DCSUG Newsletter

Washington, DC SAS Users Group

First Quarter Meeting

When:	Tuesday, March 10, 1998
Time:	8:30a.m. to noon
Place:	Bureau of Labor Statistics Postal Square Building Room G440 First Street North East Washington DC

NOTE: CHANGE OF DATE

Agenda:	8:30 - 9:00	Registration and refreshments
	9:00 - 9:15	Announcements
	9:15 - 10:00	Chris Roper, Qualex Consulting Services, Inc. ***DATA Tables: Highlighting Techniques***
	10:00 - 10:15	User-to User Questions and Answers Moderators - DCSUG officers
	10:15 - 10:30	Break
	10:30 - 11:15	Michael Raithel, U.S. Customs ***Ferreting Out Year 2000 Compliance Problems with Proc Source***
	11:15 - 11:45	Chris Roper, Qualex Conulting Services, Inc. ***Quizzer***
	11:45 - noon	Linda Atkinson, SAS Liaison ***Giveaways -*** souvenirs from NESUG97

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 3/3/98

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

★★★★★★★★★★

President's Message

Peter Knapp

Hopefully everyone has had a relaxing and restful holiday season and is ready for another year of exciting SAS challenges. For those that can make it, this year's SUGI will be held in Nashville, Tennessee on March 22-25. I am looking forward to it not only because I have never been to Nashville, but also because I always learn more about SAS and the computer profession overall. I also always learn more about SAS at our local DCSUG meetings. I hope you will attend our next one on March 10 where our featured speakers will be Chris Roper and Michael Raithel. I hope to see you there!

Speaker Abstracts

Data Tables: Highlighting Techniques

Christopher A. Roper,
Qualex Consulting Services, Inc.

Abstract

Data tables are arguably one of the best looking widgets in SAS. This makes them a popular choice for replacing the venerable Extended Table widget as a selection list, or front end to a data set. The model/viewer paradigm is an extremely powerful tool that gives the Data Table, and it's parent class the Table Editor, an extensive utility across a broad range of applications. Combined with the abundance of methods available to the Data Table class, this widget can add a very nice touch of polish to an application, giving it the professional, finished appearance a client demands and deserves. One of the most desired features of the Data Table is its ability to highlight a cell, row, or column. This paper will illustrate five scenarios for using highlighting to make the Data Table object more attractive and functional for your information consumers.

Quizzer

Christopher A. Roper,
Qualex Consulting Services, Inc

Quizzer is a fun, fast-paced Frame application game in the format of the popular TV game show Jeopardy. Up to four teams or contestants compete on their knowledge of the SAS system. Quizzer is a great way to have fun and learn new things about the SAS system, all at the same time

Christopher Roper is a Senior Systems Developer with Qualex Consulting Services, Inc. His areas of expertise are base SAS, macros, AF, Frame, and SCL. Christopher has a BS degree in Business Administration with a Concentration in Economics from Christopher Newport University. He has been a SAS user for over 10 years.

Ferreting Out Year 2000 Compliance Problems With PROC SOURCE

Michael A. Raithel, U.S. Customs

Abstract

As the new millennium rapidly approaches, numerous organizations have intensified their efforts onYear 2000 compliance projects. Many organizations have allocated significant staff time and financial resources to ensure that they find and fix their non-Year 2000 compliant applications. Predictably, a number of vendors are taking advantage of this new focus by providing packages that help to identify problematic code. However, IS shops running SAS software under MVS on mainframe servers already possess a powerful tool that can be used in Year 2000 projects.

PROC SOURCE is a Base SAS Software tool that can process MVS Partitioned Data Sets (PDS). Coupled with SAS character handling functions, PROC SOURCE can become a very powerful tool for examining PDS's for non-Year 2000 compliant entities.

This paper demonstrates how to use PROC SOURCE to process PDS's that contain source code. It also illustrates how PROC SOURCE can be used to examine load libraries for instances of specific load modules.

Michael Raithel, Computer Performance Analyst with the US Customs Service, is a well-known, award winning SUGI speaker and contributor. He is the author of ***Tuning SAS Applications in the MVS Environment****, available from the SAS Institute's Books by Users program.*

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

If you are not currently a paid member of the DCSUG and are planning to attend the meeting, please **Email** Larry Altmayer at laltmayer@census.gov or call 301-457-3859 by March 6th 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.

Desktop Special Interest Group

by David Barnes

DCSUG's Desktop Special Interest Group will meet in the Spring (Date to be announced) at Westat, Inc. 1650 Research Blvd. in Rockville at 7:30PM. If you have any suggestions for future meeting topics, speakers, or alternate locations, or if you'd like to be added to our group's mailing list so you can get announcements of upcoming meeting events, please contact David 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.

Directions 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. You can park in any non-reserved space. Signs will be posted in the lobby to direct you to the meeting room. Westat has many buildings on Research Blvd. so make sure you're at 1650.

%SYSFUNC(A Bridge ..);

by Michael Yu

In the 6.09E and 6.12 releases of SAS, there's a new macro function,

```
   %SYSFUNC( SAS_function( arguments ) <, format> );
```

which bridges regular data step functions to macro processing. In the same release, lots of SCL functions become available to DATA step programming. These two developments not only give more power but also greatly simplify macro programming. %SYSFUNC is documented in the new SAS Macro Language Reference, First Edition, January 1997. This short article gives a brief introduction to this new function and discusses several interesting applications.

The basic syntax for this routine is

```
%SYSFUNC( SAS_function( arguments ) <, format> );
```

where "SAS_function" must be either a SAS function or a function written with SAS/TOOLKIT. "Arguments" can be a macro variable or text expression. If the returned value contains special characters or mnemonic operators, such as &, %, AND, OR, EQ, then use %QSYSFUNC instead of %SYSFUNC to mask these characters. "Format" is optional and applies the specified SAS format to the value returned by the function.

For example, assume we want to include the current date as part of a title statement. We may want to use the DATA step function date() instead of the automatic macro variable &SYSDATE, because date() always returns the current date whereas &SYSDATE gives the date when the current SAS session started running.

With %SYSFUNC,

```
  title "%sysfunc( date(), weekdate18. ) Absence Report";
```

Without %SYSFUNC, it would have to be something like

```
  data _null_;
      call symput( 'date', put( date(), weekdate18. ));
  run;

  title "&date Absence Report";
  .. ..
```

Since %SYSFUNC is a macro function, you generally should not enclose character values in quotation marks as you do in DATA step functions. For example, the DATA step statement

```
      dsid= open( "sasuser.crime", "i" );
```

now becomes

```
      %let dsid= %sysfunc( open( sasuser.crime, i ));
```

You cannot nest functions within a single call to %SYSFUNC, but you can nest %SYSFUNC calls. For example:

```
  %let x= %sysfunc( trim( %sysfunc( left( &num ))));
```

The manual also lists functions not available with %SYSFUNC and %QSYSFUNC. They are DIF, INPUT, PUT, DIM, LAG, RESOLVE, HBOUND, LBOUND, and SYMGET. Although INPUT and PUT are not available, INPUTN, INPUTC, PUTN, and PUTC from Screen Control Language can be used instead. The manual also gives an example for obtaining the number of variables and observations present in a SAS data set. Without %SYSFUNC, you would need to use PROC SQL, or DATA _NULL_ with a SET statement. But, since the OPEN, CLOSE, and ATTRN functions are now available, here's a new possibility:

```
  %LET DSID= %SYSFUNC( OPEN( dataset-of-interest )); ---(1)
  %LET NOBS= %SYSFUNC( ATTRN( &DSID, NOBS ));
  %LET NVARS= %SYSFUNC( ATTRN( &DSID, NVARS ));
  %LET DSID= %SYSFUNC( CLOSE( &DSID )); ---(2)
```

It's worth noting that, in between statements (1) and (2), you can still read this open data set (e.g. in PROC PRINT), but cannot update it. An open SAS data set can stay open between PROCs.

Another example:

```
%LET DIR= /DIRECTORY/WHICH/SHOULD/BE/IN/LOWER/CASE; /* UNIX is
case-sensitive */
%LET DIR= %SYSFUNC( LOWCASE( &DIR ));
%LET DIR= %LOWCASE( &DIR ); /* Macro function abends for special character '/' */
```

It's interesting that the %UPCASE() macro function doesn't abend for '/'. The difference is that %UPCASE is a true macro function, while the %LOWCASE "function" is actually an autocall macro supplied by SAS Institute. Since %SYSFUNC readily solved this abend, I didn't bother investigating how to mask the '/' characters in the input macro variable. Note that macro functions can return more than 200 characters, while DATA step functions are limited to 200 characters.

Recently on SAS-L, a question was posted on how to obtain current SAS option settings, such as pagesize. This can be done with %SYSFUNC and the new getoption() function:

```
  %LET PS= %SYSFUNC( GETOPTION( PS, KEYWORD ));
  OPTIONS PS= new-value-for-temporary-change;
  .. ..
  RUN;

  OPTIONS &PS; /* To restore previous setting */
```

Another recent SAS-L message asked how to input datetime data into a SAS data set using PROC SQL. One attempt might be:

```
  PROC SQL;
  INSERT INTO A
  VALUES( DHMS( MDY(3,5,97),15,30,27 ), DHMS( MDY(3,4,97),10,30,27 ));
```

The above won't work, however -- since SAS is expecting numeric values, it rejects "DHMS" as characters instead of performing the desired function. Ian Whitlock contributed one possible solution, which does not use %SYSFUNC.

```
  PROC SQL;
    create table a  ( f1 numeric format = datetime16., f2 numeric format = datetime16. ) ;
    insert into a
      set f1 = DHMS(MDY(3,5,97),15,30,27), f2 = DHMS(MDY(3,4,97),10,30,27)
    ;
    select * from a ;
  quit ;
```

The problem could also be solved using %SYSFUNC as follows:

```
  PROC SQL;
    INSERT INTO A
    VALUES ( %sysfunc( DHMS( %sysfunc( MDY(3,5,97)),15,30,27)),
           %sysfunc( DHMS( %sysfunc( MDY(3,4,97)),10,30,27)));
```

%SYSFUNC is artificially invoked to produce a numeric value, as if we had used '01JAN1997'd as required by the VALUES statement.

Finally, let's see how to read a SAS data set in a macro program using %SYSFUNC.

```
%macro READ_DAT;
     %let dsid= %sysfunc( open( sasuser.houses, is ));
     %let N_style= %sysfunc( varnum( &dsid, style ));
     %do %while ( %sysfunc( fetch( &dsid )) ne -1 );
        %let style= %sysfunc( getvarc( &dsid, &N_style ));
        /* Do something based on current value of STYLE as
          in &STYLE */
     %end;
     %let rc= %sysfunc( close( &dsid ));
%mend  READ_DAT;

%READ_DAT;
```

Without %SYSFUNC, one solution could be:

```
DATA _NULL_;
  file 'file.txt';
  set sasuser.houses;
  put '/* Do something based on current value of STYLE */'
   / '/* When constructing your program logic, watch out
     UN-balanced quotes! */'
  ;
run;

%inc 'file.txt';
```

I like %SYSFUNC better because it's intuitively simple, and less cumbersome for not having to deal with quotes. Another very big advantage of the %SYSFUNC form is that it does not make a step boundary, hence the code can appear within a step, while the alternate must be placed between step boundaries.

You can also read an external file in macro with %SYSFUNC:

```
%let DIR= %sysfunc( pathname( work ));
filename data_txt "&DIR\data_txt.txt";

%let len= 8;
/* A global constant, because STYLE is 8 characters long */

data _null_;
 set sasuser.houses( obs= 3 );
     file data_txt;
     /* Make sure PRICE has different number of decimals */
     if mod( _n_, 2 ) = 0 then price= price / 10000.0;
 put style $8. +1 price;
run;

%macro READ_FIL;
   %let fid= %sysfunc( fopen( data_txt, i ));
   %do %while ( %sysfunc( fread( &fid )) eq 0 );

      /* Note the syntax, STYLE not in quotes, then  a macro variable &STYLE is automatically created */
      %let rc= %sysfunc( fget( &fid, STYLE, &len ));
      %let STYLE= %sysfunc( trim( &style ));
       /* Compute the number of bytes occupied by PRICE */
       %let rc= %sysfunc( fpos( &fid, &len + 2 ));
       %let fld_len= %eval( %sysfunc( frlen( &fid ))
                              - ( &len + 1 ));
      /* PRICE not in quotes, then its values is in &PRICE */
      %let rc= %sysfunc( fget( &fid, PRICE, &fld_len ));

      /* Do something based on current values of &STYLE and &PRICE */

    %end;
    %let rc= %sysfunc( fclose( &fid ));
  %mend  READ_FIL;

  %READ_FIL;
```

In conclusion, the new %SYSFUNC macro function is a useful way to bridge traditional macro and DATA step processing. It simplifies things we had to do without it and offers more power to traditional macro programming. Make %SYSFUNC part of your tool set, because %SYSFUNC is worth it!

Hsiwei Yu (Michael) works for Northrop Grumman Corporation, on site at Environmental Protection Agency in downtown Washington, D.C. Previously with Legent and SAS, Michael has been a SAS programmers since '87. His current assignment is to analyze EPA's corporate data and implement user-friendly dynamic queries on the Internet.

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

DCSUG Membership Directory

DCSUG is considering publishing a membership directory in 1997, containing members name, work affiliation (name of institution), work telephone number, e-mail address, and any areas of expertise that you would be willing to identify. This directory would be provided to DCSUG members only and would not be given to outside sources. If you would like to be included in such a directory please mark the boxes below, as appropriate.

- □ I would like to be included in the membership directory DCSUG is considering publishing. Include my name, work affiliation, and work telephone number.
- □ Include my e-mail address as shown on this form.
- □ Include the following areas of expertise (up to 2 products, platforms).

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

SAS Talk

By Ian Whitlock

Is SAS SQL fast or slow? Consider a fictitious argument that I over heard between two of my colleagues. I will refer to them as Dr. Fast and Mr. Slow to help remind you of their conclusions about SQL. They were discussing a file JOBS. Two of the variables were PERSID (a key to persons) and JOBSID (the key to jobs held by that person). For example:

PERSID	JOBSID
11	11
11	12
11	13
12	14
13	15
13	16
....	

Mr. Slow said, "I ran into a nasty problem the other day. I wanted to add a new variable to the JOBS file giving the originally assigned JOBSID to each record with the same PERSID. Of course there was a lot more to it, but the details don't matter. After much agonizing, because at the client's request that I had to use pure SQL code, I developed some nifty SQL to solve the problem. You know I just love subqueries. Unfortunately SQL is so slow that it ran for hours."

Dr. Fast asked for an explanation of the code. Mr. Slow responded, "Well, since it is complex, let me simplify the task for you - just subset JOBS to the first record in each group of PERSID. That, of course would be easy with FIRST dot processing in a DATA step, but remember I am limited to SQL. I have found an important method of counting using a subquery to solve this problem. Consider:

```
1  select a.jobsid, a.persid,
2        ( select count(b.jobsid)
3            from jobs as b
4            where (b.jobsid  <= a.jobsid)
5              and (b.persid = a.persid
6         ) as rank
7     from jobs a
8     where calculated rank = 1 ;
```

Dr. Fast automatically responded, "But an important principle of SQL is that you cannot assume any order to the processing of the file, so how can you guarantee that the count is always made in the desired order?"

"Well, line 5 insures that we are looking at records in the same group, and line 4 shows that we are counting the specific records where the JOBSID is less than or equal to a given JOBSID. The first time we count the lowest value of JOBSID, the next time the two lowest values of JOBSID, and so on. So you can see that there is no assumption of order to my method", replied Mr. Slow. Dr. Fast, who is really quite slow to understand some things, muttered something about people who use subqueries and returned to his office to think about the problem.

Lines 2 through 6 form a correlated subquery because the subquery's where condition in lines 4 and 5 refers to the outer query. This means that the SQL optimizer cannot process the inner query firstIn fact it must form the subset for each record in the outer query. Consequently correlated subqueries can be notoriously slow.

I soon heard Dr. Fast exclaim, "Aha! He just wants the minimum JOBSID in each group of PERSID. I wonder why he didn't simply use:

```
select persid , min(jobsid) as origid
   from jobs
   group by persid ;
```

Now Dr. Fast's code involves the calculation of a statistic, but he eliminated the correlated subquery. Surely his code would execute much faster than Mr. Slow's code. To investigate how much faster he generated some test data with the code:

```
%let npers = 10000 ;

data jobs ;

   drop temp i ;

   do persid = 1 to &npers ;

      temp = ceil(ranuni(8573124)*30) ;

      do i = 1 to 5 ;
         jobsid = put ( temp + i , z8. ) ;
         subtype = round ( ranuni ( 8573124 ) + .2 ) ;
         output ;
      end ;
   end ;
run ;
```

The macro variable NPERS allowed him to control the amount of test data easily. First he tried a small size, since correlated subqueries tend to bog down with size, and then he set NPERS to 10,000. He never tried a large number. Here is the log that he showed me.

```
16   proc sql stimer ;
NOTE: The SQL Statement used 0.16 seconds.

17
18     create table mins3 as
19       select persid , min ( jobsid ) as minjobid
20         from JOBS
21         group by persid
22       ;
NOTE: Table WORK.MINS3 created, with 10000 rows and 2 columns.

NOTE: The SQL Statement used 1.81 seconds  .

23
24     create table mins1 as
25       select a.persid , a.jobsid
26         from JOBS as a
27         where ( select count(b.persid)
28                   from JOBS as b
29                   where (b.jobsid <= a.jobsid)
30                     and (b.persid = a.persid)
31               ) = 1
32     ;
NOTE: Table WORK.MINS1 created, with 10000 rows and 2 columns.

NOTE: The SQL Statement used 2 hours 12 minutes 36.02 seconds.

33   quit;
NOTE: The PROCEDURE SQL used 0.0 seconds.
```

I had no idea the discrepancy could be so great. Dr Fast replied, "Yes, the ratio of times is a factor greater than 4,000. No wonder Mr. Slow is complaining about the speed of SQL. Perhaps he will be relieved to know that it is only his code and not the language. The difference in times here reminds me of the time when I removed 30,000 steps from some procedural macro code. I wonder if the genie will always do what you tell it to."

The lesson here is that even with a nonprocedural language like SQL it is sometimes important for the programmer to understand what he is asking the computer to do.

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 1998

10 **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:
Linda Atkinson at (202) 694-5046
atkinson@econ.ag.gov
Peter Knapp at (202) 482-1359
peter_knapp@ita.doc.gov

22-25 **SUGI 23** SAS Users Group International
Opryland Hotel
Nashville, TN
Contact:
Sally Goostrey at (616) 833-8877
sagoostr@am.pnu.com

June 1998

2 **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: TBA
Contacts:
Linda Atkinson at (202) 219-0934
atkinson@econ.ag.gov
Peter Knapp at (202) 482-1359
peter_knapp@ita.doc.gov

Other Groups

CENSUG Census SAS Users Group
Contact: Rick Denby at (301) 763-8174

Client/Server SAS Users Group
Contact: Tom Skinner at 916-356-4485
thomas_m_skinner@ccm.hf.intel.com

CONSUG Consultants SAS Users Group
Contact: Eric Brinsfield at (919) 518-1070
merecb@meridian-software.com
www.meridian-software.com/consug/consug.htm

MDSUG Maryland SAS Users Group
Contact: Mr. R.H. Miller, (410) 740-4229
refa09a@prodigy.com

NCHSSUG National Center for Health Statistics SAS Users Group
Contact: Arlene Siller at (301) 436-8522
abs2@cdc.gov
or Linda Tompkins at (301) 436-7022

NESUG NorthEast SAS Users Group
Contact: Ray Pass, 203-356-9540
raypass@worldnet.att.net
http://sasuser.pace.edu/nesug/

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

PHARMASUG Pharmaceutical Industry SAS Users Group
Contact: Ms. Susan Haviar, 619-223-7560
www.pharmasug.org

SESUG SouthEast SAS Users Group
Contact: Frank DiIorio at 919-942-2028
fcdiiorio@aol.com
www.uga.edu/~sesug/

SAS Programmer Needed

Individual needed for junior to mid-level SAS programming in a Windows environment. Will work with diverse types of data (e.g., survey data, large MIS databases) under the direction of senior staff on research projects in the area of human resources. The individual will be responsible for creating and manipulating data and generating statistical output. BA in social science with at least one course in statistics and experience in data handling and basic statistical techniques. Minimum 2 years experience using SAS BASE and STAT. Experience with mainframe SAS, Access, and Visual Basic a plus. Salary $30,000 to $40,000 per year, commensurate with skills and experience.

Please send resume to:

Personnel

Humrro

66 Canel Center
Suite 400
Alexandria, VA 22314
703-549-3611

SAS Opportunities

* SAS Programming	*Database Management
* Biostatistics	*Direct Market Modeling
* Marketing Science	*Promotion Response Modeling

For local and nationwide positions call, fax or e-mail your resume and criteria to:

M.T. Ray	* T 203 221-8939
256 Post Road East	* F 203 221-8938
Westport, CT 06880	* mtray@smihan.com

Offices in: New York*Chicago*Connecticut*Philadelphia*Atlanta

SMITH HANDLEY

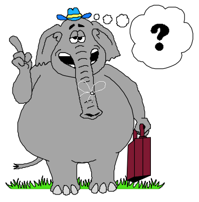
1998 Meetings

Plan ahead and mark your calendars now. The DCSUG will meet on the following dates in 1998: June 2, September 15, December 1. 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/

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



Remember to Join DCSUG!!!

Don't forget to renew your DCSUG membership for 1998. 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.