

DCSUG Newsletter

Washington, DC SAS Users Group

First Quarter Meeting

When: Tuesday, March 6, 2001
Time: 8:30a.m. to noon
Place: **Bureau of Labor Statistics**
Postal Square Building
Room G440
First Street North East
Washington DC



**We are back at
BLS**

Agenda: 8:30 - 9:00 Registration and refreshments
9:00 - 9:15 Announcements
9:15 - 9:45 Larry Altmayer, U.S. Census Bureau
CONNECTing with SAS Version 8 : Virtually Transparent to the User
9:45 - 10:15 Mike Rhoads, Westat
Hidden Nuggets in Version 8: New Informats, Formats, and Functions
10:15 - 10:30 Break
10:30 - 11:30 Chris Zogby, Zogby Enterprises
Object Oriented Programming Using SAS/AF Software: An Electronic Mail Class Example
11:30 - Noon User-to User Questions and Answers
Moderators - DCSUG officers
***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 2/28/01

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

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

FFFFFFFF If you are not currently a paid member of the DCSUG and are planning to attend the meeting, please Email Larry Altmayer at Lawrence.W.Altmayer@census.gov or call 301-457-2581 by Feb 28th 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. FFFFFFFFFF

President's Message

Peter Knapp

This quarter we are lucky to have not one DCSUG meeting but two. Our regularly scheduled morning meeting will be held on March 6. Chris Zogby and DCSUG's very own Larry Altmayer and Mike Rhoads will be presenting. The evening of May 3 Andrew H. Karp and Art Carpenter will be presenting papers at special DCSUG meeting held in downtown DC. I'm looking forward to all these presentations and hope you can make both meetings.

In addition, SUGI 26 will be held April 22-25 in Long Beach, California. DCSUG Steering Committee member Frank Fry is chairing this conference and it promises to be both fun and informative. It's not too late to make plans to go!

Speaker Abstracts

CONNECTing with SAS Version 8 : Virtually Transparent to the User

Larry Altmayer, U.S. Census Bureau

Connecting from personal computer (PC) platform to mainframe network is straightforward with SAS® version 8. We know it can be more efficient in many ways to store large data sets in a mainframe environment, and access them from the PC. This paper looks at using PC SAS/CONNECT® to access SAS course data sets from a UNIX® mainframe, a concept readily extended to larger data sets in practice. We will also touch on the basic configuration for accessing a SAS/AF® application when clicking on a SAS icon on a Windows® desktop. Next we look at ways to verify the data are being read from the UNIX mainframe.

We will also see that it is easy to convert current version 6 applications to version 8, using Destiny course materials as a starting point. Although version 8 contains many new object-oriented features, this paper will focus on converting what we already have in version 6, directly to version 8 using PROC COPY, with the version 6 legacy object concept.

Larry Altmayer is a Computer Specialist at the Census Bureau. He has been the Secretary of DCSUG for several years. He has been a Section Chair, Session Coordinator, and presenter at SUGI and other regional conferences.

Hidden Nuggets in Version 8: New Informats, Formats, and Functions

Mike Rhoads, Westat

Much of the interest in Versions 7 and 8 of the SAS® system has been in major new features such as the Output Delivery System and long variable and table names. Hidden within the "Changes and enhancements" documentation, however, are many other useful new items. This presentation summarizes several enhancements to SAS informats, formats, and functions, and describes how they can make life easier for programmers who know about them.

Mike Rhoads has served on DCSUG's steering committee for the past several years and chaired its "Discover DC" committee for NESUG '99. He has also contributed to various SUGI and NESUG conferences as a paper presenter and session coordinator. 2001 marks the 22nd anniversary of his initiation into the wonderful world of

SAS software. He is currently a Vice President in the Computer Systems and Applications group at Westat, where he still manages to sneak in an occasional DATA step or two when not overburdened by his other responsibilities.

Object Oriented Programming Using SAS/AF® Software: An Electronic Mail Class Example

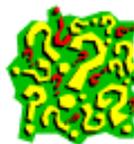
Christopher Zogby, Zogby Enterprises

Object-oriented programming (OOP) can greatly ease the burden of application development. One valuable feature of OOP is the persistent nature in which objects store their data and operations. The encapsulation of an object's data and operations make objects ideal for storing and processing application data. An electronic mail class is used to demonstrate why this feature of OOP adds value to the application development process.

Chris Zogby is an independent contractor who lives in the Washington, DC metropolitan area. Chris's professional interests include SAS application development, data warehousing, data mining and customer relationship management. In addition, Chris holds a BS and MA degree in Biology.

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

Plan now to attend a **special evening "Desktop SIG" DCSUG meeting on Thursday, May 3rd from 6 pm to 8:30 pm** at George Washington University's K Street Center for Professional Education 2020 K Street NW. The Center is conveniently located near the Farragut West Metro station on the Blue/Orange lines. Light refreshments (cookies and sodas) will be served.

The meeting will feature presentations by Art Carpenter and Andrew Karp, both well-known SAS consultants and trainers, who will be giving a series of SAS Software seminars from April 30 to May 5th at the K Street Center. DCSUG appreciates their willingness to sponsor and speak at this special evening meeting.

Their presentations are:

Art Carpenter, California Occidental Consultants

"Programming for Job Security Revisited: Even More Tips and Techniques to Maximize your Indispensability"

Abstract: Are you worried about job security in this current economic downturn? Even if you aren't, this lighthearted 50 minute presentation shows you how to write SAS code that no one will understand, that won't work (or won't work the way it is supposed to) and otherwise confuse others who have to work with it. This humorous presentation has brought down the house at SUGI, as well as when it was presented to SAS users in South Africa and the United Kingdom. If you thought there was no humor in the SAS System, just come and hear Art's talk.

Speaker Biography: Art Carpenter is a well-known SAS Software consultant and trainer, and has been selected as Chair of the SUGI 28 Conference in 2003. Art has written two best-selling books for the SAS Books By Users (BBU) program, "A Quick Start to SAS/GRAPH Software" and "Carpenter's Complete Guide to the SAS Macro Language," now in its second printing. Art has presented this and other papers at SUGI, NESUG, WUSS and the SAS United Kingdom Users Group, VIEWS. Art is a SAS Certified Professional and his firm is a Quality Partner(tm) of SAS Institute, Inc.

Andrew Karp, Sierra Information Services, Inc.

"Working With SAS Date and Time Functions"

Abstract: this beginning tutorial is suited to new users of SAS System Software that need to learn the basics of how SAS treats variables representing dates and times. Updated with the latest tools now available in Version 8, this tutorial helps you understand how the SAS System stores date and time variables and how to operate on them in the Data Step. You will also learn key SAS data step functions that you use with date and time variables, and the role of formats to portray their values. The tutorial also shows you how to calculate the amount of time that has elapsed between two dates, and how different approaches to this problem will yield different results. SAS functions such as INTCK, INTNX, and DATDIF (new to Version 8) will be explained and de-mystified. Finally, the tutorial explains how PROC EXPAND, in the SAS/ETS module, is used with data collected in the time domain.

Speaker Biography: Andrew Karp is an internationally recognized SAS Software consultant and trainer who has been a regular speaker at DCSUG meetings, as well as at NESUG, SUGI, SEUGI, and SAS user groups meetings in Australia and New Zealand. A SAS Certified Professional, Andrew graduated from The George Washington University in DC and has had his own firm, Sierra Information Services, Inc., now located in the California Wine Country, for the past seven years. His was the first consultancy in the western United States to be selected as a Quality Partner of SAS Institute.

Advance registration is not required to attend the meeting, but seating is limited, so arrive early and enjoy!

Please contact Yesvy Gustasp or Joe Mirabal (see steering Committee column) with questions, requests, suggestions, and offers for future meetings.

DCSUG 2000 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) 457-4437 (fax)
Lawrence.W.Altmayer@census.gov

TREASURER -- Arlene Siller

(301) 458-4498 (301) 458-4032 (fax)
asilier@cdc.gov

SAS LIAISON -- Mike Rhoads

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

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

Christopher Zogby

(703) 671-6330 (703) 671-6360 (fax)
czogby@erols.com

Linda Atkinson

(202) 694-5046 (202) 694-5718 (fax)
Atkinson@ers.usda.gov

Howard Schreier

(202) 482-4180 (202) 482-4614 (fax)
Howard_Schreier@ita.doc.gov

DESKTOP SPECIAL INTEREST GROUP

CO-CHAIR -- Joe Mirabal

(703) 265-4620
jmmirabal@aol.com

VICE-CHAIR -- Yesvy Gustasp

(301) 589-4530 (202) 307-5888 (fax)
Yesvy.Gustasp@mix.cpcug.org

NESUG Conference Proceedings for Sale

DCSUG will have copies of the Proceedings of the NorthEast SAS Users Group Conference available for sale at the March meeting. These contain a wealth of information on a wide variety of topics. The CD version, which will sell for \$15, has papers from the 1997, 1998, 1999, 2000 meetings. The hardcopy of the 2000 Proceedings will sell for \$10, and while supplies last, a copy of the 1999 Proceedings will be given free with each purchase.

Due to the high cost of mailing out newsletters, DCSUG will be emailing notices of the availability of accessing the quarterly newsletter (and other important announcements) at the DCSUG Internet site. If you are a member, please make sure your email address is current in our database. Please notify Larry Altmayer at Lawrence.W.Altmayer@census.gov with the email address you want to use to receive notification of the publication of the newsletter. Members will still have the option of continuing to receive a hard copy of the newsletter upon request.

NESUG '01 Conference

The NorthEast SAS Users Group (NESUG) will hold its annual conference, September 30-October 3, at the Renaissance Harborplace Hotel in Baltimore, Maryland. An expanded two and a half day format will give attendees more time to hear presentations, visit the SAS Demo Area and the Code Clinic, and network with other SAS users. Conference details, an online abstract submission process, and a volunteer application are available at <http://www.nesug.org>. Deadline for submitting an abstract for consideration is March 31. Make plans now to join Conference Co-Chairs Linda Atkinson and Sue Douglass for NESUG '01!

Calendar of Meetings and Events

This feature is meant to give users an idea of some of the activities of area users groups and special interest groups, as well as regional and international groups. If your group would like to be regularly included in this feature, please call Mike Rhoads or one of the DCSUG officers. There is a DCSUG directory with phone, fax, and E-mail contact information elsewhere in this newsletter.

March

6 **DCSUG** DC SAS Users Group, General Meeting
Time: 8:30 am - 12:00 noon
Site: Bureau of Labor Statistics
2 Massachusetts Ave. N.E.
Program: See front page announcement
Contacts:

Mike Rhoads at (301) 251-4308
RhoadsM1@Westat.com
Peter Knapp at (202) 482-1359
peter_knapp@ita.doc.gov

April

22-25 **SUGI 26** SAS Users Group International
Site: Long Beach Convention Center
Long Beach, California
Chair: Frank Fry
Further Info:

www.sas.com/usergroups/sugi/sugi26/intro.html

May

3 **DCSUG** DC SAS Users Group, Desktop SIG Meeting
Time: 6:00 - 8:30 pm
Site: George Washington Univ.
2020 K Street N.W.
Program: See article

June

5 **DCSUG** DC SAS Users Group, General Meeting
Time: 8:30 am - 12:00 noon
Site: TBA
Program: TBA

September

11 **DCSUG** DC SAS Users Group, General Meeting
Time: 8:30 am - 12:00 noon
Site: TBA
Program: TBA

Other Groups

CENSUG Census SAS Users Group
Contact: Andrea Potter at (301) 457-1687

MDSUG Maryland SAS Users Group
Contact: Mr. R.H. Miller, (410) 740-4229
millerrh@juno.com

NCHSSUG National Center for Health Statistics SAS Users Group
Contacts: Arlene Siller at (301) 458-4498
asiller@cdc.gov
Linda Tompkins at (301) 458-4533

NIHSUG National Institutes of Health SAS Users Group
Contact: Ray Danner at (301) 496-6037

NOVASUG Northern Virginia SAS Users Group
Contact: Joe Mirabal at (703) 265-4620
jmmirabal@aol.com

VASUG Virginia SAS Users Group (Richmond area)
Contact: Peter Rikard at (804) 828-4878
prikard@mail.ucc.vcu.edu

Object-oriented Programming Using SAS/AF[®] Software: An Electronic Mail Class Example

Christopher Zogby
Zogby Enterprises, Alexandria, VA
czogby@erols.com

Abstract

Object-oriented programming (OOP) can greatly ease the burden of application development. One valuable feature of OOP is the persistent nature in which objects store their data and operations. The encapsulation of an object's data and operations make objects ideal for storing and processing application data. An electronic mail class is used to demonstrate why this feature of OOP adds value to the application development process.

Introduction

While working for a particular client, our development team was responsible for writing a SAS AF/Frame application that allowed its users to create LAN-based Excel spreadsheets from SAS data in mainframe data warehouse. The application fulfilled its role successfully, however it would have been even more valuable if it was able to inform project management which of the 200 plus spreadsheets had been updated. The application's users were saddled with this task, until we developers realized SAS supported electronic mail services. Once we became familiar with the necessary syntax, all that stood in the way of adding electronic email services was a decision on how it should be integrated.

A short debate took place before it was decided that email services were to be added using a structured programming approach. This was mainly due to the fact that very little of the AF/Frame application was designed using object-oriented techniques. Much of the new email code was spread between two screen control language (SCL) method blocks. (This application was developed using SAS Version 6.12. In later SAS versions, screen control language was replaced with SAS component language). To inform the necessary parties of data updates, a user would select a check box that, when in the 'checked' position, caused the application to call the necessary email methods. Email method parameters were used to define the email's message line, message body content and recipients. The email software was unit tested, integrated into the application, then the application was tested. All was well until the application needed to be modified to accommodate the host machine's system constraints.

The host machine, which was not dedicated to our application, occasionally failed to have the necessary disc space to accommodate both 'downloaded' mainframe data and their respective Excel spreadsheets. The application code was modified to delete local SAS data sets once their respective spreadsheets were created. This was an adequate solution to the unpredictable availability of disc space, however it caused us to have to reorganize the existing email software. At this point I realized, for a variety of reasons, that the ensuing email software problems could have been obviated had email services been provided in an object-oriented manner. Time was spent abstracting the properties and actions necessary to assemble a generic Email class.

OOP concept of encapsulation causes an object, its data and operations to exist as one entity. All the operations and data necessary to assemble and dispatch an email are defined entirely within the Email class. While in existence, all data passed to the email object (i.e. message line, message body content, list of recipients, etc.) are persistently stored until explicitly cleared, overwritten or the object is destroyed. The distinction between the application code and email services software as well as an email object's ability to persistently store its own data and operations made the O-O solution more attractive than the original structured programming solution.

Email Class Definition

The Email class was created by subclassing SAS's Object class (SASHELP.FSP.OBJECT.CLASS). By virtue of inheritance, all of the Object class's data and operations, known in SAS version 6.12 as instance variables and methods, are available to the Email class. Abstraction of the proposed Email class's functions led to the addition of 6 new instance variables and 16 new methods. Table 1 lists the new instance variables' name, whether or not they are automatic, and their data type. Note that automatic instance variables are created at object instantiation. Possible data types for instance variables (version 6.12) are numeric, character and list.

The instance variables in Table 1 were considered necessary to allow software developers full utilization of SAS's supported email services. Email addresses to be listed as 'to' and 'cc' (carbon copy) recipients are stored in the *to* and *cc* lists. Email subject line

and message body are stored in the object's *subject* variable and *body* list. File attachments are stored as full directory paths in the attachment variable. The *master_list* variable is used to manage the object's instance variables.

Table 1. Email Class's New Instance Variables.

NAME	AUTOMATIC	TYPE
ATTACHMENT	YES	CHAR
BODY	YES	LIST
CC	YES	LIST
MASTER_LIST	YES	LIST
SUBJECT	YES	CHAR
TO	YES	LIST

The Email class's 16 new methods, which either clear, set or retrieve object data, or dispatch an email, are listed in Table 2. Table 2 lists the new methods' name, their parameters' data type and their respective utility. In addition to adding new methods, the inherited *_INIT_* method was overridden to include code that organizes an email object's instance variables by inserting them into the *master_list* list.

Table 2. Email Class's New Methods.

NAME	PARAMETER DATA TYPE	UTILITY
CLEAR_ALL	N/A	Clears all lists, and sets all instance variables to null.
CLEAR_ATTACHMENT	N/A	Clears ATTACHMENT variable.
CLEAR_BODY	N/A	Clears BODY list.
CLEAR_CC	N/A	Clears CC list.
CLEAR_TO	N/A	Clears TO list.
GET_ATTACHMENT	CHAR	Retrieves value stored in ATTACHMENT field.
GET_BODY	LIST	Retrieves list content stored in BODY list.
GET_CC	LIST	Retrieves list content stored in CC list.
GET_SUBJECT	CHAR	Retrieves value stored in SUBJECT field.
GET_TO	LIST	Retrieves list content stored in TO list.
SEND_MAIL	N/A	Assembles and dispatches email from stored data.
SET_ATTACHMENT	CHAR	Sets ATTACHMENT variable value.
SET_BODY	LIST	Sets BODY list content.
SET_CC	LIST	Sets CC list content.
SET_SUBJECT	CHAR	Sets SUBJECT variable value.
SET_TO	LIST	Sets TO list content.

Sample Code

In order to use an email object we must first load its class, then, through the process of instantiation, create an object. In version 6.12 SCL the *LOADCLASS* and *INSTANCE* functions are used to load the Email class and create an email object. The code below demonstrates the proper syntax.

```
* Load Email Class;
EMAIL_CLASS = LOADCLASS(' LIBNAME. CATALOG. EMAIL. CLASS' );

* Create Instance of Email Class (i.e. Email Object);
EMAIL_OBJECT = INSTANCE(EMAIL_CLASS);
```

The object, *EMAIL_OBJECT* is ready to receive data for our proposed email. First let's pass email addresses for the 'to' recipient list. We will be passing this information using the email object's *SET_TO* method. Note that this method's parameter is of *LIST*

data type. The section of code below shows the necessary SCL to create a list, populate it with email addresses, and pass the data to our email object.

```
* Create List For TO Recipients;
TO_ADDRESSES = MAKELIST();

* Populate List With Email Addresses;
RC = INSERTC(TO_ADDRESS, 'DICK@AOL.COM', -1);
RC = INSERTC(TO_ADDRESS, 'JANE@AOL.COM', -1);

* Set The Object's TO List With TO_ADDRESS List;
CALL SEND(EMAIL_OBJECT, 'SET_TO', TO_ADDRESS);
```

Now let's set the subject line and attach a SAS configuration file.

```
* Set The Email Subject Line;
CALL SEND(EMAIL_OBJECT, 'SET_SUBJECT', 're: Requested Config File');

* Set Configuration File As Attachment;
CALL SEND(EMAIL_OBJECT, 'SET_ATTACHMENT', 'C:\SAS\CONFIG.SAS');
```

Finally, let's create the email body, pass its value to the object, and then send the email.

```
* Create List For Email Message Body;
BODY = MAKELIST();

* Populate List With A Message;
RC = INSERTC(BODY, 'Dear Dick and Jane:', -1);
RC = INSERTC(BODY, ' ', -1);
RC = INSERTC(BODY, 'Please find the attached SAS config file.', -1);
RC = INSERTC(BODY, ' ', -1);
RC = INSERTC(BODY, 'Regards, ', -1);
RC = INSERTC(BODY, ' ', -1);
RC = INSERTC(BODY, 'Chris', -1);

* Set Email Message Body;
CALL SEND(EMAIL_OBJECT, 'SET_BODY', BODY);

* Send Email;
CALL SEND(EMAIL_OBJECT, 'SEND_MAIL');
```

Our email regarding the requested SAS configuration file has been sent to our recipients. Now we need to destroy our object and our loaded Email class. The code below returns back to memory, that which was used to store our loaded class and object.

```
* Destroy Email Object;
CALL SEND(EMAIL_OBJECT, '_TERM');

* Destroy Email Class;
CALL SEND(EMAIL_CLASS, '_TERM');
```

Conclusions

The email object's ability to store its own data and operations make the OOP solution superior to the original structured programming attempt. In this particular case, OOP provides a more robust and versatile solution with far less programming effort. This is mainly due to the OOP concepts of encapsulation and inheritance. Furthermore, the generic Email class can be used to create a variety of specific Email subclasses without causing code redundancy. Further attempts to increase application programming efficiency by using OOP techniques should be explored.

Please feel free to send email to czogby@erols.com, if you wish to receive additional information on, or source code for the Email class.

Looking for something better?

Got SAS?

Look to
Smith Hanley
Associates!

Smith Hanley offers a diversity of placement opportunities for SAS programmers and statisticians.

Choose from permanent opportunities with leading companies worldwide. Your career path can take on an exciting new direction. So, if you are looking for a change, look to Smith Hanley!

SAS Opportunities!

Call or email us today!

Rachel Pereira
(203)319-4300

rpereira@smithhanley.com

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

for 1 year is \$10.00

Corporate sponsorship

for 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

2001 Meetings

Plan ahead and mark your calendars now. The DCSUG will meet on the following dates in 2001: March 6, June 5, September 11, December 4. 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.

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>

Remember to Join DCSUG!!!



Don't forget to renew your DCSUG membership for 2001. 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

Washington DC SAS Users Group
P.O. Box 44670
Washington, DC 20026-4670