



## HARTFORD AREA SAS® USERS GROUP

Volume 17 Issue 2

Second Quarter, May 2005

### Table of Contents

Notes from the Chairman	2
Meeting Announcement, The Hartford Insurance, Hartford, CT	2-3
Highlights of First Quarter 2005 Meeting	3
Calendar of Events	3
Book Review	4
Editorial and Production Staff	4
Continuing with Education	4-5
HASUG Sponsorship Policy	5
HASUG Recruiting Policy	5
Consultants Special Interest Group	6
HASUG Email Distribution List	6
Hyperlinks of Interest	6
Tips and Techniques, Disclaimer	7
Tips and Techniques	7-14
Steering Committee	15
Directions to The Hartford, Hartford, CT	16-17

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

## **Interim Chairperson's Notes Karol Katz**

HASUG locally, NESUG regionally and SUGI nationally and internationally offer SAS users an excellent opportunity to network, and improve or learn new programming skills.

Looking back, I remember how my affiliation with HASUG began. As a new graduate of the Research and Measurement program at Southern CT State University, my first position was in the Biostatistics Consulting Unit, Department of Epidemiology and Public Health at Yale. A co-worker was presenting a poster at NESUG '93 in Springfield, MA and invited me to attend. Since I was just driving up for the day and would not need money for lodging, our boss agreed to pay for my registration.

At NESUG, I attended several Beginning Tutorials in the morning. At lunch I joined my co-worker at her poster. She was scheduled to be there for an hour to get some feedback and answer attendees' questions about her poster. Michael Davis came by and invited us to join HASUG. I attended when my schedule permitted and found many of the presentations helped me directly in my SAS programming position.

After attending several meetings, I decided this was a worthwhile organization for which to work and I joined the steering committee. My first duty was as secretary. I maintained the mailing list and prepared the labels used to mail the HASUG FLASH to our members. That was when we produced only a hard copy version. Times have changed and at HASUG mailing labels are a thing of the past.

Then there was a request to hold a quarterly meeting in the New Haven area because of its central location in the state. Though Yale students use most of the auditorium facilities as large classrooms throughout the year, they are not used much in the summer.

With a few phone calls, I was able to get permission to use Winslow Auditorium as the site for that August meeting. We have used this site several times over the years and it will be the location again for the August meeting this year.

I hope you can attend our May meeting. We are fortunate to have Diane Olson, from SAS,

Cary, NC speak about migration from V8 to V9. She supports PROC MIGRATE, the new utility procedure. Jonas Bilenas, a BBU author, will explain the sometimes tricky PROC TABULATE syntax, so the table you want is the table you get. Please take a moment to complete the survey you receive. We listen to your requests and try to schedule presentations based on the results of the surveys.

Consider joining our steering committee. The time commitment is four meetings annually, immediately after the regular meeting (while having lunch) and 4 conference calls, usually one month after the quarterly meeting, for about an hour. Elections will be held at the May meeting. If you are interested in becoming a member of the steering committee, please let me know (karol.katz@yale.edu). Be a part of HASUG.

## **HASUG MEETING ANNOUNCEMENT**

Next meeting is Thursday May 19, at The Hartford's Tower Building, Atrium Conference Room, Hartford Plaza, Hartford, CT 9:00 am to noon. Topics include:

### **Making Sense of PROC TABULATE By Jonas Bilenas**

The TABULATE procedure in SAS® provides a flexible platform to generate tabular reports. Many beginning SAS programmers have a difficult time understanding the syntax of PROC TABULATE and tend to avoid using the procedure. This tutorial will explain the syntax of PROC TABULATE and, with examples, show how to grasp the power of PROC TABULATE. The data used in this paper represents simulated consumer credit card usage data and the code was developed using SAS 9. This updates the paper presented at NESUG in 2001 with the same title.

### **Speak Biography:**

Jonas V. Bilenas, Vice President of Product and Price Optimization at JP Morgan Chase, develops SAS applications for the consumer credit card industry, specializing in credit risk management, credit scoring, forecasting, pricing optimization, reporting, experimental design, and simulation. A SAS user since

1986, he has extensive experience using Base SAS software and applying the FORMAT, FREQ, and TABULATE procedures and the DATA step. Jonas holds an M.B.A. in quantitative analysis and is a frequent presenter at local SAS users group meetings. He is also an active contributor and speaker at the Northeast SAS Users Group (NESUG) conference.

**Proc Migrate: How to Migrate Your Data and Know You've Done It Right!**  
By Diane Olson of SAS Institute

Now that SAS 9.1 is available, what needs to be done to your data libraries in order to make use of the new version? Do you have to migrate your data? If you do, how can you do it most easily? Migrating your data to a new version of SAS presents challenges; depending on the attributes of your data libraries, your migration can be relatively simple or complex. Some issues to consider when adopting SAS 9 include the version of SAS in which your data currently resides, what member types exist in your libraries and whether you must move members from 32-bit libraries to 64-bit libraries. To address these issues, SAS 9.1 includes a new utility procedure, Proc Migrate. Proc Migrate streamlines the process of moving libraries forward to a new release. This presentation introduces Proc Migrate and discusses pitfalls of migrating with the traditional methods - Proc Copy, Proc Cport/Proc Cimport and Proc Catalog. Base SAS also provides tools that can help you guarantee the content and attributes of your data after migration. The validation tools are also presented and can be used to validate the integrity of your migrated libraries.

**Speak Biography:**

Diane Olson has been with SAS for a total of thirteen years and is a senior systems developer in the I/O group in the BASE R&D Division. She supports the utility procedures, including PROC APPEND, CONTENTS, COPY, DATASETS and the new MIGRATE. Her other

responsibilities include supervisor I/O subsystems, notably multi-threading.

**February HASUG Highlights**  
By Barbara Moss

The quarterly meeting of HASUG was held on February 24, 2005 at CIGNA in Bloomfield, CT. The opening presentation was, "What's New in SAS9" given by Rick Langston of SAS Institute. An energetic and entertaining presenter, Rick touched upon many of the new features in the Base Product, sometimes at a high level and sometimes giving specifics. Items catching my attention were; flexible concatenate functions for easier handling of blanks, a MEDIAN function, ANYDATE formats combining all the date format muscle in one place, sorting variables across a row, PIPES in MP Connect, and a PUTLOG statement for easier messaging to the log.

The second presentation was, "A Peek at PROC DOCUMENT" given by Andrew Karp of Sierra Information Services, Inc. Andrew intertwined pictures and stories of his recent world travels, including a trip to China, into his technical sharing. PROC DOCUMENT, new to SAS9, offers tools for managing ODS objects. A large benefit of the new procedure, beyond organization, is being able to save and re-use objects. This saves time and resources by eliminating the need to re-run code each time there is a need to produce the output. Along with the helpful walkthrough of functionality and capabilities, Andrew also pointed out the areas where there is room to grow and improve this new SAS offering.

Thanks to Rick, Andrew and all the folks in attendance, over 80 people, for contributing to an interesting, successful meeting.

**CALENDAR OF EVENTS 2005**

**Future General Meetings and Conferences:**

May 19,	The Hartford, Hartford, CT
May 22-25,	PharmaSUG, Phoenix, AZ
Aug. 18,	Yale EPH, New Haven, CT
Sept. 11-14,	NESUG, Portland, ME

**HASUG Steering Committee Meetings:**

May 19,	Immediately after HASUG meeting
June 16,	Conference call, 11:30-12:30

Aug. 18, Immediately after HASUG meeting

**Book Review**  
**By Janet Stuelpner,**  
**Left Hand Computing, Inc.**

**Title:** *Output Delivery System: The Basics*  
Author: Lauren E. Haworth  
Order Number: 59087  
ISBN number: 1-58025-859-X  
Cost: \$41.95 USD  
Length 320 pages

I always thought that the Output Delivery System (ODS) was very complicated and difficult to use. This book points out how easy it is to get started to make your output look more interesting and abolish the limitations of a normal listing. The author starts out in part one with the basics. She discusses why you need to use ODS and the definitions and syntax. In part two, she discusses each type of output (HTML, RTF and printer output). The last part goes in to more details of how to modify the basic template to get a more tailored output. There are lots of examples in the book with easy to follow code. Many of the procedures are discussed and you are shown how to make your own custom reports. One thing that makes this book easy to follow and different from many others is the way the discussion of a topic and the code are written on one page and the output is on the facing page. You don't have to go hunting through the book for the output and its matching code.

I highly recommend this book for anyone who want to create beautiful output from your procedures and transform ordinary listings to something very different. The intermediate programmer can benefit from this book along with the beginner. If you don't understand ODS, grab this book and learn quickly how to get started.

**EDITORIAL and PRODUCTION STAFF:**

Editor: Janet Stuelpner  
Layout: Albert Lavoie  
Member Lists: Charles Patridge  
Web master: Charles Patridge  
Contributors: Karol Katz  
Barbara Moss  
Chuck Patridge  
Peter Prause  
Janet Stuelpner

**Continuing With Education...**  
**By Peter Prause, The Hartford**

**The SAS Institute Regional Training Center** in Glastonbury, CT is offering the following courses. Course information for these courses and many other non-local courses is available on the Internet at the support.sas.com web site. To register for a course or to get more information, phone SAS Institute at 1-800-333-7660. Note: Only courses through June were available in time

SAS Macro Language	May 4–5
SAS Macro Programming: Advanced Topics	May 6
SAS Programming I: Essentials	May 10-12
SAS Programming II: Manipulating Data w/ DATA Step	Jun 1–3
Querying and Reporting Using SAS Enterprise Guide	Jun 13–14
SQL Processing with SAS	Jun 15–16

When time and distance are an issue, SAS Institute offers Live Web classes and self-paced e-learning to help you get the training you need while accommodating your busy schedule. Call SAS for more details.

*Education continued on (page 5)*

(Education continued from page 4)

**Destiny Corporation** is offering the following public courses at their training facility located at 2075 Silas Deane Highway, Rocky Hill, CT. You can reach Destiny at 1-800-787-2464 or at 860-721-1684, or send email to [info@destinycorp.com](mailto:info@destinycorp.com). You can visit their web site at [www.destinycorp.com](http://www.destinycorp.com).

Macros in SAS Software	May 2–4
Introduction to Programming Using SAS Software	May 12–13
Programming III: Advanced Techniques	May 16–17
Output Delivery System (ODS) Basics	Jun 2–3
Programming II: Data Manipulation w/ Data Step	Jun 6–8
SAS Version 9 Changes and Enhancements	Jun 9–10
Programming I: SAS Essentials	Jun 20–22
SQL Processing in SAS Software	Jun 27–28
Web Processing: Creating Static & Dynamic Solutions with SAS/IntrNet	Jul 7–8
Report Writing: A Programming Approach	Jul 18-20
Querying and Reporting With Enterprise Guide	Jul 21-22

In addition to Destiny's public course offerings, many of Destiny's courses are offered in a Virtual Learning Environment. Experience the quality and benefits of "Instructor Led" training from the comfort of your own pc. These courses cover the same basic content as traditional classroom training with the benefit of year long access and instructor support. Call or visit [www.vdestiny.com](http://www.vdestiny.com) for more details.

## THE 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 the 3rd Thursday from 9:00 am to noon at various locations throughout Connecticut. Past locations have included Hartford, Meriden, Middletown, Norwalk, Glastonbury, Stamford 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.

## THE 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 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 elsewhere in this

newsletter or visit [www.sconsig.com](http://www.sconsig.com)

## **CONSULTANTS' SPECIAL INTEREST GROUP (SCONSIG)**

The SAS Consultants' Special Interest Group (SCONSIG) operates and maintains a web site located at [www.sconsig.com](http://www.sconsig.com). The site lists employment opportunities, contracts, vitas, resumes, and companies seeking SAS professional talent. It is updated weekly. You are encouraged to visit that web site or to contact Charles Patridge at:

[Charles\\_S\\_Patridge@prodigy.net](mailto:Charles_S_Patridge@prodigy.net)

## **HASUG2-L DISTRIBUTION LIST**

This service is provided free of charge to all who wish to subscribe to the email notification process. In order to sign up for this service, you need only point your browser to the HASUG web site, <http://www.hasug.org>, and click the hyperlink

[Join HASUG Email Distribution List](#)

Or go directly to

<http://listserv.pace.edu/archives/hasug2-l.html>

then click on

[Join or leave the list \(or change settings\)](#)

After entering your name and e-mail address, there are a number of options available under "Subscription Type". If you know and understand the various options, feel free to alter them as desired. However, HASUG suggests you leave them "as is".

Your name and email address are not displayed nor given out to any party, including members of the HASUG steering committee. Only the list administrators have access to subscription information.

If you should change your email address under which you subscribed, you will need to unsubscribe with your old email address, and then re-subscribe using your new email address.

Typically, a notice is sent approximately 3 weeks before meetings, as well as a couple of days beforehand to remind people of the date, time, location and topics to be presented.

***Only List administrators will be able to send out email messages to the HASUG2-L members.***

## **HYPERLINKS OF INTEREST**

SAS Users often ask, "How can I find any recent SUGI papers that deal with a particular subject?" Lex Jansen has put up the following link that lets one search the online SUGI proceeds using the Google search engine:

<http://www.lexjansen.com/sugi/>

Similarly, to search the SAS-L archives, one can use the following two links:

<http://www.axomsoftware.com/sas-l.php>

<http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&group=comp.soft-sys.sas>

The SAS Consultants Special Interest Group's web site URL is:

<http://www.sconsig.com>

Read Past SUGI Proceedings Online at:

(link supplied by Curtis Smith):

<http://support.sas.com/usergroups/sugi/proceedings/index.html>

Information about the North East SAS Users Group (NESUG), including information about the 2005 conference in Portland ME can be found at:

<http://www.nesug.org>

SAS OnlineDoc is available from SAS on the World-Wide Web. To register to view SAS Version 8 OnlineDoc on the web, register at the following URL:

<http://www.sas.com/v8doc>.

To register to view SAS Version 9 Online Doc on the web, register at the following URL:

<http://www.sas.com/v9doc>.

Some of the papers and other handouts distributed at recent HASUG meetings can be found at the following web site:

<http://www.hasug.org>

## TIPS AND TECHNIQUES FROM SCONSIG

Disclaimer:

"Each tip included here addresses a problem, and presents a use of one or more SAS concepts. Use it to stimulate your thinking and help you on your way to solving your own problem. Some tips may or may not work in your environment or with your version of the SAS Software, and you should use these tips with this in mind. Neither HASUG nor contributing Author(s) should be held responsible for any direct and/or indirect damage to you or your applications from using these free tips."

*SAS, SAS/ACCESS, SAS/CONNECT, SAS/FSP, SAS/GRAPH, SAS/IntrNet, and SAS/STAT are registered trademarks of SAS Institute Inc., Cary, North Carolina.*

[Tips in blue/bold are located within this newsletter.](#)

Destiny's tip on migrating from V8 to V9 -  
Read More at <http://www.destinycorp.com/eKnowledgeBase/Data%20and%20Catalog%20Migration.htm>

Quick Tip: Using the MISSING Function  
Ever need to check for a missing value, but not sure if the variable is character or numeric? It's not a problem when using the MISSING function, explains BBU author Ron Cody.  
Read More at: [http://support.sas.com/publishing/bbu\\_tip/cody\\_missingFunction.html?ETS=2793&PID=44951](http://support.sas.com/publishing/bbu_tip/cody_missingFunction.html?ETS=2793&PID=44951)

Free Tutorial: Getting Started with SAS® ETL Studio  
This free tutorial provides step-by-step instructions and sample data for practicing common tasks using SAS ETL Studio. It also guides you through the process of building a sample data warehouse.  
Read More at: [http://www.sas.com/apps/elearning/elearning\\_details.jsp?pubcode=59274&ETS=2793&PID=44951](http://www.sas.com/apps/elearning/elearning_details.jsp?pubcode=59274&ETS=2793&PID=44951)

Explore SAS Communities

SAS communities are here to address your needs as a SAS user and offer you an intimate look at new and existing SAS products and solutions. The resources available from each community are provided to enhance your productivity and improve your SAS software experience. Choose from Base SAS, data visualization, enterprise intelligence, learning edition, migration, scalability and performance, statistics and operations research, and Web technologies. Read More at: <http://support.sas.com/rnd/intro.html?ETS=2793&PID=44951>

Quick Tip: Automated Storage Monitoring Routine for UNIX/Solaris Servers

Bryan Beverly, software architect and team leader with BAE Systems Information Technology, explains how to monitor file system storage for UNIX/Solaris servers. This two-part tip shows how to detect and report space problems, and how to prevent other production jobs from executing by "tripping a circuit breaker."  
Read More at: <http://support.sas.com/sassamples/quicktips/05feb/monitorunix.html?ETS=2793&PID=44951>

Quick Reference Guide to SAS Functions and Formats - TS486 (Dated 2/24/2000) -  
Thanks to Peter Crawford

Peaceful Coexistence: The Transition from SAS® 8.2 to SAS®9  
Moving from one release of SAS to another requires planning and testing. As you expand your SAS environment to include SAS®9, you must test the new environment and compare results with your existing one. Your testing and migration phase can stretch over several months; therefore, we have developed guidelines to help you through this transition. Note: Many SAS®9 and SAS 8.2 products are extremely compatible, making the transition smooth and straightforward.  
Read More at: <http://support.sas.com/rnd/migration/resources/peaceful.html?ETS=2791&PID=44951>

## The Relationship Between SAS/CONNECT® and the MACRO Facility

Many times when working with MACRO and SAS/CONNECT, the results you see are not what you expected. It's important when using RSUBMIT within the MACRO facility to have an understanding of what happens at compile time versus what happens at execution time. Knowing the behavior of this interaction will help you in using MACRO and SAS/CONNECT together. This article details how compiled code and text affect remote submitting code within a MACRO, as well as options and functions that can help in debugging. Techniques for creating MACRO variables on the local and remote host are also discussed.

Read More at: <http://support.sas.com/techsup/technote/ts697.pdf?ETS=2791&PID=44951>

## Quick Tip: Clean Up Your Messy Raw Data

Sometimes you need to read data that just doesn't line up in nice columns or have predictable lengths. When you have these types of messy files, ordinary list, column or formatted input simply isn't enough. You need more tools in your bag - tools like the '@'character' column pointer and the colon modifier. Find out more from BBU authors Lora D. Delwiche and Susan J. Slaughter.

Read More at: [http://support.sas.com/publishing/bbu\\_tip/delwiche\\_reading\\_messy\\_data.html?ETS=2789&PID=44951](http://support.sas.com/publishing/bbu_tip/delwiche_reading_messy_data.html?ETS=2789&PID=44951)

## Quick Tip: Concurrent Access to SAS/SHARE® with an Exponential Backoff

Phil Busby, a computer science major at North Carolina State University, submits the following tip to handle collision of packets: "Frequently, if one SAS process tries to access data in a table locked by another SAS process, it will come back to the user with an error, but continue executing the job. This can be trapped by checking &syserr." Get the complete tip from Phil.

Read More at: <http://support.sas.com/sassamples/quicktips/05jan/syserr.html?ETS=2789&PID=44951>

## Manipulating Data with PROC SQL by Kirk Paul

Lafler.

Read More at [http://www.sconsig.com/sastips/manipulating\\_data\\_with\\_proc\\_sql.pdf](http://www.sconsig.com/sastips/manipulating_data_with_proc_sql.pdf)

**TIP00405** - Macro to calculate deciles of a list of SAS variables from a single SAS dataset, and merge all the deciles onto a single record into an output SAS dataset using Proc Univariate by Charles Patridge

**TIP00406** - Find and delete records that offset (net zero) each other but only when the absolute values are equal by Charles Patridge

**TIP00407** - A Macro to apply the same function for a list of SAS variables where a function requires only a single variable as an argument by Charles Patridge and Peter Crawford (Multiple Solutions)

TIP00408 - Macro to return an expanded list of variables based on a SAS dataset by Ian Whitlock (Multiple Solutions)

## TIP 00407 Solution # 1 by Charles Patridge

```
/******  
/*** repeatfn.sas ***/  
/*** Author: Charles Patridge ***/  
/******  
/*Purpose is to apply the same function for */  
/*a list of SAS variables where function */  
/*requires only a single variable. */  
/*It then creates the SAS variables of the */  
/*output of the desired function and keeps */  
/*them in the same dataset. */  
/*User needs to be careful of the functions */  
/*called on certain variables for appropriate */  
/*values. */  
/*1st parameter: name of file to put temp */  
/* SAS code into. */  
/* */  
/* 2nd parameter: list SAS vars for function */  
/*to be applied to. */  
/* */  
/*3rd parameter: prefix for output var name*/  
/* */
```

```

/*4th parameter: function to be repeated. */
/*
/*Names of SAS Variables should be short */
/*enough to apply the user supplied prefix */
/*not exceed max length of a SAS variable */
/*(32 characters). */
/*
/*Only makes sense if you need to do the */
/*same function for many SAS variables */
/*(say more than 20) */
/*****/

%MACRO repeatfn( _tmpfile_ , _varlist_ ,
  _prefix_ , _function_ );
  options noxwait;
  /* delete tmp file to be sure it is clean */
  x del &_tmpfile_ ;
/* need to use appropriate OS command to
delete file */
  filename mytmp "&_tmpfile_" ;
/*TMPFILE where SAS code will be saved */
  data _null_ ;
  file mytmp notitle;
  /* determine number of SAS variables
being called */
  number = 1 + ( length(trim("&_varlist_"))
- length(compress("&_varlist_")) );
  DO J=1 TO NUMBER;
    word = scan( "&_varlist_" , j );
/* build the appropriate SAS code for function
called */
    string = "&_prefix_" || left(trim
(word)) ||
      " = &_function_(" || left(trim
(word)) || ')';
    put string;
  END;
  run;
%MEND repeatfn;

```

```

/****example of using and calling the
repeatfn macro ****/
/* example created for Windows OS platform
need to use appropriate file naming conven-
tion for your OS */

```

```

%let tempfila = c:\download\tmplog.txt;
%let tempfilb = c:\download\tmpexp.txt;
%let tempfilc = c:\download\tmpcos.txt;
%let tempfild = c:\download\tmpabs.txt;

%repeatfn ( &tempfila , x y z, ln_ ,
log );          /**** do log function ****/

%repeatfn ( &tempfilb , x y z, ex_ ,
exp );          /**** do exp function ****/

```

```

%repeatfn ( &tempfilc , x y z, co_ ,
cos );          /**** do cos function ****/

```

```

%repeatfn ( &tempfild , x y z, ab_ ,
abs );          /**** do abs function ****/

```

```

data test;
  x = 1.5;
  y = -2.0;
  z = 3.5;
  %include "&tempfila";
  %include "&tempfilb";
  %include "&tempfilc";
  %include "&tempfild";
run;

```

```

/*****

```

```

tmplog.txt will contain:

```

```

  ln_x = log(x);
  ln_y = log(y);
  ln_z = log(z);

```

```

tmpexp.txt will contain:

```

```

  ex_x = exp(x);
  ex_y = exp(y);
  ex_z = exp(z);

```

```

tmpcos.txt will contain:

```

```

  co_x = cos(x);
  co_y = cos(y);
  co_z = cos(z);

```

```

tmpabs.txt will contain:

```

```

  ab_x = abs(x);
  ab_y = abs(y);
  ab_z = abs(z);

```

```

*****/

```

## Solution #2 by Charles Patridge

```

/*****
/* repeatfn.sas */
/* Author: Charles Patridge */
/*****/
/*
/* Purpose is to apply the same function for*/
/*a list of SAS variables where function */
/*requires only a single variable. */

/*It then creates the SAS variables of the */
/*output of the desired function and keeps */
/*them in the same dataset. */
/*
/*User needs to be careful of the functions */
/*called on, certain variables for */
/*appropriate values. */

```

```

/*
/* 1st parameter: list of SAS vars for
/*Function to be applied to.
/*
/* 2nd parameter: prefix for output var
/*name
/*
/* 3rd parameter: function to be repeated
/*
/* Names of SAS Variables should be short
/*enough to apply the user supplied prefix
/* to not exceed max length of a SAS
/*variable (32 characters)
/*
/*
/* Only makes sense if you need to do the
/*same function for many SAS variables
/*(say more than 20)
/*****
    %MACRO REPEATFN(_varlist_, _prefix_,
    _function_ );
        %LOCAL _J_ _NUMBER_;

/*determine number of SAS variables in
_varlist_*/
        %let _NUMBER_ = %eval( 1 + (%
        length(%sysfunc( complbl(&_varlist_))) -
        %length(%
        sysfunc(compress(&_varlist_)))));
        %DO _J_=1 %TO &_NUMBER_;

/* now generate SAS code for function call */

        &_prefix_%scan(&_varlist_, &_J_)
= &_function_(%scan(&_varlist_,&_J_));
        %END;
%MEND REPEATFN;

data test;
x = 1.5;
y = -2.0;
z = 3.5;

        /*** do log function ***/
        %REPEATFN(x y z, log_, log );

        /*** do abs function ***/
        %REPEATFN(x y z, abs_, abs );

        /*** do cos function ***/
        %REPEATFN(x y z, cos_, cos );

run;

```

**Solution #3 by Peter Crawford**

```

/*-----*/
/* Program : MLOOOPSX.SOURCE
/*-----*/
/* Purpose : List processing: for each
/* element of a list.
/* call a macro, without going through the
/* statement boundary.
/*****
/* Author : Peter Crawford—
/* Crawford Software Consultancy Limited
/* Version : 2. * Created : 14.10.02
/*****
/* Modification History
/* Peter Crawford 14.10.02
/* prefix local macro vars with macro name,
/* to avoid contention with any macro vars
/* inherited by the macro called.
/* Peter Crawford 20.02.04
/* apply %superQ() to macro vars in tests,
/* to avoid arithmetic in implicit %eval()
/*****
    %macro mLoopsX( execut =
/* name of macro to invoke */
        , with =
/* delimited list of calls */
        , withdlm = %str( )
/* list delimiter */
        ) /des='execute a macro for each
word in a list' ;

/*****
examples:
suppose you have a macro like
    %macro syk( op );
        %sysfunc( getoption( &op,keyword ) )
    %mend syk ;
then
    %put oldops = %mloopsx( execut= syk,
with=mprint mlogic symbolgen);

loads re-usable settings for these system
options
The macro can operate on larger scale as a
statement like
    %mloopsx( execut=main_process, with=
London New-York Tokyo )
*****/

    %if &execut eq or %superQ(with) eq %
then %goto mParmError ;
    %local mloopsx_i mloopsx_j ;
    %let mloopsx_i = 1 ;
    %let mloopsx_j = %scan( &with,
&mloopsx_i, &withdlm );
    %do %until( %superQ(mloopsx_j) eq ) ;
        %&execut.( &mloopsx_j )

```

```

    %let mloopsx_i = %eval
    ( &mloopsx_i + 1 );
    %let mloopsx_j = %scan( &with,
&mloopsx_i, &withdm );
    %end ;
    %goto mExit ;

    %mParmError:
    %put mLoopsX-Error need parameters,
got execut=&execut and with=&with ;
    %mExit:
    %mend mLoopsX ;

/** create your own little macro functions **/
%macro mylog(_x_) ;
log_&x_ = log( &x_ ); /* log function */

    %mend mylog;

%macro myabs(_x_) ;
abs_&x_ = abs( &x_ ); /* abs funtion */
    %mend myabs;

%macro mycos(_x_) ;
cos_&x_ = cos( &x_ ); /*cos function */
    %mend myabs;

/*now call your macro functions and apply
them to your list of SAS variables you want */

%let mylist = x y z;
/** list of SAS variable I need to apply my
functions to ***/

data test;
x = 1;
y = -2.5;
z = 3.2;
%mLoopsX( execut=mylog,
with=&mylist ); /*** apply log ***/
%mLoopsX( execut=myabs,
with=&mylist ); /*** apply abs ***/
%mLoopsX( execut=mycos,
with=&mylist ); /*** apply cos ***/
run;

/*****
/** end of tip 00407 ***/
*****/

```

#### TIP 00406

```

/*****
/* Find and delete records that offset (net */

```

```

/*zero) each other but only when the */
/*absolute values are equal */
/*****

/*** create sample test file ***/
data test;
claimno = 'A100';
x = -100; output; /* will offset with next
record */
x = 100; output; /* will offset with prior
record */
x = 100; output;
x = -50 ; output;
x = -50 ; output; /* will offset with next
record */
x = 50 ; output; /* will offset with prior
record */
x = 90 ; output;
x = 70 ; output; /* will not offset with
next 2 records */
x = -35 ; output;
x = -35 ; output;
x = 70 ; output;
run;

/*** separate transactions into positive and
negative datasets ***/

data plus minus;
set test;
absx = abs(x);
if x ge 0.00 then output plus;
else output minus;
run;

proc sort data=plus ; by claimno x ; run;
proc sort data=minus; by claimno absx;
run;

/*** create unique id for each duplicate
valued record ***/
data plus;
set plus;
by claimno absx;
if first.absx then _cnt_ = 1;
else _cnt_ + 1;
run;

/*** create unique id for each duplicate
valued record ***/

data minus;
set minus;
by claimno absx;
if first.absx then _cnt_ = 1;
else _cnt_ + 1;
run;

```

```

proc sort data=plus ; by claimno x
  _cnt_ ; run;
proc sort data=minus; by claimno absx
  _cnt_ ; run;

/**** now join positive and negative records
for each equal ABS and delete if equal ****/

```

```

data join;
  merge plus (in=p)
        minus (in=m)
  ;
  by claimno absx _cnt_;
  if p and m then delete;
run;

```

```

*****/
/**** end of tip 00406 ****/
*****/

```

#### TIP 00405

```

/*****/
/**** _deciles_.sas ****/
/**** ****/
/**** Author: Charles Patridge ****/
/**** Date: November 30, 2004 ****/
/**** ****/
/*Macro to calculate deciles of a list of SAS */
/*variables from a single SAS dataset, and */
/*merge all the deciles onto a single record */
/*into an output SAS dataset. */
/*****/
/**** ****/
/*1st parameter = name of SAS dataset to */
/*create deciles from ie: mylib.mydataset */
/**** ****/
/*2nd parameter = list of SAS variables to */
/*use to make deciles each SAS variable */
/*needs to be separated by a blank. */
/* medage medincome weight height */
/* ****/
/*No SAS variable name can be longer than */
/* 29 positions. */
/* ****/
/***** SAS variables are not checked for */
/* misspellings, or if they exist. */
/* ****/
/*3rd parameter = name of SAS output */
/*dataset to store deciles mylib.mydeciles */
/**** ****/
/**** ****/
/*4th parameter = name of SAS variable for*/
/*weighting deciles */
/* (leave blank if none needed) */
/**** ****/

```

```

/
*****
/
/**** this libname used for demo purposes
**/

```

```

**** libname mylib
'c:\download\sas_salary\';

```

```

%macro _deciles_ ( _sasdsnin_=,
  _sasvars_=, _sasdsnout_=,
  _sasweight_=);

```

```

/*get number of SAS variables being
decided*/

```

```

%let numwords = %eval(1 + %length
(%cmpres(&_sasvars_)) -
  %length(%sysfunc(compress
(&_sasvars_)));

```

```

%put "Number of SAS Variables being
decided = &numwords";

```

```

/**** check to see if _sasweight_ is empty
orblank ****/

```

```

%if "&_sasweight_.dummy" =
"dummy" %then %let _sasweight_ =
_dummy_;

```

```

%do i = 1 %to &numwords;
/**** loop through list of SAS variables to
create deciles for each one ****/

```

```

%let _stat_ = %scan( &_sasvars_,
&i);
proc univariate data=&_sasdsnin_
  noprint;
  var &_stat_;

```

```

%if &_sasweight_ ne _dummy_ %
then %do;
/**** apply a weighting factor if requested
**/

```

```

  freq &_sasweight_;
%end;

```

```

  output out=&_stat_ pctlpts=10 to
100 by 10 pctlpre=&_stat_;
run;

```

```

%end;

```

```

/*now merge all deciles onto a single record*/

data &_sasdsnout_;
merge &_sasvars_;
run;

data &_sasdsnout_;

/** join deciles with input SAS dataset */
if _n_ = 1 then set &_sasdsnout_; set
&_sasdsnin_;

%do i = 1 %to &numwords;
%let _stat_ = %scan( &_sasvars_, &i);

/** create decile number for decile by
attaching bin to SAS variable name */

if &_stat_ le &_stat_.100 then
&_stat_.bin = 10;
if &_stat_ le &_stat_.90 then
&_stat_.bin = 9 ;
if &_stat_ le &_stat_.80 then
&_stat_.bin = 8 ;
if &_stat_ le &_stat_.70 then
&_stat_.bin = 7 ;
if &_stat_ le &_stat_.60 then
&_stat_.bin = 6 ;
if &_stat_ le &_stat_.50 then
&_stat_.bin = 5 ;
if &_stat_ le &_stat_.40 then
&_stat_.bin = 4 ;
if &_stat_ le &_stat_.30 then
&_stat_.bin = 3 ;
if &_stat_ le &_stat_.20 then
&_stat_.bin = 2 ;
if &_stat_ le &_stat_.10 then
&_stat_.bin = 1 ;

/** drop stat variables - no longer needed*/
drop
&_stat_.10 &_stat_.20 &_stat_.30
&_stat_.40 &_stat_.50
&_stat_.60 &_stat_.70 &_stat_.80
&_stat_.90 &_stat_.100;
%end;
run;

/** delete the temporary SAS datasets */

proc datasets library=work nolist; delete
&_sasvars_ ; quit;

%mend _deciles_;

%let mystatvars = salary hrlyrate billrate ;

```

```

/** example call to _deciles_ */

/** %_deciles_( _sasdsnin_ = mylib.
salary,
_sasvars_ = &mystatvars,
_sasdsnout_ = accumulate,
_sasweight_ =
); */

/*****
/** end of tip 00405 */
*****/

```

### BBU Tip - Using the MISSING function from SAS Functions by Example

support.sas.com > SAS Publishing --  
Navigate our Site -- Contact Us  
Communities Documentation Search  
Software Downloads Technical Support  
Training www.sas.com

Using the MISSING function from SAS  
Functions by Example  
By Ron Cody

Ever need to check for a missing value, but you're not sure if the variable is character or numeric? No problem when you use the MISSING function. This function takes either character or numeric variables and it checks for the .A, .B, .\_ numeric missing values as well. For example:

```

DATA MISSING;
INPUT CHAR $ X Y;
IF MISSING(CHAR) THEN N_CHAR + 1;
IF MISSING(X) THEN N_X + 1;
IF MISSING(Y) THEN N_Y + 1;
DATALINES;
CODY 5 6
...
WHATLEY .A ._
LAST 10 20
;
PROC PRINT DATA=MISSING NOOBS;
TITLE "Listing of MISSING";
RUN;

```

A listing of MISSING, below, shows that the MISSING function works correctly with character and numeric values, including all the alternative numeric missing values:  
Listing of MISSING:

CHAR	X	Y	N_CHAR	N_X	N_Y
CODY	5	6	0	0	0
WHATLEY	A	_	1	2	2
LAST	10	20	1	2	2

```

THomaS eDISON
albert einstein
;
PROC PRINT DATA=PROPER NOOBS;
  TITLE "Listing of Data Set PROPER";
RUN;

```

Keep up with SAS current and new tips at  
<http://support.sas.com/publishing/tip.html>.  
 For other tips, please visit our [Tip Archive](#)

### **BBU Tip - Using the PROPCASE function from SAS Functions by Example**

support.sas.com > SAS Publishing --  
 Navigate our Site -- Contact Us  
 Communities Documentation Search  
 Software Downloads Technical Support  
 Training [www.sas.com](http://www.sas.com)

Using the PROPCASE function from SAS  
 Functions by Example

By Ron Cody

The "old" way to capitalize the first letter of words was to use LOWCASE, UPCASE, and the SUBSTR function, like this:

```

DATA CAPITALIZE;
  INFORMAT FIRST LAST $30.;
  INPUT FIRST LAST;
  FIRST = LOWCASE(FIRST);
  LAST = LOWCASE(LAST);
  SUBSTR(FIRST,1,1) = UPCASE(SUBSTR
(FIRST,1,1));
  SUBSTR(LAST,1,1) = UPCASE(SUBSTR
(LAST,1,1));
DATALINES;
ronald cODy
THomaS eDISON
albert einstein
;
PROC PRINT DATA=CAPITALIZE NOOBS;
  TITLE "Listing of Data Set CAPITALIZE";
RUN;

```

With the PROPCASE function in SAS 9.1, it's much easier.

```

DATA PROPER;
  INPUT NAME $60.;
  NAME = PROPCASE(NAME);
DATALINES;
ronald cODy

```

## THE HASUG STEERING COMMITTEE

<u>Name</u>	<u>Organization</u>	<u>Telephone</u>	<u>E-Mail</u>
James Alling	Independent	(860) 447-2253	jtalling2001@yahoo.com
Rachel Alt-Simmons	Hartford Life	(860) 843-4825	rachel.altsimmons@hartfordlife.com
Jonas Bilenas	JP Morgan Chase Bank	(212) 552-7303	Jonas.Bilenas@chase.com
Jim Blaha	CCSU	(860) 832-1737	blahajam@ccsu.edu
Miguel Hernandez	Client Associated Business	(203) 882-8080	mahernandezjr@hotmail.com
Cindy Johnson	Bristol-Myers Squibb	(203)-677-6547	GjCjtwink@aol.com
Karol Katz	Yale University	(203) 764-7527	Karol.Katz@yale.edu
Rene Laurencot	Pfizer	(860) 686-1574	husky_123@yahoo.com.
Albert Lavoie	Independent	(860) 644-7632	alavoie1@cox.net
Tom Mansfield		(203) 937-5126	lmansfield@snet.net
Barbara Moss	CIGNA	(860)226-1504	Barbara.Moss@CIGNA.com
Chuck Patridge	Deloitte Consulting	(860) 725-3329	Charles_S_Patridge@prodigy.net
Peter Prause	The Hartford	(860) 547-7484	Peter.Prause@thehartford.com
Kevin Smith	Oxford Health Plans	(203) 459-6145	KevSmith@oxhp.com
Janet Stuelpner	Left Hand Computing	(203) 588-8583.	jstuelpner@usa.net
Gerald Zuckier		(203) 624-0940	GZuckier@alum.mit.edu

**The Hartford Main Office  
Hartford Plaza  
Hartford, Connecticut 06115**

Note: Directions are from the East or West via I-84.; To reach I-84, use I-91, Route 15, or another route that connects with I-84.;

**FROM EAST (via I-84):**

Approaching Hartford, follow I-84 West to Exit 48 (Asylum Ave Exit).  
Approaching the traffic light at the end of the ramp, get in the right lane.  
Make a right turn for Garden Street, just before the traffic light. There is a stop sign before entering Garden Street. Proceed up Garden Street.  
At next traffic light, take left onto Cogswell Street.  
Take immediate first right into Visitors Parking.  
Enter Hartford Tower using the Atrium entrance, which is West of the Visitors Lot.

**FROM WEST: (via I-84):**

Approaching downtown Hartford, follow I-84 East to Exit 48A (Asylum Ave Exit).  
At the traffic light take left onto Asylum Avenue  
At the first traffic light, go straight. Then bear right at the Y split to stay on Asylum Ave.  
At the second traffic light, take a right onto Cogswell Street.  
Before the next traffic light, take a left into Visitors Parking.  
Enter Hartford Tower using the Atrium entrance, which is West of the Visitors Lot.



