 (toward Darnestown). Once on 28, go past the traffic light at the Shell station and turn right at the next traffic light, which is Research Blvd. The 1650 building is less than 1/4 mile on the left, just over the crest of the. You can park in any non-reserved space. Signs will be posted to direct you to the meeting room.

If using Metro, take the Red line towards Shady Grove and get off at the Rockville station. Take Ride-On bus #54 in the direction of Lakeforest Mall. The bus goes out on 28 past I-270 and the Shell station and then turns onto Research Blvd. It takes 8-10 minutes from the Metro station to Westat. The #54 bus runs every half hour.

Note: Don't be confused by a large Westat sign that is in front of another Westat building on Research Blvd (1500 & 1550).

## DCSUG 1999 Steering Committee OFFICERS

### **CHAIR -- Peter Knapp**

(202) 482-1359 (202)482-1388(fax)  
peter\_knapp@ita.doc.gov

### **SECRETARY -- Larry Altmayer**

(301) 457-2581 (301) 457-1777 (fax)  
laltmaye@census.gov

### **TREASURER -- Arlene Siller**

(301) 436-8522 x188 (301) 436-5452 (fax)  
abs2@cdc.gov

### **SAS LIAISON -- Linda Atkinson**

(202) 694-5046 (202) 694-5718 (fax)  
atkinson@econ.ag.gov

### **OTHER COMMITTEE MEMBERS**

#### **Karen Dennis**

(301) 294-3876 (301) 294-2034 (fax)  
dennisk1@westat.com

#### **Frank Fry**

(202) 452-2666 (202) 452-6433 (fax)  
ffry@frb.gov

#### **Michael Raithe**

(202) 927-0675 (202) 927-1896 (fax)  
maraithe@erols.com

#### **Mike Rhoads**

(301) 251-4308 (301) 294-2040 (fax)  
rhoadsm1@westat.com

#### **Howard Schreier**

(202) 482-4180 (202) 482-4614 (fax)  
Howard\_Schreier@ita.doc.gov

#### **Douglas McAllaster**

(703) 697-7619  
mcalldl@hqda.army.mil

### **DESKTOP SPECIAL INTEREST GROUP**

#### **CHAIR -- David Kiasi-Barnes**

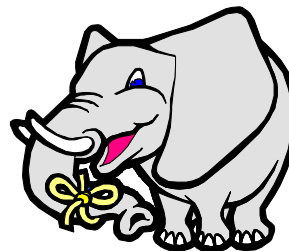
(301) 350-4752 (301) 350-4785 (fax)  
appalt@aol.com

#### **VICE-CHAIR -- Yesvy Gustasp**

(301) 589-4530 (202) 633-2668 (fax)  
Yesvy.Gustasp@mix.cpcug.org

## 1999 Meetings

Plan ahead and mark your calendars now. The DCSUG will tentatively meet on the following dates in 1999: June 8, September 14, December 7. Meeting times and locations to be announced. If at any time you have a question about an upcoming meeting or other DCSUG information, call one of the members of the Steering Committee.



### Remember to Join DCSUG!!!

Don't forget to renew your DCSUG membership for 1999. Dues are used to defray the costs of producing and mailing the newsletter and provide refreshments at our meetings. You'll make sure that you keep receiving the newsletter and keep DCSUG going. If you have not joined DCSUG, now is a great time to do so. Joining DCSUG is as easy as completing the membership form included in this newsletter. Individual memberships are only \$10; corporate memberships are \$50.

## You're Invited

### To visit the DCSUG home page

The URL for the DCSUG home page on the Web:  
<http://www.ita.doc.gov/industry/otea/dcsug>



# SAS Talk

## By Ian Whitlock

Let's take a look at parameters. For SAS they come in two varieties - program parameters and macro parameters.

Program parameters are defined at the program level and control how the program as a whole functions. They are defined with the %LET macro statement creating a global macro variable. The values are then referenced with an ampersand, &, in front of the name. As program parameters the assignments should be defined at the top of the program. They form a contract between the user and the program - given these assignments the program will carry out its task.

```
%let mode = test ;
%let month = 199903 ;

/* ----- */

libname project "n:\abc\&mode\dat" ;
filename out "n:\abc\&mode\dat\rep&month.txt" ;
filename progs "n:\abc\&mode\pgm" ;

%inc progs ( report1 ) ;
```

From this little example we see that program parameters typically control large objects like libraries and files. Here MODE allows us to run in production mode or test mode. Note that there are two parallel systems of files. MODE essentially determines which system we are on, so at the setting of one parameter we can test the program and run it in production. Why not use

```
%let root = "n:\abc\test" ;
```

It hides the function of separating test mode from production mode and it now places a stronger requirement on the organization of the files - everything has to go off one root directory or we need more parameters to accomplish the test/production separation. A little better would be

```
%let root = "n:\abc" ;
```

But it suggests that the program may execute on a lot of different root directories, where in fact the program was written for a specific project that has a very stable setup of files. Thus it hints at more flexibility than the program requires. Flexibility is good, but needless flexibility is not, and it can hurt because it is misleading. We don't want to suggest that the user can change the root; he cannot. Now what about MONTH? We used it as part of the member REP&MONTH. Why not use

```
%let flname = rep199903 ;
```

As before, we would always have to type the letters, REP. In addition one can imagine that the month may be used in many places in the program, for example, in titles and data set names. By putting too much information in the parameter, it becomes less useful. If you find the parameters defining small ideas in the program, then they are probably being defined at the wrong level. This is an indication that some of the code should be written as a macro and that your parameters are actually parameters of that macro.

A macro can be thought of as a unit of SAS code and the macro parameters control this unit of code. They are defined with default assignments in the %MACRO statement.

```
%macro report ( data = &syslast ,
                out = _data_ ,
                vars = _all_ ) ;
```

Notice the names of the parameters have been chosen to correspond to SAS parameter names on procedures. Even the default values \_DATA\_ and \_ALL\_ are typical key words in SAS that should not require explanation. The value, &SYSLAST, should be discussed, but it also follows the convention that if the input data set name is not provided, then use the last created SAS data set. SYSLAST is an automatic variable created by the system holding the name of the last created data set. When is the variable evaluated? Remember that from the %MACRO statement down to the %MEND statement, the macro is simply compiled, i.e. stored away for future use. No resolution of macro variables is made at this time. In particular the reference, &SYSLAST is stored, not the value of the reference, which would most likely be inappropriate most of the time. During invocation, if the DATA parameter is not given, the reference is resolved. That, of course, is precisely when it makes sense to resolve the reference.

Now consider the ubiquitous NOBS macro, which should assign a macro variable the number of observations in the given data set. How should the communication take place? Should the variable name be assumed, or should it be part of the parameter list? I favor making it part of the parameter list because I think the parameter list should



be a complete list of understandings that are required for use of the macro. In addition, it makes it easier to use the macro. Suppose we insisted that the consumer always look in a global variable NOBS. What would happen if a user already had a variable NOBS? What would happen if there were more than one call to %NOBS in the user's code? Each call writes over the previous value assigned. I would suggest the interface:

```
%macro ( data = &syslast , nobsv = nobsv ) ;
```

Here NOBSV specifies the variable whose value is to be assigned. For convenience the default variable NOBS will be declared global by the macro. For all other values, it is the user's responsibility to declare the variable. It may be global or local. Say the variable is MYNOBS. How does it get a value in the macro? The reference, &NOBSV, resolves to MYNOBS; hence the macro knows the name of the variable. Here is the macro as defined in the pre-SQL/SCL days.

```
%macro nobsv ( data = &syslast , nobsv = nobsv ) ;
  %if %upcase(&nobsv) = NOBS %then
    %global nobsv ;
  data _null_ ;
    call symput ("&nobsv", left(put(nobsv,8.)) );
    stop ;
    set &data nobsv = nobsv ;
  run ;
%mend nobsv ;
```

Now the macro has the simplicity of creating a fixed global variable, while providing the more sophisticated user with the means of using his/her variable names. The only cost is that he/she must declare the variable before calling the macro.

One problem is not illustrated in the code. What if we wish to write to the log a message giving the value of the variable?

```
%put Note: Data set &data has &&&nobsv observations. ;
```

Here one needs three ampersands. On the first scan two ampersands resolve to one and &NOBSV resolves to the user's variable name. On the second scan the macro facility sees a reference to the user's variable and provides it.

The issue of parameters is important. They control how easy it will be to use your code, and they control how well self-documented the code will be. They also provide the means to get a lot more leverage of code, through flexibility, with relatively minor changes to that code.

---

---

## Problem Solvers Needed

**WESTAT** is looking for creative, motivated professionals who enjoy developing SAS-based solutions in areas such as data collection and management, statistical analysis, and reporting. As one of the nation's leading employee-owned research companies, we employ more than 300 computer professionals to support a variety of projects in health, education, and other areas. SAS software has been a significant element of our information management strategy for the past 20 years. We maintain an active in-house SAS users' group and offer a variety of SAS-related training courses. Westat staff are often on the leading-edge of new SAS technologies, and they participate actively in the SAS user community.

We have multiple openings at various levels for staff who can design and develop sophisticated SAS programs, using DATA step, SQL, and other tools. Previous background in a research environment is a plus, as is experience with DBMS, Microsoft Access, or other programming languages.

**Westat** offers a competitive benefits package that includes participation in the Employee Stock Ownership Plan (ESOP), a 401k Plan, and life and health insurance plans. For immediate consideration, please send your resume and cover letter by **one** of the following methods to:

**WESTAT**

Dept. 99-313

1650 Research Blvd.\* Rockville, MD 20850

E-mail: [HR@westat.com](mailto:HR@westat.com) \* FAX: (301) 294-2040

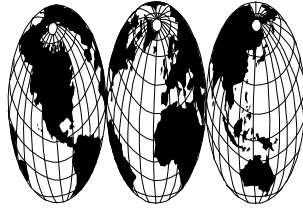
For a listing of other Westat opportunities, visit us at [www.westat.com](http://www.westat.com).

EOE

DCSUG, First Quarter 1999, page 8



CAPITAL★TRADE  
INCORPORATED



*Capital Trade Inc. is an economic consulting firm specializing in international trade.*

## **SAS Programmer Needed**

We are seeking a highly motivated, detail-oriented SAS programmer familiar with BASE SAS in the Windows environment. Must have experience in creating and manipulating large data bases. In order to succeed, must be able to respond quickly to data requests from our multinational clients. Must have good oral and written communications skills. Experience with other data base packages a plus.

*1200 18<sup>th</sup> Street, N.W., Suite 601, Washington, D.C. 20036*

*Phone: 202-463-1850*

*Fax: 202-463-1855*

*Internet: [captrade@aol.com](mailto:captrade@aol.com)*

*Attn: Mary Ann McCleary*

**JOHNS HOPKINS**  
UNIVERSITY

## **KRIEGER SCHOOL OF ARTS & SCIENCES**

### **PART-TIME GRADUATE PROGRAMS**

## **MASTER OF ARTS IN APPLIED ECONOMICS**

**1776 Massachusetts Avenue, NW**

**Washington, DC 20036**

**202-452-0714**

**Designed for full-time working adults**

**Dupont Circle Metro Station**

**[pgp-as@jhu.edu](mailto:pgp-as@jhu.edu)**

**April Open House - Call for date and directions**

# Membership Application/Renewal

Date: \_\_\_\_\_

Name: \_\_\_\_\_

Company Name: \_\_\_\_\_

Mailing Address: \_\_\_\_\_

City, \_\_\_\_\_ State, \_\_\_\_\_ Zip \_\_\_\_\_

Phone # \_\_\_\_\_ Fax # \_\_\_\_\_

E-Mail: \_\_\_\_\_

Preferred mode of contact (check one):

Fax e-mail

SAS Version(s) \_\_\_\_\_

SAS Product(s): \_\_\_\_\_

Operating System(s) \_\_\_\_\_

Check one:

Individual membership Corporate sponsorship for 1 year is \$10.00for 1 year is \$50.00

Please indicate a contact person for Corporate Sponsorships!

Check one: New membership Renewal membership

Mail to:

Washington DC SAS Users Group

P.O. Box 44670

Washington, DC 20026-4670

**Washington DC SAS Users Group**

**P.O. Box 44670**

**Washington, DC 20026-4670**