



SESUG Speaker Sharing Program

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

Speaker:

Neil Howard
i3 Data Services

Bio:

Neil Howard is Manager of Statistical Programming at i3 Data Services in Basking Ridge, NJ; she was with Pfizer previously. A SAS user for more than twenty years, Neil has been an invited speaker since 1983 on such topics as: efficiency techniques, DATA step processing and internals, advanced DATA step techniques, testing and validation, graphics, effective presentations, and interviewing/hiring SAS programmers. She was a contract instructor for SAS Institute for seven years teaching fundamentals, programming, macro, report writing, graphics and the annotate facility. She has been a member of the SUGI Executive Committee since 1993 and was proud to chair SUGI 20 in Orlando. She is currently the Academic Chair for SESUG '04 to be held in Nashville in October 2004.

Presentation Topics:

- How SAS Thinks
 - OR Why the DATA Step Does What It Does
 - OR Anatomy of a DATA Step
 - OR DATA Step Essentials
- 'LAG with a WHERE' and other DATA Step Stories
- Testing and Validating SAS Programs
 - OR Beyond Debugging: Program Validation
 - OR Programming on Purpose
- The Ultimate Match Merge: Hiring the Best SAS Programmers
- Advanced DATA Step Techniques



SESUG Speaker Sharing Program

Abstracts:

How SAS Thinks OR
Why the DATA Step Does What It Does OR
Anatomy of a DATA Step OR
DATA Step Essentials

The DATA step is the most powerful tool in the SAS System. Understanding the internals of DATA step processing, what is happening and why, is crucial in controlling your programs. This tutorial provides answers to the DATA step mysteries: what is the Program Data Vector (PDV), what are automatic SAS variables and how are they used, what is the SAS Supervisor, why you need to understand the internals of DATA step processing, what happens at program compile time, what's actually happening at execution time, how the DATA step functions as a read/write loop, what is the logic behind a MERGE, how are variable attributes captured and stored, and more. By understanding DATA step processing, you can debug your programs and interpret your results with confidence.

'LAG with a WHERE' and other DATA Step Stories

This tutorial illustrates the importance of understanding the internals of DATA step processing, particularly when you invoke mysterious functions like LAG, code RETAIN statements, rely on WHERE constructs, and start combining various features of the DATA step language. This anthology includes such spellbinding stories as "LAG with a WHERE", "When RETAIN Doesn't Retain", "To LAG or to LEAD", "Don't Order My Variables Around", "A DIFferent LAG", and "The Case of the Missing Values".

Testing and Validating SAS Programs OR
Beyond Debugging: Program Validation OR
Programming on Purpose

"Act in haste and repent at leisure; code too soon, and debug forever." Raymond Kennington

Overheard at an interview for a SAS programming position: "But you don't have to test SAS programs!!!" As the interviewers quickly escort the confused candidate out the door, they recall how often it is assumed that a fourth generation language "does so much for you" that you don't have to test the code. The SAS system is easy to use, and the learning curve to productivity is relatively short. But SAS is just as easy to ABUSE. Programmers and analysts must not lose sight of the indisputable facts: data is seldom clean, logic is too often faulty, and fingers walk clumsily over keyboards. Condition codes and clean logs are not accurate indicators of successful programs.

Good debuggers make good programmers. And good analysts and problem-solvers make good programmers. And just because a SAS® program is free of errors, warnings, notes, and bugs does not guarantee that the program is doing what it is supposed to do. This paper addresses the process of debugging, testing and validating SAS programs and the tools available in the SAS language to facilitate the process. It covers techniques for ensuring that the logic and intent of the program is correct, that the requirements and design specifications are met, and that data errors are detected. Since as much as 80% of a programmer's time is invested in testing and validation, it's important to develop a personal approach to testing and a routine testing methodology in your group.



SESUG Speaker Sharing Program

The Ultimate Match Merge: Hiring the Best SAS Programmers

'The closest to perfection a person ever becomes is when he fills out a job application form.' (Stanley J. Randall)

You are interviewing a candidate in a sharp navy suit, with an impressive resume and ten years of SAS experience. But, how do you tell if this ten years of experience is really ten years, or the same year over and over again? If, indeed, past behavior or past job performance is the best indicator of future performance, how is the SAS community probing for the facts? What are our assessment criteria?

The author surveyed select SAS programmers, analysts, trainers, human resource experts, and managers as a reality check into our performance as interviewers and our experience as interviewees. What techniques are being used to interview and assess SAS talent?

This paper explores the results of this survey, looking at the following interview techniques: identifying resume red flags, conducting useful phone interviews, structuring general face-to-face interviews, "measuring" intangibles, proficiency testing, and code walk-throughs. The paper includes a collection of technical SAS interview questions gathered from past SUGI papers, SAS users and SAS-L contributors; and discusses the development of probing technical questions, metrics, and standards.

Advanced DATA Step Techniques

Understanding the intricacies of the DATA step can make all the difference in your SAS programming. This tutorial focuses on the more advanced techniques that capitalize on the power of the DATA step and working with (and around) the default actions. Topics include:

- compile versus execution time activities;
- organizing your data to maximize execution;
- data defaults, data conversions;
- missing values, formatting values;
- ordering variables;
- functions for:
 - editing data,
 - assigning values,
 - shortening expressions
 - performing table lookup;
- data management;
- effectively creating SAS data sets;
- the logic of the MERGE;
- efficiency techniques.

'Real world' examples are presented to illustrate each topic.