

The logo for HASUG FLASH is a black, jagged, lightning-bolt-like shape pointing to the right. The words "HASUG" and "FLASH" are written in white, bold, sans-serif capital letters across the shape. "HASUG" is positioned above "FLASH".

HASUG FLASH

HARTFORD AREA SAS® USERS GROUP

Volume 10, Issue 4

Fourth Quarter, November, 1999

Table of Contents

Note from the Chairman	2
Meeting Announcement.....	2
Continuing with Education	3
Steering Committee Members	3
Recruiting Policy	4
Sponsorship Policy.....	4
Consultant Special Interest Group (CONSIG).....	4
August Meeting Highlights	4
Editorial and Production Staff	5
Pictures from August Meeting	5
Free Manuals at August Meeting.....	6
TIP 00081, convert SAS data to SPSS	6
TIP 00083, convert fraction to floating point numeric	7
TIP 00084, convert variables to IN operator format	7
TIP 00085, rewriting Merge statement	10
TIP 00086, convert from Fahrenheit to Celsius	11
TIP 00087, writing dates in DD-MMM-YY format	11
TIP 00088, look-up table merge	11
TIP 00089, easy way to rename variables in sequence.....	13
TIP 00090, capture and setting back option settings	13
TIP 00091, selectively process SAS data sets	14
TIP 00092, simulate PROC HIER.....	15
Directions to August Meeting	17
Hyperlinks of Interest	17
Membership Form	18

The HASUG Flash is available for reading on the World Wide Web at <http://www.hasug.org>
SAS is a registered trademark of SAS Institute Inc., Cary, North Carolina

Note from the Chairman

Miguel Hernandez
PRT Group, Inc.

Once again HASUG continues to evolve. This is our first totally electronic edition of the HASUG Flash. No more trying to come up with the money for duplicating and mailing. No more labels, printing, and postage. Because of the new format we are no longer limited to ten pages.

We can now make available all the tips Chuck "Birdman" Patridge can find. We can aggressively push all our HASUG members to write articles to be printed. Who knows, writing an article or a book review could be a stepping stone towards presenting a paper at HASUG, NESUG, or SUGI. What better way to network and get your name out into the market place. If you would like help putting together an idea with which you have been toying, give one of the committee members a call or email. He/she will be able to help you get started in the right direction.

Our last quarterly meeting held at Destiny had a very good turn out of fifty to sixty people. On November 18th we will be having our last meeting of the year (century) in Waterford, CT. It will be the first time HASUG has met in that part of the state and I am very optimistic about the attendance. We hope to see many of our fellow SAS programmers from eastern Connecticut and even some people from Rhode Island. If you have SAS friends in that area, please extend an invitation to them. Let them know that we welcome everyone at our quarterly meetings.

If your company is interested in hosting one of our meetings, please let one of the committee members know and we will work with you and your company to set it up. As always, we are in search of ideas for topics. Any suggestions you have can be sent to any committee member or me. We will endeavor to see your suggestion become a reality.

This being the last Flash of 1999, on behalf of the HASUG steering committee, I wish everyone a Happy and Healthy Holiday season. May Y2K be good to you.

HASUG MEETING ANNOUNCEMENTS

The next HASUG meeting is to be held on Thursday, November 18, 1999, 9:30 to noon at Waterford Library in Waterford, CT. It is hosted by Kendle. Make note of the change from our usual start time. Doors at the library will open at 9:00a.m. Please pre-register for the meeting by sending an email to James Alling at alling.jamest@kendle.com. See later in this newsletter for directions.

Our first presentation will be by Janet Stuelpner. Janet will present an "Introduction to the Report Window". Although it appears that PROC REPORT can be complicated at times and difficult to master, there is an easy way to get started. Through the display manager, you can point and click your way to beautiful reports. The great thing about this method is that you can save the code and rerun it at any time. This technique can save the report veteran a great deal of time as well. It allows you to start to create all of the DEFINE statements without typing a word. This point and click method is a nice way to learn how to create interesting reports. Janet will demonstrate the different ways to bring up the report window and how to use it effectively to create the reports that your management wants to see.

Janet Stuelpner provides programming support in the areas of clinical trials and the financial industry. She also provides both individual and group technical training when needed at client sites. Janet has several degrees in the sciences. She has been a SAS user for 18 years on many different platforms. Originally a systems programmer, she has now turned her focus to applications programming and teaching. She is a section chair at NESUG and has been a presenter at users groups of all levels.

Our second presentation will be a tutorial given by Sue Freimuth, based on a paper by Chris Yindra. The tutorial will introduce the SQL (Structured Query Language) procedure through a series of simple examples. It will initially discuss choosing variables (SELECT) from SAS® data sets (FROM) where a specific criteria is met (WHERE). Next, calculating and formatting values will be explained. Once the basic SQL syntax has been covered more advanced features of SQL will be presented such as grouping and ordering data, selecting based on summary values, applying CASE logic and simple joins. Finally a comparison will be made of simple SQL queries to base SAS. This tutorial will provide attendees with all the tools necessary to write simple SQL queries. It is intended for SAS programmers who have had no prior exposure to the SQL procedure as well as those new to SAS.

Sue Freimuth is a principal in CY Associates Inc., a consulting firm specializing in reporting solutions for the healthcare and credit risk management industries. She was first introduced to SAS software over 10 years ago, and has been working extensively with it ever since. She has presented papers on various aspects of the SAS system at SUGI, and other SAS user group meetings.

Continuing With Education...

Peter Prause, The Hartford

Local SAS training is available at Destiny Corporation, of Wethersfield, CT. Below is a list of their scheduled classes through June 2000. You can reach Destiny at 1-800-787-2464 or at 860-721-1684, or send email to info@destinycorp.com (website www.destinycorp.com).

Fundamentals of the SAS System.....Nov 15 – 17
 Report Writing In SAS Nov 18 – 19
 Programming in SAS Software.....Dec 6 – 8
 Advanced SAS Prgmng Techniques.....Dec 9 – 10
 Getting Started with SAS V8Jan 5 – 7
 Fundamentals of the SAS System.....Jan 10 – 12
 SAS V8 Changes and Enhancements.....Jan 13 – 14
 Macros in SAS Software.....Feb 7 – 9
 Report Writing In SAS Feb 10 – 11
 Programming in SAS Software.....Mar 6 – 8
 Advanced SAS Prgmng Techniques.....Mar 9 – 10
 Fundamentals of the SAS System ch.....Mar 13 – 15
 SQL Processing with the SAS System.....Mar 16
 The Output Delivery System.....Mar 17
 SAS V8 Changes and Enhancements.....Apr 20 – 21
 Macros in SAS Software.....May 8 – 10
 Report Writing In SAS .ach..... May 11 –13
 Web Publishing with SAS Output.....May 22 – 23
 Getting Started with SAS V8May 24 – 26
 Fundamentals of the SAS System.....Jun 5 – 7
 Advanced SAS Prgmng TechniquesJun 8 - 9

The SAS Institute Regional Training Center in Glastonbury, CT. is offering these courses through June 2000 also. Course information for these courses and many other non-local courses is available at the www.sas.com website. To register for a course or to get more information, phone SAS Institute at 1-800-333-7660.

SAS Fundamentals.....Nov 17 - 19
 SAS Macro Language.....Dec 2 – 3
 SQL Processing with the SAS System.....Dec 15 - 16
 Processing with SAS/ACCESS..... Dec 17
 SAS Fundamentals.....Jan 12 – 14
 SAS Programming.....Feb 2 – 4
 Advanced SAS Prgmng TechniquesFeb 17 – 18
 Getting Started with SAS Software.....Feb 24 – 25
 New Features in Version 8.....Mar 9 – 10
 SAS Programming I: Essentials.....Mar 29 – 31
 SAS Report Writing.....Apr 6 – 7
 SAS Programming II: Manipulating Data.....May 3 – 5
 SAS Macro Language.....Jun 8 - 9
 Processing with SAS/ACCESS.....Jun 23

If your organization has need for training which is not scheduled, please speak to a Steering Committee Member about it.

STEERING COMMITTEE MEMBERS

Name	Organization	Telephone	E-Mail
James Alling	Kendle	(860) 434-4950x116	alling.jamest@kendle.com
Robert Clauss	Fleet Bank	(860) 986-3878	robert_g_clauss@fleet.com
Michael Davis	Bassett Consulting	(203) 562-0640	michael@bassettconsulting.com
Miguel Hernandez	PRT, Inc.	(860) 687-2262	mhernandez@prt.com
Cindy Johnson	Bristol-Myers Squibb	(203) 677-6922	johnsonc@bms.com
Karol Katz	Yale University	(203) 764-4338	Karol.Katz@yale.edu
Rob Krajcik	Bristol-Myers Squibb	(203) 677-6125	Robert.Krajcik@bms.com
Rene Laurencot	Pfizer	(860) 228-4786	renehana@gateway.net
Albert Lavoie	Bristol-Myers Squibb	(203) 677-6166	lavoiea@bms.com
Barbara Moss	CIGNA	(860) 726-1504	Barbara.Moss@CIGNA.com
Chuck Patridge	The Hartford	(860) 547-6644	Charles_S_Patridge@prodigy.net
Peter Prause	The Hartford	(860) 547-7484	Peter.Prause@thehartford.com
Janet Stuelpner	consultant	(212) 513-3452	stuelj@pfizer.com
Gerald Zuckier	CHIME	(203) 624-0940	GZuckier@alum.mit.edu

HASUG Recruiting Policy

HASUG's primary mission is to provide a forum for SAS professionals to meet and share experiences. HASUG also recognizes that searching for new employment opportunities is a normal activity when professionals meet. Since we depend on our respective employers to support HASUG by providing time for members to attend, prepare presentations, and provide meeting facilities, we do not wish to jeopardize those relationships we have all nurtured.

Therefore, when attending any of our quarterly meetings, we request that all parties engaged in recruiting activities be "professional and discreet", and suggest that such activities be carried on outside and after our meetings. Those seeking or offering positions may display materials at a location designated by HASUG at each quarterly meeting. The materials may include resumes, job openings, contract opportunities, business cards, etc. None of the materials may include salary information. All such materials left after the meeting will be discarded. Violators of this policy may be asked to leave the meeting.

For other recruiting opportunities, please see the SAS CONSIG article on this page or visit www.sasconsig.com.

HASUG Sponsorship Policy

All parties are encouraged and welcome to attend our quarterly HASUG meetings scheduled during the months of February, May, August, and November of each year. Usually, these meetings are scheduled to take place on the 3rd Thursday from 9:00 am to noon at various locations throughout Connecticut. Past locations have included Hartford, Meriden, Middletown, Norwalk, Glastonbury, and New Haven. We are always looking for new hosts within the state of Connecticut. If you feel your organization can host a meeting of up to 70 or so people, please contact one of the Steering Committee members.

SAS CONSIG

The SAS Consultants Special Interest Group (SAS CONSIG) operates and maintains a web site www.sasconsig.com. This site lists employment opportunities, contracts, vitas, resumes, and companies seeking SAS professional talent, and is updated weekly. You are encouraged to visit that web site or to contact Charles Patridge (Email: Charles_S_Patridge@prodigy.net) for further information.

August HASUG Highlights

Rob Krajcik
Bristol-Myers Squibb

The August 19th HASUG meeting was held at Destiny Corp. in Wethersfield.

The first speaker was Mike Zdeb from the New York State Department of Health in Albany, NY. An experienced instructor, Mike held everyone's attention while he presented "An Introduction to Macro variables". Mike started off by saying that users determine how complicated something is by whether it has a separate manual. Macros do, but it is not as complicated as one might think.

Reasons for using macros:

- avoid repetitious code
- create generalizable & flexible code
- pass information between steps
- conditionally execute data & proc steps
- dynamically create code

Mike also discussed what really happens with macros and the compiler versus what happens with regular SAS code, and noted that Macro resolution takes place before execution time.

Debugging help:

- options: symbolgen mprint mlogic
- %put _all_ (all macro variables)
- %put _automatic_ (created at session start)
- %put _local_ (macros known only to that data step)
- %put _global_ (macros in the Global Symbol Table)
- %put _user_ (macros defined in the code, both _local_ and _global_)

Tips:

- Use %eval for integer arithmetic
- Neat trick for commenting out code:
%let cm=*; /* if you want to comment out the code */
%let cm=; /* if you don't */
.
lots of SAS code
.
&cm PROC PRINT DATA=BIGDATA;
&cm RUN;
&cm PROC FREQ;
Etc.
- Macro statements can be commented out as well:
%&cm.put _user_;

(Continued on page 5)

(August Highlights, Continued from page 4)

Dana Rafiee of Destiny Corporation gave the second presentation. He gave a concise overview of Web Publishing using SAS® Software. He began with a description of the differences between running local SAS and running SAS on a web server.

Static HTML reports can be created running local SAS. Software requirements include: a browser (IE 4.0 or Netscape 4.x), SAS version 6.12 at TS055 maintenance level, and web macros to create HTML code: %ds2htm, %tab2htm, and %out2htm. These macros are available on the SAS website (WWW.SAS.COM), or as part of versions 7 and 8, under the Output Delivery System (ODS). Dana showed how to use these macros.

Dynamic HTML reports can be created using SAS/Intrnet and a web server, along with the above software. Dana showed how an HTML form is passed to the web server, which invokes the application broker. The location of the broker Common Gateway Interface (CGI) program is provided by the ACTION attribute of the HTML form or by the HREF attribute on a hypertext link. The broker prepares fields such as text areas and check boxes for conversion to SAS macro variables, which enables the SAS program to process the information. With SAS/Intrnet, the broker sends the data via file i/o and launches the application server (a SAS session). The application server runs on the web server machine and is terminated when the request is done.

The same three web macros used in local SAS above are used here with two changes: "HTMLFREF=_WEBOUT" and "RUNMODE=S" (for server).

Dana also discussed running the web server on the same laptop or desktop the SAS/Intrnet software runs. He prefers Apache Web Server to Microsoft's Personal Web Server software because "Apache Web Server is less likely to crash than Personal Web Server."

Pictures from August Meeting



Destiny's Office in Wethersfield



Mike Zdeb



Dana Rafiee

Editorial and Production Staff:

Editor	Barbara Moss
Layout	Michael Davis
Member Lists	Karol Katz
	Charles Patridge
Photographs	Charles Patridge
Contributions	Miguel Hernandez
	Peter Prause
	Charles Patridge
	Rob Krajcik
	Albert Lavoie
	Barbara Moss

FREE Manuals !!!

A number of SAS manuals and technical reports have been made available by some of our members. Attendees of the next meeting will be able to take home books for FREE! Most of the manuals are from Version 6 of the SAS software. Most of these pre-owned books are in good shape but a few new manuals and CD-ROMs will be made available. Come to the meeting to see how you might add something from the list - and others - to your library.

Technical Reports: U-113, U-112, R-108, P-256, P-252, P-251, P-243, P-242, P-229, P-222, P-221, P-202, P-197

Course Notes: Report Writing, Building SCL Applications, Fundamentals of the SAS System

SAS / IML – Usage and Reference
SAS / ETS – User's Guide
SAS Screen Control Language Reference
SAS / STAT User Guide, Volume 1 & 2
Getting Started with the Frame Entry, Developing Object-Oriented Applications

The Little SAS Book
Solutions @ Work: SAS/IntrNet Examples (CD-ROM)

TIP 00081

I am an SPSS user and I am attempting to move data from a SAS file into a file with ASCII data to be used with SPSS. How do I convert the SAS file so that I may use SPSS?

A solution by:

Glenn Curtiss, Ph.D.
Neuropsychologist
Defense/Veterans Head Injury Program PM&RS (117)
J. A. Haley VA Medical Center
13000 N. Bruce B. Downs Blvd.
Tampa, FL 33612
Voice: 813-972-2000, ext.-7118
Fax: 813-978-5852
Preferred email: gcurtiss@msn.com
Alternate email: curtiss.glenn@tampa.va.gov

For those of you who would like a complete example of exporting a SAS data file to an SPSS portable file and then bringing it in to SPSS, see the following column.

Scenario:

Assume you are working with two SAS data sets that contain different information collected on the same set of

individuals or cases. Data set "DEMO" contains basic demographic information like gender, age, education, etc. Data set "MRI" contains results of a neurological imaging procedure. However, not all individuals in the DEMO data set underwent MRI. You need to join the DEMO data set variables with the MRI variables and then export this new MRI data set to SPSS. The SAS data sets all reside in the directory 'c:\projects\neuro\data' and you are going to export the SPSS portable file to the 'c:\projects\neuro\data\spss' directory. The SPSS portable file will be named "mri_demo.por".

The following LIBNAME statements are necessary to define the paths and the portable SPSS data set. Please note the explicit naming of the export file within the destination directory and the use of the SASV5XPT export option on the LIBNAME statement. Again, if this were being run on a mainframe, you would specify XPORT instead of SASV5XPT at that location on the LIBNAME statement.

```
LIBNAME in_sas 'c:\projects\neuro\data';  
LIBNAME out_spss sasv5xpt 'c:\projects  
\neuro\data\spss\mri_demo.por';
```

```
*/This section merges the two SAS data-  
sets and retains the matched cases.*/;
```

```
PROC SORT  
    data = in_sas.mri out = mri_srted;  
BY subject;  
RUN;
```

```
PROC SORT  
    data = in_sas.demo out = demosrted;  
BY subject;  
RUN;
```

```
DATA matched;  
MERGE mri_srted (in = in_mri)  
    demosrted (in = in_demo);  
BY subject;  
IF in_mri & in_demo then OUTPUT;  
RUN;
```

```
*/ The following section writes the SPSS  
portable file. */;
```

```
DATA out_spss.mri_demo;  
SET matched;  
OUTPUT;  
RUN;
```

```
*/ Note: Writing the SPSS portable file  
could have been accomplished in the data  
step that merges the sorted files by sim-  
ply replacing the statement: DATA  
matched; with:
```

(Continued on page 7)

(SASTIP 00081, Continued from page 6)

```
DATA out_spss.mri_demo;
/*;
```

Once the SAS job has been run and the SPSS portable file generated, the following statement is used in SPSS to access the file:

```
GET SAS /DATA='c:\projects\neuro\data
\spss\mri_demo.por'.
EXECUTE.
```

TIP 00083

Solution By:

Andrew J.L. Cary
Chief Curmudgeon
Cary Consulting Systems
<http://www.caryconsulting.com>

Using an input statement, how do you read a fraction and convert it to a floating point numeric? There is a format to convert a numeric to fraction (FRACT.w) but I couldn't find anything to convert a fraction to numeric.

I can't think of a way to do this except by parsing a string. This is however quite easy...

```
a="123/32456";
X=input(scan(a,1,'/'),best16.) /
input(scan(a,2,'/'),best16.) ;
```

TIP 00084

```
/******/
/**Tip00084.sas ***/
/** Author: Charles Patridge ***/
/** *** ***/
/** Read a SAS Character Variable which contains a ***/
/** list of 3 character words and convert to a macro ***/
/** variable which takes these words and surrounds ***/
/** each one with quotes, in order to be used with ***/
/** the "IN" operator as part of a where clause ***/
/** *** ***/
/** SAS MACRO variables can contain up to 32K of data***/
/** unlike a maximum of 200 characters for a SAS var ***/
/******/
```

The SAS variable Mapping (length \$200) equals

```
001 002 003 004 005 006 007 008 009 010 011 012 013 014 015
016 017 018 019 020 021 022 023 024 025 026 027 028 029 030
031 032 033 034 035 036 037 038 039 040 041 042 043 044 045
046 047 048 049
```

The SAS variable Mapping2 (length \$200) equals

```
050 051 052 053 054 055 056 057 058 059 060 061 062 063 064
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 091 092 093 094
095 096 097 100
```

The SAS Variable Mapping3 (length \$200) equals

```
101 102 103 104
```

```
1
2 %MACRO LISTVARS(PREFIX, START, NUMBER, PRE= );
3 /** GENERATE LIST OF MACRO VARS ***/
4 %LOCAL J ;
```

(Continued on page 8)

(TIP 00084, continued from page 7)

```
5      %DO J=&START %TO &NUMBER; &PRE&PREFIX&J %END ;
6      %MEND LISTVARS;
7
8      %global mapping mapping2 mapping3 ;
9      data commontb;
10     set sasuser.commontb;
11     if custseg = '0F4';
12     call symput('mapping' , mapping );
13     call symput('mapping2', mapping2);
14     call symput('mapping3', mapping3);
15     run;
```

NOTE: The data set WORK.COMMONTB has 1 observations and 24 variables.

MAPVAR is the name of MACRO VARIABLE you wish to create

MAXNO is the maximum number of words contained in MAP

MAP is the character string that contains a list of words to be
parsed and quoted for each word within the string

```
16
17 %macro charmap(mapvar, maxno, map ) ;
18 %global &mapvar ;
19 data _null_ ;
20 %do ii = 1 %to &maxno ;
21     map = scan("&map", &ii, ' ' ) ;
22     if map ne ' ' then do ;
23         cnt + 1 ;
24         map&ii = "" || trim(map) || "" ;
25         call symput( "map&ii", trim(map&ii)) ;
26         call symput( "mapnum", put(cnt, best.) ) ;
27     end;
28 %end;
29 run;
30
31 %let &mapvar = %listvars( map, 1, &mapnum, pre=& ) ;
32 run;
33
34 %put &&&mapvar ;
35 %mend charmap ;
36
```

The Results of each call to CHARMAP

```
37 %charmap( mapvar1, 50, &mapping ) ;
'001' '002' '003' '004' '005' '006' '007' '008' '009' '010'
'011' '012' '013' '014' '015' '016' '017' '018' '019' '020'
'021' '022' '023' '024' '025' '026' '027' '028' '029' '030'
'031' '032' '033' '034' '035' '036' '037' '038' '039' '040'
'041' '042' '043' '044' '045' '046' '047' '048' '049'

38 %charmap( mapvar2, 50, &mapping2);
```

(TIP 00084, continued on page 9)

(TIP 00084, continued from page 8)

```
'050' '051' '052' '053' '054' '055' '056' '057' '058' '059' '060'  
'061' '062' '063' '064' '065' '066' '067' '068' '069' '070' '071'  
'072' '073' '074' '075' '076' '077' '078' '079' '080' '081' '082'  
'083' '084' '085' '086' '087' '088' '089' '090' '091' '092' '093'  
'094' '095' '096' '097' '100'
```

```
39 %charmap( mapvar3, 50, &mapping3);  
'101' '102' '103' '104'
```

Concatenating the Created Macro Variables can now be used with the IN operator as part of a WHERE clause.

```
40
```

```
41 %put &mapvar1 &mapvar2 &mapvar3 ;
```

```
'001' '002' '003' '004' '005' '006' '007' '008' '009' '010'  
'011' '012' '013' '014' '015' '016' '017' '018' '019' '020'  
'021' '022' '023' '024' '025' '026' '027' '028' '029' '030'  
'031' '032' '033' '034' '035' '036' '037' '038' '039' '040'  
'041' '042' '043' '044' '045' '046' '047' '048' '049' '050'  
'051' '052' '053' '054' '055' '056' '057' '058' '059' '060'  
'061' '062' '063' '064' '065' '066' '067' '068' '069' '070'  
'071' '072' '073' '074' '075' '076' '077' '078' '079' '080'  
'081' '082' '083' '084' '085' '086' '087' '088' '089' '090'  
'091' '092' '093' '094' '095' '096' '097' '100' '101' '102'  
'103' '104'
```

Example :

```
IF CUSTSEG IN ( &mapvar1 &mapvar2 &mapvar3 ) ;
```

where CUSTSEG is a character variable.

=====

A better and more elegant solution for same problem provided by Ian Whitlock <whitloi@westat.com>

```
/** generate a list of blank separated numbers for testing ***/  
%macro gendata ( n = 1 ) ;  
  %global list ;  
  %local i ;  
  %let list = ;  
  %do i = 1 %to &n ;  
    %let list = &list %sysfunc(putn(&i,z3.)) ;  
  %end ;  
%mend gendata ;  
  
/* Put quote marks around each word in a space separated list */  
%macro quoteit ( list = , /* list of space separated tokens*/  
  quotemark = %str("%") /* type of quote mark to use*/);  
  
/* -----  
MODULE: QUOTEIT  
  
PURPOSE: Aid in coding when each token in a list should  
be enclosed in quotes.tc
```

(Continued on page 10)

(TIP 00084, continued from page 9)

CLASS: Macro function returning quoted list.

USAGE: %let wanted = 0001 0023 0432 0589 1200 ;
proc print data = w (
 where =(basid in (%wquoteit (list = &wanted)))) ;

PARAMETERS: R LIST = List of space separated tokens.

QUOTMARK = %STR("%") Type of quote mark to use.

SIDE EFFECTS: None.

NOTES:

SYSTEMS: All (tested WIN).

HISTORY: Part of the original CATISAS macro system IW

DOCUMENT: Here.

SUPPORT: Ian Whitlock <whitloi1@westat.com

*/

```
%local i word newlist ;
%let i = 1 ;
%let word = %scan ( &list , &i , %str( ) ) ;
%do %while ( %length(&word) 0 ) ;
  %let newlist = &newlist %unquote(%str( &quotmark&word&quotmark )) ;
  %let i = %eval ( &i + 1 ) ;
  %let word = %scan ( &list , &i , %str( ) ) ;
%end ;
&newlist
%mend quoteit ;

%gendata(n=105) ; /*** generate a list of 21 numbers ***/
%put list=&list ; /*** display this list to see it ***/

%let list = %quoteit(list=&list) ; /*** now quote this list ***/
%put list=&list ; /*** now display quoted list ***/
```

TIP 00085

```
/* **** */
/** TIP00085 ***/
/** Solution by:
    Paul M. Dorfman
    Jacksonville, FL <sashole@earthlink.net
    rewriting merge/data statement ***/
```

Problem:

Is there a more concise way to write the following ?:

```
DATA Z_BH_L;
  MERGE BUY_L1 BUY_L2 BUY_L3 BUY_L4 BUY_L5
        BUY_L6 BUY_L7 BUY_L8 BUY_L9 BUY_L10 BUY_L11
        BUY_L12 BUY_L13 BUY_L14 BUY_L15 BUY_L16 BUY_L17
        BUY_L18 BUY_L19 BUY_L20 BUY_L21 BUY_L22;
  by sasyear; run;
```

(TIP 00085, continued on page 11)

For example;

```
Data Z_BH_L;
MERGE BUY_L(1-22);
by sasyear; run;
```

=====

That would be great. Unfortunately, we are limited to something like

```
%macro mrglist(out=,by=,prefix=,from=1,to=);
DATA &out;
MERGE
%do i=&from %to &to; &&prefix&i %end;;
by &by;
run;
%mend mrglist;
```

TIP 00086

Solution by: Fabrizio
Email: Fabrizio1@usa.net

Are there any functions or formats to convert temperature from C to F or vice versa without having to right any code ? Not in SAS. You have to write it manually, and pick the right choice whether you convert temperatures or temperature intervals.

```
Intervals:
C = F * 5 / 9;
F = C * 9 / 5;
Temperatures:
C = (F - 32) * 5 / 9;
F = C * 9 / 5 + 32;
```

TIP 00087

Author: Charles Patridge
Email: Charles_S_Patridge@prodigy.net
Web: www.sasconsig.com

Is there a way to write dates in format DD-MMM-YY?
For example:

08/27/99 = 8-AUG-99 in version 6.12?

```
data test;
today = put( today(), worddatx.);
/** dd Mmmmmmmm yyyy */
fmtdate = scan( today, 1) || '-' ||
/** get day */
```

```
substr( scan( today, 2), 1, 3) ||
'-' || /** get month */
substr( scan( today, 3), 3, 2);
/** get year */
run;
```

Author: Chris Jones
Respond to: chris@telespan.demon.co.uk
Web: http://www.telespan.demon.co.uk

```
data _null_;
dn=today();
dd=day(dn);
mm=month(dn);
yy=year(dn);
put dd 2. "-" dn monname3. "-" yy;
run;
```

TIP 00088 Lookup Table Merge

I have two datasets. Data Set ONE is small (200 obs), while Data set TWO is huge (40 million obs). I only want to extract the records from Data TWO that match on the ID in the smaller table. I'm seeking the method that requires the least amount of resources. Merge doesn't seem to be it. Any suggestions?

(TIP 00088, continued on page 12)

(TIP 00088, continued from page 11)

A Solution by
Author: Larry Kemp
Email: larryk@airlinedata.com

This is the way I handle those situations.

```
FILENAME TEMP 'c:\format.SAS';

Data ONE;
infile 'xfile';
input @1 ID $4.;

DATA _NULL_;
  SET one END=EOF;
  FILE TEMP;
  YES = 'Y';
  IF _N_ = 1 THEN DO;
    PUT @1
      'PROC FORMAT LIBRARY=LIBRARY;
      VALUE $fmt';
  END;
  PUT @1      ""
      @2      id      $char4.
      @6      "'=' "
      @9      yes $char1.
      @10     ""      ;
  IF EOF THEN PUT @1
    "OTHER = ' ';" ;
RUN;

OPTIONS NOSOURCE2;

%INCLUDE TEMP1;
RUN;

Data TWO;
infile 'yfile';
input @1 ID $4.
      @5 Var1 $5.
      @10 Var2 $5.;
if put(id,$fmt.) = 'Y';
run;
```

A Solution by
Author: Jacksen Lou
Email: earthroads@aol.com

Most data manipulations in SAS can be easily solved.
There is a simple way to do it by using the PROC SQL
sub-query.

```
proc sql;
  create view THREE as
  select      ID, VAR1, VAR2
  from        ONE
  where       ID in
    (select ID from TWO);
quit;
```

A Solution by
Author: Charles Patridge
Email: Charles_S_Patridge@prodigy.net

```
Data ONE (keep=fmtname start label);
  RETAIN FMTNAME '$getrecd';
  infile cards ;
  input @1 ID $4.;
  start = id;
  label = 'Y';
  cards;
0001
0005
0009
0011
  ;;;;
run;

PROC SORT DATA=ONE OUT=ONE NODUPKEY;
BY START;
RUN;

PROC FORMAT LIBRARY=work CNTLIN=ONE;
RUN;

Data TWO;
  infile cards;
  input @1 ID $4.
        @5 Var1 $5.
        @10 Var2 $5.;
  if put(id,$getrecd.) = 'Y';
  cards;
0001 var1 var2
0002 var1 var2
0003 var1 var2
0004 var1 var2
0005 var1 var2
0006 var1 var2
0007 var1 var2
0008 var1 var2
0009 var1 var2
0010 var1 var2
0011 var1 var2
0012 var1 var2
0013 var1 var2
0014 var1 var2
0015 var1 var2
  ;;;;

proc print data=TWO; run;
```

TIP 00089

Solution by: Fabrizio
Email: Fabrizio1@usa.net

I have 500 variables in my dataset. Is there any efficient way to name all the variables into var1,var2,...,var500?

```
DATA _NULL_;
  SET SASHELP.VCOLUMN(KEEP=LIBNAME MEMNAME NAME);
  WHERE LIBNAME='libname'
  AND MEMNAME='memname';
  FILE 'temp.sas';
  IF _N_=1 THEN PUT 'RENAME';
  PUT NAME '= VAR' _N_;
RUN;

PROC DATASETS LIBRARY=libname;
MODIFY memname;
%INCLUDE 'temp.sas'; ;
RUN;
QUIT;
```

Notes:

- 1) replace lowercase words 'libname' with your libname in upper case.
- 2) same for 'memname'
- 3) Note the double semicolon following the %INCLUDE statement.

TIP 00090

Capture Options settings and set them back

Author: Don Stanley, B.SC, Dip O.R.S, MNZCS
Director, Sysware Consulting Group
Box 634, Wellington, NEW ZEALAND
Ph 064 21 479 863,
<http://www.sysware.co.nz>
Email: don_stanley@xtra.co.nz

Does anyone have an elegant solution to capturing the Options settings at the start of a program and setting it back when the program ends? I tried reading the dictionary options but setting them back is not that easy. The mprint, source, nosource, nomprint are easy. I just copied them to a macro and just called the macro at the end. My problem, for example, is the MISSING option. I had to write a code specifically identifying if it is missing and had to set them by putting the quotes with the space. There must be another way!!!

Definitely a better way from release 6.12 onwards, or 6.09E on mainframes.
Use the %sysfunc function to invoke the GETOPTION function. Here is an example using the MPRINT option.

```
options nomprint ;
%let mprint = %sysfunc(getoption(mprint)) ;
%put &mprint ;
options mprint ;
%let mprint1 = %sysfunc(getoption(mprint)) ;
%put &mprint1 ;
options &mprint ;
%let mprint2 = %sysfunc(getoption(mprint)) ;
%put &mprint2 ;
```

TIP 00091

Selectively Processing SAS Datasets

Sunil Kumar Gupta
Gupta Programming
Email: SunilKGupta@GuptaProgramming.com
Web: www.GuptaProgramming.com

How do you automate the process of generating the contents and a sample printing of all the sas datasets in a given directory without hard coding the names of each dataset? The following technical tip accomplishes this by reading a table of contents file and storing the sas dataset names into a field. Then, macro variables are created for each dataset name and are utilized by another macro to print the contents and a sample listing. First, go to the directory containing all your SAS datasets. Execute the directory command to save the names of all your datasets in the directory to a file to be read in by the SAS program:

```
C:\SASDATADIR *.SSD TOC.LST
```

Once you have this file, create your SAS program. Define a libname statement where the SAS datasets are located.

```
LIBNAME SDATA 'C:\SASDATA';
```

Create a dataset containing all of the dataset names in the dname field.

```
DATA OUTLST (KEEP=DNAME);  
LENGTH NAME $14. DNAME $8.;  
Access the table of contents file;  
INFILE 'C:\SASDATA\TOC.LST' END=FINAL;  
INPUT NAME;
```

Save only the dataset name to the dname field with the scan function.

```
DNAME=SCAN(NAME, 1, '.');
```

Print the values of name and dname to verify the scan function;

```
PUT NAME = DNAME=;
```

Create the macro variable total containing the total number of sas datasets.

```
IF FINAL THEN CALL SYMPUT('TOTAL', _N_);  
RUN;
```

Verify the contents of the dataset generated.

```
PROC CONTENTS DATA=OUTLST; RUN;  
PROC PRINT DATA=OUTLST; RUN;  
  
DATA NULL;  
SET OUTLST;
```

Create a macro variable for each dataset name. The root name is 'AN' with the suffix number incrementing with each dataset. The value of the macro variable is the name of the sas dataset contained in the dname field.

```
CALL SYMPUT('AN' || LEFT(_N_), DNAME)  
RUN;
```

(TIP 00091, continued on page 15)

Verify the contents of the macro variables.

```
%PUT AN1 = &AN1;
%PUT AN2 = &AN2;
%PUT TOTAL = &TOTAL;
```

Define a macro to generate the contents and a sample of the dataset.

```
%MACRO INFO(PDATA) ;
PROC CONTENTS DATA=&PDATA;
RUN;
PROC PRINT DATA=&PDATA (obs%) label;
RUN;
%MEND INFO;
```

Define a macro to select each of the macro variables just created containing the dataset name and execute the info macro for each dataset selected.

```
%MACRO PROCESS;
  %LET I=1;
  Process each macro variable one at a time;
  %DO %WHILE (&I <= &TOTAL);
    Apply the trim function to remove any trailing blanks in the field;
    %LET ITRIM=%TRIM(&AN&&I);
    Execute the info macro for each dataset in the directory;
    %INFO(SDATA.&ITRIM);
    Increment I by 1;
    %LET I = %EVAL(&I + 1);
  %END;
%MEND PROCESS;
```

Execute the process macro.

```
%PROCESS;
```

TIP 00092

```
/******
/****TIP00092.sas                                     ***/
/**** Author: Charles Patridge                       ***/
/****          PDPC, Ltd.                             ***/
/****          172 Monce Road                         ***/
/****          Burlington, CT 06013                  ***/
/****          Email: Charles_S_Patridge@prodigy.net ***/
/****          web: www.sasconsig.com                 ***/
/****          ***/
/**** A simple sas program to simulate the OLD PROC HIER ***/
/**** routine back in 1985.                           ***/
/****          ***/
/**** This is very crude but could easily be expanded ***/
/**** upon.                                           ***/
/****          ***/
/**** Ex: Input                                       ***/
/**** 3209 APTY                                       ***/
```

(Continued on page 16)

(Tip 00092, continued from page 15)

```
/**/ 3210     APTY                               ***/
/**/ 3212     APTY                               ***/
/**/ 3213     ACOP                               ***/
/**/ APTY     AIMS                               ***/
/**/ ARAF     AIMS                               ***/
/**/ ACOP     QTIM                              ***/
/**/ QTIM     SIAM                              ***/
/**/
/**/ Ex: Desired Output                         ***/
/**/   OBS      VAR1  VAR2  VAR3  VAR4  VAR5     ***/
/**/   1        3209  APTY  AIMS                ***/
/**/   2        3210  APTY  AIMS                ***/
/**/   3        3212  APTY  AIMS                ***/
/**/   4        3213  ACOP  QTIM  SIAM          ***/
/**/   5        APTY  AIMS                      ***/
/**/   6        ARAF  AIMS                      ***/
/**/   7        ACOP  QTIM  SIAM                ***/
/**/   8        QTIM  SIAM                      ***/
/**/*****/
```

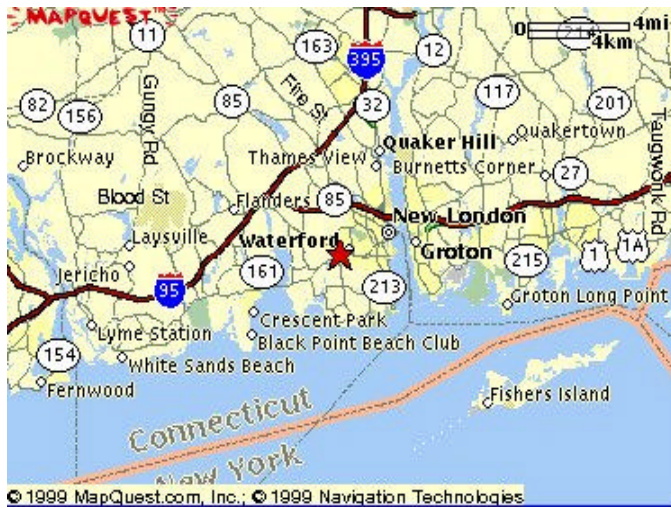
```
Data Test;
  length var1 $ 4 var2 $ 4;
  infile cards4;
  input var1 var2;
cards;
  3209     APTY
  3210     APTY
  3212     APTY
  3213     ACOP
  APTY     AIMS
  ARAF     AIMS
  ACOP     QTIM
  QTIM     SIAM
;;;

data testfmt (keep=start label fmtname);
  retain fmtname '$codefmt';
  set test;
  start = var1;
  label = var2;
run;

proc format library=work cntlin=testfmt; run;

data test1;
  set test ;
  var3 = put( var2, $codefmt. );
  if var2 = var3 then var3 = ' ';
  var4 = put( var3, $codefmt. );
  if var3 = var4 then var4 = ' ';
  var5 = put( var4, $codefmt. );
  if var4 = var5 then var5 = ' ';
run;

/**/ would have to expand last datastep based upon number of ***/
/**/ parent child relations you desired and I am sure a macro ***/
/**/ could be devised to figure when to stop the expansion ***/
```



Directions to the Waterford Library 49 Rope Ferry Road, Waterford, Connecticut

From I-95 North (Connecticut Turnpike East):
 take I-95 to Exit 81
 at stop sign turn left (cow fields will be on the right)
 immediately at light turn right, onto Crossroads
 see Continue below.

From I-95 South (Connecticut Turnpike West)
 take I-95 to Exit 81
 at stop sign turn left (Walmart will be on the right)
 at light, turn left onto Crossroads

Continue...

turn left at light onto Rte 1.
 at next light turn right onto Avery Lane
 (Silva's Package Store on corner).
 at next light turn left onto Rope Ferry Road
 (just past Post Office).
 Waterford Library is on your left in 1/8 mile.
 entrance is the 2nd driveway. Follow driveway to
 rear of building for parking.



Index of Tips in Newsletter

- | | |
|-----------|-------------------------------------|
| Tip 00081 | SAS to ASCII with SPSS |
| Tip 00083 | Fraction to Floating Point Numeric |
| Tip 00084 | Surround variable list with quotes |
| Tip 00085 | Multiple Datasets in Merge |
| Tip 00086 | C to F temperature conversion |
| Tip 00087 | DD-MMM-YY format of dates |
| Tip 00088 | Lookup Table Merge |
| Tip 00089 | Naming 500 variables efficiently |
| Tip 00090 | Capturing OPTIONS settings |
| Tip 00091 | Selectively Processing SAS Datasets |
| Tip 00092 | Simulate PROC HIER |

Hyperlinks of Interest

A method in SAS/AF and SAS/FSP to force date entries with 4-digit years.

<http://www.sas.com/service/techsup/unotes/V6/G101.html>

Accurately calculating age in only one line. Discussion of true leap years.

http://www.sas.com/service/doc/periodicals/sascomm/4q98/SC4Q98_38.htm

SAS Certification Discussion

<http://www.pace.edu/nesug/EXPRESS/nesp9910.pdf>

