



SESUG Speaker Sharing Program

To arrange for a SESUG speaker, contact Marje Fecht at Marje.Fecht@prowerk.com

Speaker:

Gary Schlegelmilch
Census Bureau

Bio:

Gary Schlegelmilch has been enthusiastically working in SAS for eight years. His current position with the Census Bureau finds him developing software for various economic surveys, primarily using Base SAS, SAS CONNECT, and Macros on UNIX and VMS. His programming career spans four decades, and he has used and informally taught several Computer languages. Gary presents papers at local, regional, and international SAS conferences, and chaired the SouthEast SAS Users Group conference in St. Pete Beach, Florida in 2003. He is an active member of the SESUG Executive Council, and is chairing the Posters section at SESUG 2006 in Atlanta.

Presentation Topics:

- Programming for Failure: When Programs Faw Down and Go Boom
- Shuffling, Shifting, and Structuring Using the Program Editor
- SAS as a Utility Language: Using SAS for Solving Little Problems



SESUG Speaker Sharing Program

Abstracts:

Programming for Failure: When Programs Faw Down and Go Boom

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.

Shuffling, Shifting, and Structuring Using the Program Editor

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.

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

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.