



## SESUG Speaker Sharing Program

To arrange for a SESUG speaker, contact Marje Fecht at [Marje.Fecht@prowerk.com](mailto:Marje.Fecht@prowerk.com)

### List of Presentation Titles

- Analysis of Longitudinal Data: Comparison between PROC GLM and PROC MIXED.
- An Introduction to the Simplicity and Power of SAS/Graph®
- A Pragmatic Programmers Introduction to ETL Studio: A Hands on Workshop
- Base SAS vs. ETL Studio: Understanding ETL and the SAS tools used to support it
- Best Practices for Automated Testing and Real Time Notification in SAS Applications
- DATALINES, Sequential Files, CSV, HTML and More – Using INFILE and INPUT Statements to Introduce External Data into the SAS® System
- Dictionary Tables: Essential Tools for Serious Applications
- Don't be a Slave to your SAS Programs
- Fun with Functions
- Getting Data into SAS®: INFILE and INPUT
- How to Incorporate Old SAS® Data into a New DATA Step, or “What is S-M-U?”
- Individual Growth Analysis Using PROC MIXED
- In Search of the LOST CARD
- Program Comprehension: A Strategy for the Bewildered
- Programming for Failure: When Programs Faw Down and Go Boom
- Pruning the SASLOG – Digging into the Roots of NOTEs, WARNINGs, and ERRORs
- Real Time Decision Support: Creating a Flexible Architecture for Real Time Analytics
- Real Time: What is it and what are we doing about it
- Rules for Tools - The SAS Utility Primer
- SAS as a Utility Language: Using SAS for Solving Little Problems
- Shuffling, Shifting, and Structuring Using the Program Editor
- The SAS Debugging Primer
- The Utter “Simplicity?” and Power of the TABULATE Procedure
- The Utter Simplicity of the TABULATE Procedure
- Using BI Tools from SAS for Clinical Reporting
- Using ETL Studio for Clinical Research
- Version 9 Programming Enhancements
- Web Development: Best Practices for Developing Enterprise Applications
- Writing the "Best" Program: The How and When of Efficient Programming
- XML and SAS: An Advanced Tutorial



## SESUG Speaker Sharing Program

### List of Abstracts

#### **Analysis of Longitudinal Data: Comparison between PROC GLM and PROC MIXED**

Maribeth Johnson

Longitudinal data refers to datasets with multiple measurements of a response variable on the same experimental unit made over a period of time. These types of data require special attention because they involve correlated data. The relationships between repeated measurements are important in assessing reliability and tracking of those measurements. The proper variance-covariance structure in the analysis model is essential to the understanding and interpretation of those relationships. The assumption of compound symmetry necessary for correctly using the intraclass correlation as a measure of tracking can be tested against other variance structures using PROC MIXED. This paper compares the variance, covariance and correlation estimates obtained from the GLM and MIXED procedures of SAS/STAT® on two sets of data, one of which has missing data.

#### **An Introduction to the Simplicity and Power of SAS/Graph®**

Dan Bruns

Given the ever-increasing demand for more information better and faster, the graphics tools from SAS provide a powerful way of presenting information succinctly. The use of the procedures in SAS/GRAPH is consistent with base SAS procedures, so they are easy to learn and experiment with. This tutorial will explore how to apply your SAS skills to the syntax, statements, features and options available in SAS/GRAPH to expand and enhance your reporting capabilities. You will take away an understanding of the flexibility of the product the kinds of impressive and informative graphs that can be printed, imbedded in other documents, or made available to web sites.

#### **A Pragmatic Programmers Introduction to ETL Studio: A Hands on Workshop**

Greg Nelson

ETL is the process of moving data from a source system (such as operational systems or a table in a database) into a structure that supports analytics and reporting (target). This workshop will guide participants through a structured, hands-on exercise designed to give them a broad overview of what things we can accomplish with ETL Studio. Here we will prepare data for use by extracting data from an external file, dealing with the exceptions (bad data), creating transformations that enrich our data, combining it with other data for completeness and finally loading the data into tables that are part of a star schema. The goal of this workshop will be to get users comfortable with the tool and demonstrate its capability.



## SESUG Speaker Sharing Program

### **Base SAS vs. ETL Studio: Understanding ETL and the SAS tools used to support it** Greg Nelson

Every data warehouse, data mart and data hut needs it. Every good business intelligence interface depends on it. It has been the hallmark of what SAS programmers have done for over 30 years – the ability to beat data into submission (a.k.a. ETL - extract, transform and load data). Now all grown up and ready to take on the world, SAS' ability to get at just about any data source, massage, cajole, beat, cleanse and transform information and then make it sing and dance in your tool of preference makes SAS the ideal choice for modern decision support applications.

So which tools, technologies and/or approaches can we use for ETL and which make sense for any given application? We will explore the world of ETL from the perspective of the tasks that we have to accomplish and compare using BASE SAS tools versus ETL Studio. To that end, we will highlight what a good ETL system should be able to do by taking a lesson from Ralph Kimball and his book outlining the 38 subsystems for ETL. Here we will touch on several of the key tasks found in ETL and show you how to accomplish these using both BASE SAS and ETL Studio. In addition, we will summarize the major capabilities of each approach as a quick reference for management.

### **Best Practices for Automated Testing and Real Time Notification in SAS Applications** Greg Nelson

Data management is one of the cornerstones of SAS as a language. SAS programs that access, manage, analyze and report on data are often taken from vast libraries of tools that are used over and over again for consistency and desirable for their reuse in similar projects. Over time, the number of potential uses of any one program or macro is challenged by the amount of time it takes to test, retest and validate these programs. As these programs become part of the production eco-system in a development environment, it is important they their testability, robustness and manageability become "built-in" to the software development process.

This paper outlines as a specific approach to building in that process to each and every program to monitor the conditions SAS programs encounter and proactively test for and announce any validation issues. We will explore the concept of automated tests through assertions, events and their attributes, event status management, and automatic notification of events to interested parties. These concepts are presented from the perspective of the SAS programmer and the systems analyst.

### **DATALINES, Sequential Files, CSV, HTML and More – Using INFILE and INPUT Statements to Introduce External Data into the SAS® System**

Andrew T. Kuligowski

The SAS® System has numerous capabilities to store, analyze, report, and present data. However, those features are useless unless that data is stored in, or can be accessed by, the SAS System. This presentation is designed to review the INFILE and INPUT statements. It has been set up as a series of examples, each building on the other, rather than a mere recitation of the options as documented in the manual. These examples will include various data sources, including DATALINES, sequential files, CSV files, and HTML files.

Last Update: 27Jun2006



## SESUG Speaker Sharing Program

### **Dictionary Tables: Essential Tools for Serious Applications**

Frank Dilorio

Dictionary tables and views are some of the most useful and least heralded tools in the SAS System. They contain a wealth of information about SAS datasets, catalogs, system options, and external files. While much of this information is available via cumbersome procedure output datasets, dictionary tables and views (which we will simply refer to as "the tables") put this information at the disposal of the application developer in a straightforward manner. The use of dictionary tables in combination with the macro language and PROC SQL, two much more utilized SAS products, provides a powerful set of tools for the SAS programmer.

This paper: explains the organization of the tables, describes their content, and presents several short applications demonstrating how the tables can be used in combination with the macro language and PROC SQL to create more efficient and dynamic applications. The reader should come away from the discussion with an understanding of the benefits of the dictionary tables as well as some ideas for how they may be used in their programming environments.

### **Don't be a Slave to your SAS Programs**

Marje Fecht

Do you spend a lot of time updating your programs to adjust inclusion criteria? Do your programs run forever while you patiently wait? This beginning tutorial focuses on maintenance - free, efficient coding techniques so that you can spend your work time being more productive!

Topics include:

- macro coding techniques
- efficient programming tips
- code reduction tricks
- maintenance-free programming suggestions.

### **Fun with Functions**

Marje Fecht

Functions can be fun (and useful) if you know when, how, and why to use them. Functions can be frustrating if you don't know the tricks behind them.

In this presentation, we will explore the most commonly used functions to streamline data processing and reduce your programming effort. You will learn how to use summary functions, string functions, conversion functions, and date functions. We will also explore some of the 60 functions added to Version 9 SAS software.

This presentation will also help you understand WHERE to use functions, since they aren't "just for the DATA Step".



## SESUG Speaker Sharing Program

### Getting Data into SAS®: INFILE and INPUT

Andrew T. Kuligowski

The SAS System has numerous capabilities to store, analyze, report, and present data. However, those features are useless unless that data is stored in, or can be accessed by, the SAS System. This presentation includes an introduction to the INPUT and INFILE statements, which combine to provide a simple, yet powerful, method to pass data into the SAS System. Matters to be addressed are reading fixed and variable length files, .CSV files, in-stream data, informats, and more.

### How to Incorporate Old SAS® Data into a New DATA Step, or “What is S-M-U?”

Andrew T. Kuligowski

S-M-U. Some people will see these three letters and immediately think of the abbreviation for a private university and associated football team in Texas. Others might treat them as a three-letter word, and recall a whimsical cartoon character created by Al Capp many years ago. However, in the world of the SAS® user, these three letters represent the building blocks for processing SAS datasets through the SAS DATA step. S, M, and U are first letters in the words SET, MERGE, and UPDATE – the 3 commands used to introduce SAS data into a DATA step.

This presentation will discuss the syntax for the SET, MERGE, and UPDATE commands. It will compare and contrast these 3 commands. Finally, it will provide appropriate uses for each command, along with basic examples that will illustrate the main points of the presentation.

### Individual Growth Analysis Using PROC MIXED

Maribeth Johnson

Individual growth models are designed for exploring longitudinal data on individuals over time. PROC MIXED allows the growth parameters for each individual to be examined as random effects in the model. Individual-level covariates can be entered into the model as fixed effects to determine their impact on the dependent variable alone and in interaction with the growth parameters. The structure of the variance-covariance matrix of the repeated measurements can also be examined and entered into the model. A model building exercise will be demonstrated using up to eight systolic blood pressure measurements of youths aged 7-22



## SESUG Speaker Sharing Program

### **In Search of the LOST CARD**

Andrew T. Kuligowski

"Everyone who's not here, raise your hand". It's an old joke, but it points out the difficulty of identifying persons or things that are not present. The SAS® system has its own version of this chestnut, the SASLOG message indicating that there are one or more gaps in one's input data:

NOTE: LOST CARD.

This presentation will focus on the creation and use of adhoc's to explore the positions where the input data might be incomplete. The goal will be to identify where the missing data should be, so that you can code around the limitations of your data.

*NOTE: This is only a 20-minute presentation. The user group who requests this might want to consider it as a second paper to another being presented by the author.*

### **Program Comprehension: A Strategy for the Bewildered**

Frank Dilorio

Here, unfortunately, is a not uncommon workplace scenario. A neophyte SAS programmer is assigned to maintain, debug, or enhance an application. The atmosphere is sink or swim, the system is complex, the code is sophisticated, the documentation is scant, and the programmer is bewildered. Questions slowly take shape. "What, exactly, am I supposed to do?" "What part(s) of the application need my attention?" "Will changing program X affect program Y?" And, most critically, "Where do I start?"

What the programmer needs is a strategy for understanding the program, then finding its "sweet spots" as efficiently as possible.

This paper presents a generalized approach for programmers, particularly SAS "newbies", to develop an understanding of how applications work. It also shows how to translate this comprehension into effective coding. The paper identifies and discusses the rationale for questions the programmer should ask about: task definition, program-level code, supporting code, system design and specification documents, and required domain knowledge.

Attendees should come away with a better understanding of how to frame the programming problem and effectively gather the resources needed to obtain a solution. They will also come to believe that the coding of, say, a DATA step is usually simple, but the real art of programming is learning what to code, and why.



## SESUG Speaker Sharing Program

### **Programming for Failure: When Programs Faw Down and Go Boom**

Gary Schlegelmilch

Programs fail.

Despite the best efforts of the best programmers, a non-working combination of data, software, and processing is invariably going to happen - and the program is not going to run. Typical result is a sometimes instantaneous, sometimes arduous effort to find where the program has failed - and fixing it.

This presentation attempts to cover two major topics: (1) a few ideas for finding the elusive bug, and (2) planning and structuring a SAS program so that perhaps a few less of them happen.

### **Pruning the SASLOG – Digging into the Roots of NOTES, WARNINGS, and ERRORS**

Andrew T. Kuligowski

You've sat through constant design meetings. You've endured countless requests for "just one more little change". You even managed to find a creative solution to that nagging technical problem. But, you persevered, and despite all of the obstacles, you've managed to eliminate the final syntax error in your newest SAS routine. Time to sit back and relax -- uh, not quite ...

The primary focus of this presentation will be on techniques to ensure comprehension of your input data. We will look at several messages that are often found in the SASLOG, such as:

NOTE: MERGE statement has more than one data set with repeats of BY values.

that imply that there may be gaps in your knowledge of your data! Special emphasis will be placed on the use of ad-hoc queries to assist in finding data anomalies that can cause problems with your SAS code. It is assumed that the reader has a basic understanding of the SASLOG, including its composition, format, and the SAS system options which control its content.

### **Real Time Decision Support: Creating a Flexible Architecture for Real Time Analytics**

(Note: appropriate as a Keynote/ Plenary Session presentation)

Greg Nelson

Leaders have focused their entire careers on their ability to gather, assess, evaluate and assimilate data to effectively drive change. The deployment of enterprise systems and strategic initiatives to support customer intimacy and organizational preparedness has often led to the development of data warehousing and business intelligence applications that optimize the data paths between those who know and those who should know. The end result of much of this effort is a complete infrastructure designed to move data through the enterprise. Drip feeds, wipe and load, "slowing changing" dimension management, swim-lanes, parallelization, data optimization – all technical details that obscure the fact that data is still 12 hours old. This presentation focuses on the things that we can do today to make data movement happen so that decisions can be made with better quality, in near real time. In addition, attention will be paid to when we should drive for real-time decision support and when it might not be appropriate. Finally, we will discuss a framework that supports low cost, incremental improvements in your information architecture at the same time optimizing the business processes to ensure information transparency across the enterprise.

Last Update: 27Jun2006





## SESUG Speaker Sharing Program

### **Real Time: What is it and what are we doing about it**

Greg Nelson

Real-time computing is the term used to describe systems which are subject to constraints in time. Typical data warehousing processes are, more often than not, batch processes that are allowed as much time as needed to perform the required transformations and loading processes. Business Intelligence applications demand exceptional performance by the user and people work hard to optimize for performance, yet performance problems rarely incur any serious penalty.

SAS Software is applied in many contexts and different scenarios. Common uses include as the driver for an ETL process (to a data warehouse/mart/hut/superstore) and as a just-in-time reporting tool. What may not be apparent are the opportunities for introducing real time techniques in the heavy lifting world of batch processing and ETL, analytics and of course, just in time reporting and notification (business intelligence).

Leading industry analysts (Gartner, Bloor among them) agree that real time business activity monitoring is the missing link in most BI strategies. We will discuss the ways in which SAS Software is ready (and unready) for this next technological challenge. In this paper, we will take inventory of the possible uses of real time processes throughout the SAS system. Here we will focus on what critical business issues seem best solved by real time agents that cleverly attract the notice of systems administrators and those that create value for business analysts.

### **Rules for Tools - The SAS Utility Primer**

Frank Dilorio

Let's start with the premise that good programmers are lazy by nature. They want to use tools such as formats and ODS for execution-time efficiency or to pretty-up our output, functions to perform calculations, and so on. Another hallmark of a good programmer is a keen eye for pattern recognition. Rather than rewrite basically the same program over and over, they identify similarities and parameterize the program, making it into a general-purpose program, a "utility."

This paper steps through the life cycle of a simple utility. It starts with "naïve" code that doesn't exploit program similarities, then illustrates how a general-purpose utility may be developed. It ends with the initial program becoming a call to a simple, powerful routine in a macro library. The transition from simple, brute-force programming into a compact, general-purpose utility isn't a random event. The last sections of the paper present a set of design principles for utilities.

Although we focus on Base SAS in Version 9.0, the principles and techniques are readily extended across SAS versions and products. The reader will come away from this paper with an appreciation of both the process and the tool set required to build generalized programs.





## SESUG Speaker Sharing Program

### **SAS as a Utility Language: Using SAS for Solving Little Problems**

Gary Schlegelmilch

SAS, from its very conception, was intended to provide methods for determining and handling statistics cleanly, clearly, and effectively. It does so with speed, strength, and versatility. However, the tools that affect its versatility lend themselves with equal ease to nonstatistical projects as well. Small day-to-day tasks – impromptu reports, quick reviews of data, and other small utility programs – are handled in SAS quickly and efficiently.

Because SAS provides minimal data field definition, a modular approach in the DATA step, and clearly defined processes in the PROC step, small tasks can be programmed with a minimum of effort. In addition, the turnaround time will frequently be less than the same routine written with a compiled/linked language. When time is of the essence, SAS may very well be the best candidate.

### **Shuffling, Shifting, and Structuring Using the Program Editor**

Gary Schlegelmilch

Invariably, when we go to a SAS® programming class, we are taught the basics of how the language works and ways to lay out a program. The problem is that once we leave an academic arena and move into the professional one, we don't always develop the habits we need to write readable, maintainable code. Or, we inherit code that was written by someone else who never learned the skills and are required to update it for new requirements. In this paper, we'll discuss some of the basic structuring tools available within the SAS Program Editor® window, and how to use them effectively. This will help even the most advanced programmer, as well as the novice in the business, to both write new code and to update outmoded, unstructured, and just generally bad coding.

### **The SAS Debugging Primer**

Frank Dilorio

Meet an accomplished SAS programmer and you meet someone who's probably learned by making (and fixing) lots of mistakes along the way. The breadth of the SAS System's target applications, the variety of its "dialects" (Base SAS, macro, SCL, IML, SQL), and the quirky procedural/non-procedural environmental mix conspire to make mastery of the SAS System a slippery slope to ascend. Debugging is the art of gracefully recovering and learning from falls during the ascent.

This paper discusses techniques for debugging SAS programs. Its purpose is two-fold. First, it provides behavioral and technical tips for fixing code (how to read error messages in the SAS Log, knowing when there is a problem with the program even if SAS says there isn't, using the DATA step debugger, identifying system options, using PROCs for data validation, using macro variables to control debugging output, etc.) The second focus of the paper is its presentation of design and coding methods that make the programming process more reliable, thus reducing the need for debugging in the first place.

The paper's target audience is relative newcomers to the SAS System. More seasoned users may find or rediscover some of the techniques and features being discussed. Emphasis is placed on Base SAS and the macro language, although the techniques themselves are applicable to SCL and other products.

Last Update: 27Jun2006



## SESUG Speaker Sharing Program

### **SAS 9 Programming Enhancements**

Marje Fecht

Performance improvements are the well-publicized enhancement to SAS 9, but what else has changed that impacts your “SAS Programs”? This Hands-On Workshop explores many new SAS 9 features including:

- new functions that may eliminate the need for complex expressions
- changes and additions to informats and formats
- interface additions and improvements
- new export capabilities
- procedure enhancements
- and more. . .

This workshop is beneficial for Version 8 SAS users who are getting started with Version 9.

### **The Utter “Simplicity?” and Power of the TABULATE Procedure**

Dan Bruns

Well, here we are again Tabulate fans. It never ceases to amaze me how many folks give up on using TABULATE because they cannot quite figure out how to get it to produce the table they want. Once you learn the basics, it is quite simple to produce relatively simple reports/tables. But as you want to produce more complex tables, it does not seem to be quite as simple—especially percentages. During this presentation we will introduce you to some of the more advanced features TABULATE, i.e. percentages, multilevel formats, new statistics, new options, and maybe even a trick or two.

### **The Utter Simplicity of the TABULATE Procedure**

Dan Bruns

Is the TABULATE procedure a mystery to you? Master the mysteries of PROC TABULATE.

Many people have told me they just gave up trying to figure TABULATE out and found another way to get what they needed, which usually took a lot longer. PROC TABULATE is much easier than it looks. Learn how TABULATE can meet your reporting needs.

TABULATE looked very promising when it was first introduced, but it took a lot of experimenting to discover its power and simplicity. This workshop will cover the essentials for effectively using PROC TABULATE to produce meaningful, useful reports. Emphasis will be placed on the TABLE statement which is the heart and soul of the procedure and how to code it to have it give you what YOU want.

The examples will start with a simple application, then progress logically demonstrating how you can control every aspect and stratification of the table and every cell. Adding more options and statements will reveal how simple the effect or impact of the changes on the tables produced.

Last Update: 27Jun2006



**SESUG Speaker Sharing Program**  
**Using BI Tools from SAS for Clinical Reporting**  
Greg Nelson

Tables, figures and listings in the clinical world have always been treated like an artisan's affair – each program has to be carefully crafted to fit the requirements of the protocol and while standard macro libraries, program templates and coding standards have been adopted to varying degrees, the convergence on a standard approach to the generation of tables, figures and listings has been fairly elusive. In this workshop, we will let you decide by understanding just what the Enterprise BI Server offers you and will include a tutorial on the creation of Stored Processes for use in SAS Web Report Studio and the SAS Add-in for Microsoft Office.

**Using ETL Studio for Clinical Research**  
Greg Nelson

Preparing data for use in clinical research has traditionally been relegated to the BASE SAS programmer. With advances in standards like CDISC, FDA Guidance documents (e.g., Item 11), ICH, HL7 and others along with the convergence on clinical data management systems and modernization strategies, could ETL Studio really be used as a tool to help read data from data management to prepare derived datasets and analysis tables? This workshop is designed to give the programmer and manager a look at using ETL Studio for clinical research. In this workshop, we will focus on using ETL Studio to support the business process of reading from data management systems to prepare derived datasets and analysis tables (to any standard, internal or external) or more generally, how to use ETL Studio to support the processes of clinical research.

In this workshop, we will touch on how ETL Studio could be used to read and write to some of the aforementioned standards.

**Web Development: Best Practices for Developing Enterprise Applications**  
Greg Nelson

Software development has a rich history that stems from 40-plus years of tried-and-true practices. From structured programming models to object-oriented methods, the student is often left to her own devices to implement code based on a highly creative -- and subjective -- design process.

In this paper, we will explore some of the foundations of our software development methodologies and apply these to Web programming for SAS applications. We will present coding techniques, lessons learned from our experiences as architects and experiences proven to be too painful to be ignored.

Examples will include reference to HTML, JavaScript, SAS/IntrNet®, JavaServer Pages, XML and ODS.



## SESUG Speaker Sharing Program

### Writing the "Best" Program: The How and When of Efficient Programming

Frank Dilorio

It's relatively easy to write programs that optimize the use of CPU and other machine resources. There is a large and continually growing body of literature on the subject. What isn't as straightforward is knowing **\*\*when\*\*** to employ the techniques - blind implementation of tuning techniques is often not required by the task at hand and can sometimes even be counterproductive.

This paper addresses both the "how to" and "when to" aspects of writing efficient programs. It describes design and coding techniques that conserve hardware resource usage. It also identifies other, non-machine implications of their usage that could dissuade the programmer from their use. For example, using temporary array elements is more efficient than using named elements but has the documented-but-obscure behavior of retaining values across observations. Maintenance of such code by other than "seasoned" and up to date programmers can be unexpectedly problematic.

The concept of efficiency used in the paper includes all aspects of the program life cycle. We apply the "how and when" question to system design issues, system startup, DATA steps, procedures, and macros. Emphasis is on Base SAS software. The reader should finish the paper comfortable with the idea that the "best" program is not always the one that minimizes hardware resources.

### XML and SAS: An Advanced Tutorial

Greg Nelson

One of the goals for SAS applications developers has been to develop three-tier and n-tier applications where the application logic (business rules) is separate from the data, which, in turn, is isolated from the user interface. In a previous paper (Barnes Nelson, 1999) we discussed how to implement this logic separation using the SAS Component Language. This paper extends that line of thinking by introducing SAS developers to XML. eXtensible Markup Language, or XML, is a protocol of sorts that can be described as a technique for separating data from its presentation. In this paper, we will discuss XML in the context of SAS applications and how it can be used in the preparation and presentation of data. We will explore some of the features of XML that makes it a good partner for SAS-based applications.