



The Missing Semicolon™

Volume 9, Number 1

Fall 2007

Enterprise Guide for the New SAS User

Traditional SAS can be daunting for a new user. The editor window is blank. The cursor is flashing. There are hundreds of procedures to choose from and countless minute syntax details to be aware of. A single misplaced parenthesis can turn an otherwise perfect program into error-generating junk. Under such conditions, an analyst ripe to benefit from the vast data analysis capabilities of SAS can easily wither into a state of unproductiveness. Short of hiring a SAS consultant to perch on the edge of your desk and coax you through your first programs, what can be done? One solution is SAS Enterprise Guide.

Enterprise Guide is SAS with training wheels. It is SAS with a friendly face, or at least a friendly, visually-intuitive graphical interface. The first thing you may notice about Enterprise Guide is how familiar it seems. As you point, click, drag and drop, it is easy to forget that you are using a sophisticated statistical analysis package. The menu- and wizard- heavy interface gives it the feel of every other Windows-based application. SAS code is built behind the scenes and can be viewed, altered, and exported, but it is perfectly acceptable to ignore all matters programmatic and interact with Enterprise Guide purely on the level of the user-friendly interface. Enterprise Guide hides the sometimes ugly world of coding behind a much prettier point-and-click façade.

The sheer convenience of Enterprise Guide does not come without a trade-off cost, however. The flexibility of a programming language cannot be adequately captured in the cookie cutter format of a front-end like Enterprise Guide. Enterprise Guide developers were faced with the challenge of distilling SAS down to its most commonly used procedures and the most essential features within these procedures. In all, about 80 SAS procedures play a role in Enterprise Guide's Task Wizards, with varying degrees of functionality and flexibility. Is this enough, or is Enterprise Guide doomed to be a vastly inferior SAS wannabe? The answer depends on who you are and what you are trying to do with SAS.

An experienced SAS programmer accustomed to using SAS in highly innovative ways will find Enterprise Guide extremely limiting. On the other hand, someone interested in the more everyday, basic uses of SAS will be pleasantly surprised by the wide range of what can be done using Enterprise Guide. Data can be pulled into Enterprise Guide from a variety of sources, including SAS datasets, Microsoft Excel spreadsheets, and text files. Enterprise Guide will tell you essentially anything you might want to know about a data set, ranging from simple summary functions like sums and frequencies to more complicated statistical objects like distributions and time series analyses. Data sets can be manipulated easily in a variety of ways including sorting, joining, subsetting, and calculating new variables. Enterprise Guide may not be ideal for constructing highly complex systems of programs, but it is the ad hoc analyst's dream.

Enterprise Guide is also a great tool for creating reports, especially when time is a limiting factor. In Enterprise Guide, results can be generated in HTML, PDF or RTF form just by changing a single setting, while in SAS, specialized programming would be necessary for output in these forms. Furthermore, the appearance of

Continued on page 2.....



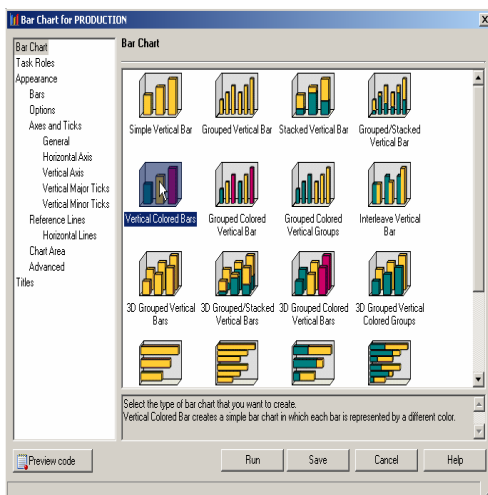
SYSTEMS SEMINAR CONSULTANTS, INC.

2997 Yarmouth Greenway Drive
Madison, WI 53711

www.sys-seminar.com
train@sys-seminar.com
1-800-997-7081

In This Issue

Enterprise Guide for the New User	Page 1
Kathleen Nosal.....	Page 1
Enterprise Guide Class	Page 2
.....	Page 2
President's Letter	Page 2
Steve First.....	Page 2
SAS and Sudoku	Page 3
Steve First.....	Page 3
Quick Tip	Page 3
Rosalind Gusinow.....	Page 3
Placement Services	Page 4
Open Positions	Page 5
Erin Ward.....	Page 5
Quick Tip	Page 5
.....	Page 5
Technical Interview Tips	Page 6
Erin Ward.....	Page 6
Advantages of a Datamart	Page 6
.....	Page 6
SAS Online Training	Page 6
SAS Training Solutions	Page 7
.....	Page 7
Technical Credit and Recognition	Page 7



Letter From the President



Dear SAS User:

After several sluggish years, the IT community is enjoying the best economy in several years. We have seen a great increase in open IT positions, contract work, and training.

In the SAS world, the traditional SAS applications are more vibrant and dependable than ever, and

exciting new enhancements and products are becoming more and more popular.

SAS Business Intelligence is now one of the most popular BI products. The SAS developers assure us that they are well aware of the features of competitive BI systems, and they are working very hard to become the number one BI provider.

Systems Seminar Consultants has been working with clients to implement and optimized the SAS Business Intelligence Suite. We have also just added a new SAS Enterprise Guide class to our curriculum.

Enterprise Guide 4.0 allows the casual users to generate SAS code via wizards and menus, and it offers more functionality than ever before. Experienced SAS users can also use EG to alter generated code, or they can use it to write and alter SAS programs. EG gives welcome shortcuts that simplify management of data and libraries. Libraries, SAS datasets, Excel data, and raw data and can be very confusing to many new SAS users, and EG does a nice job of eliminating tedious coding for those new users. We are very excited about the addition of this innovative product to our training curriculum.

We enjoy helping clients get the most out of their SAS products, whether it be just Base SAS to the vertical BI line!



SYSTEMS SEMINAR CONSULTANTS, INC.

Copyright © 2007 Systems Seminar Consultants, Inc. Madison, WI. All rights reserved.
Printed in USA. The Missing Semicolon is a trademark of Systems Seminar Consultants, Inc.
SAS, SAS/IntrNet, and SAS/ACCESS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Enterprise Guide for the New SAS User

Continued from page 1

Enterprise Guide output can easily be changed using customizable style sheets. Results from multiple tasks can be combined into a single HTML document that can also be customized in various ways. Creating a polished looking report requires minimal time and effort in Enterprise Guide. Why waste valuable time fussing over how output will look when you could instead spend time on analytics?

Eventually, an Enterprise guide user may want to use a feature of SAS that is not available through an Enterprise Guide wizard. Fortunately, the transition from Enterprise Guide to SAS can be extremely smooth. Enterprise guide is an effective way to learn many SAS concepts. Through Enterprise Guide, you can become familiar with what a procedure does, what variables are required for input, and what type of output it produces. Transferring this knowledge to SAS is mainly a matter of learning syntax rather than learning concepts all over again. Conveniently, SAS code can be written and tested from within the Enterprise guide environment, using the included fully functional SAS code editor. A more timid beginning programmer may want to try modifying code generated by Enterprise Guide rather than plunging right into writing SAS code from scratch. This option allows for a more gradual introduction to programming and provides an opportunity to mimic the syntax already in place.

Enterprise Guide is an ideal product for the SAS beginner. It is a portal to the power of SAS that sidesteps many of the difficulties that have previously intimidated new SAS users. Enterprise Guide fills a gap in the statistical software market by superbly balancing functionality with accessibility.



Introduction to SAS Enterprise Guide

- ◆ Create projects
- ◆ Import, organize, format, and sort data
- ◆ Join and filter data
- ◆ Calculate basic statistics
- ◆ Create/customize summary reports and graphs
- ◆ Design documents and reports
- ◆ Create flexible tasks and queries
- ◆ Automate and schedule processes
- ◆ Review and modify code
- ◆ Import and export SAS code

Call 1-800-997-7081 to discuss details!

Sudoku and SAS Arrays, Part 1

Sudoku is a wildly popular and highly addictive number puzzle. Most newspapers print daily puzzles with difficulty ratings ranging from simple to demonic. There are entire books, societies, and web pages devoted to Sudoku. The puzzle craze reaches across age and position levels. I personally have seen small children as well as members of Congress working on the puzzles.

Sudoku is a variation of what mathematicians call a “Latin square” where all the digits 1-9 across a row and column must appear only one time. Sudoku adds that within each 3 x 3 “quadrant” the digits again can only occur once. The puzzle is presented with some digits filled in, and it is up to the individual to solve the missing cells.

			7					
	9		5		6			8
		8	4		1	2		
	5	9				8	4	
7								6
	2	3				5	7	
		5	3		7	4		
	1		6		8			9
			1					

Strategies for solving the puzzle range from simple “trial and error” to medium and sophisticated algorithms.

So what does this have to do with SAS? Having an interest in this puzzle, I accepted the challenge of writing a Sudoku-solving program. Since SAS has a good array structure, this is one approach that can be used. I must note that my program is very basic and only implements two algorithms. As such, it can only solve very basic puzzles. Since this can be a very long program, I will describe only the data structures and the mainline portion at this time. In a future issue, I will show the algorithms within the program and perhaps write a trial and error program implementing Knuth's “Dancing Links Algorithm X”. If you are interested, you may copy my source program which is available on our web site.

Several newspapers have online Sudoku puzzles. I especially like the Minneapolis Star Tribune's and use their style of displaying the puzzle. They use a big bold letter to represent the given and solved digits and display the nine possible digits in each cell. As digits become ineligible, they are blanked out. The cell is solved when only a single digit remains in each cell, and the puzzle is solved when all cells are solved.

Continued on page 4

QUICK TIP

If you need the name of a flat file as it resides on your system, you can obtain it using one of the following methods.

In the following examples, we obtain the name and then create a macro variable with it's value. This is then used in a later step as part of an email note to those people who will be using the newly created file.

1) Use the JFCB option on the INFILE statement:

```
DATA _NULL_;
    INFILE FLATIN JFCB=JFCB;
    DSNAME=SUBSTR(JFCB,1,44);
    CALL SYMPUT('JFCB',DSNAME);
    STOP;
RUN;
%PUT *** JFCB=&JFCB ***;
```

NOTE: 0 records were read from the infile FLATIN.
NOTE: The DATA statement used 0.00 CPU seconds.

```
*** JFCB=SSC.TEST.INFILE.FLAT.GO100V00 ***
```

2) Use the PATHNAME function:

```
DATA _NULL_;
    PATH=PATHNAME("FLATIN");
    CALL SYMPUT('JFCB',PATH);
    PUT '**** ' PATH= ' ****';
RUN;
```

```
**** PATH=SSC.TEST.INFILE.FLAT.GO100V00 ****
```

NOTE: The DATA statement used 0.00 CPU seconds.



Below is a display showing the given digits and the possibilities for the rest.

Step1 – initial load

1 2 3 1 2 3 5 1 2 3 1 2 3 8 1 2 3	4 5 6 4 5 6 9 4 5 6 4 5 6 8 4 5 6	7 8 9 7 8 9	1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
4 2	5	9 8 6	
1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
4 2			
1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	5
1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
4 2	6	9 8 6	5
1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
4 2	9 7 5 2 1	4 5 6	
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
6	4	5	2 9
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	
8	7	2	6 4 5 6
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3	4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6 4 5 6	7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9 7 8 9	

The data structures used for the puzzle are as follows. There are 81 character variables defined to correspond with the 81 cells in a puzzle. Each variable will contain all 9 digits to start. As the puzzle is solved, unused digits will be eliminated until only 1 remains. Since all algorithms will work the same whether dealing with a row, column, or quad, I decided to set up 3 different arrays to reference these 81 variables. In that way, if I am working with a row, I can use the row array; when dealing with columns, the column array, etc. In each case, the different arrays actually refer to the same underlying data values. There is an intermediate work array where the actual searching and changing takes place. The changes are then moved back to the appropriate row, column, or quad.

Placement Services

We provide placement services for our clients in:

- ◆ Financial
- ◆ Marketing
- ◆ Healthcare
- ◆ Insurance

Let our experienced recruiters assist you in finding the ideal candidate.

Call Erin at (608)-278-9964, ext 313.

Below is the mainline portion of my program along with the reporting section. The algorithm sections will be shown in a future issue, so please stay tuned.

```

/*****
* sudoku program that implements naked and
* hidden single algorithm */
*****/

%let graphics=y; /*graphics available? */
/*include annotate macros */
%inc 'C:\Program Files\SAS\SAS 9.1\core\sasmacro\annomac.sas';
goptions hpos=60 vpos=36 target=winprtc;

options mprint; options nodate nonumber ls=132;title;
/*system options */
data sudoku /*create two datasets */

annods(keep=pagenochar style function color
XSYS YSYS HYSYS
when position X Y size linetype line text);

Infile 'C:\Documents and Settings\sfirst\Desktop\sudoku\
oct15_lv12.dat';
Infile 'C:\Documents and Settings\sfirst\Desktop\sudoku\
sep7_1v14.dat';

```

Eighty-one variables, indexed by row and column number, represent the cells in the sudoku. For example, the variable 'r6c3' corresponds to the cell in the sixth row and third column. The variables have a length of nine in order to store the nine digits that, a priori, the cell might contain. Later in the program, digits are eliminated as they are systematically ruled out as values for the cell.

```

/* define 81 variables that initial contain 9 digits each */

length r1c1 r1c2 r1c3 r1c4 r1c5 r1c6 r1c7 r1c8 r1c9 $9;
length r2c1 r2c2 r2c3 r2c4 r2c5 r2c6 r2c7 r2c8 r2c9 $9;
length r3c1 r3c2 r3c3 r3c4 r3c5 r3c6 r3c7 r3c8 r3c9 $9;
length r4c1 r4c2 r4c3 r4c4 r4c5 r4c6 r4c7 r4c8 r4c9 $9;
length r5c1 r5c2 r5c3 r5c4 r5c5 r5c6 r5c7 r5c8 r5c9 $9;
length r6c1 r6c2 r6c3 r6c4 r6c5 r6c6 r6c7 r6c8 r6c9 $9;
length r7c1 r7c2 r7c3 r7c4 r7c5 r7c6 r7c7 r7c8 r7c9 $9;
length r8c1 r8c2 r8c3 r8c4 r8c5 r8c6 r8c7 r8c8 r8c9 $9;
length r9c1 r9c2 r9c3 r9c4 r9c5 r9c6 r9c7 r9c8 r9c9 $9;

```

An array is defined that contains the 81 variables.

```

Array all181{9,9} $ r1c1--r9c9;
/* redefine all 81 vars into an array */

```

Three arrays structure the variables as rows, columns, and quadrants. The arrays all reference the same 81 variables that are defined above. While the variables store the comprehensive list of potential digits for a cell, the arrays allow the program to access the variables in meaningful groups.

```

Array rowa {9,9} $ r1c1--r1c9
/* redefine all 81 vars into row array */

r2c1--r2c9
r3c1--r3c9
r4c1--r4c9
r5c1--r5c9
r6c1--r6c9
r7c1--r7c9
r8c1--r8c9
r9c1--r9c9;

```

Continued

```
/* redefine all 81 vars into column array */
```

```
Array cola{9,9} $ r1c1 r2c1 r3c1 r4c1 r5c1 r6c1 r7c1 r8c1 r9c1  
r1c2 r2c2 r3c2 r4c2 r5c2 r6c2 r7c2 r8c2 r9c2  
r1c3 r2c3 r3c3 r4c3 r5c3 r6c3 r7c3 r8c3 r9c3  
r1c4 r2c4 r3c4 r4c4 r5c4 r6c4 r7c4 r8c4 r9c4  
r1c5 r2c5 r3c5 r4c5 r5c5 r6c5 r7c5 r8c5 r9c5  
r1c6 r2c6 r3c6 r4c6 r5c6 r6c6 r7c6 r8c6 r9c6  
r1c7 r2c7 r3c7 r4c7 r5c7 r6c7 r7c7 r8c7 r9c7  
r1c8 r2c8 r3c8 r4c8 r5c8 r6c8 r7c8 r8c8 r9c8  
r1c9 r2c9 r3c9 r4c9 r5c9 r6c9 r7c9 r8c9 r9c9;
```

```
/* redefine all 81 vars into quad array */
```

```
Array quada{9,9}$ r1c1 r1c2 r1c3 r2c1 r2c2 r2c3 r3c1 r3c2 r3c3  
r1c4 r1c5 r1c6 r2c4 r2c5 r2c6 r3c4 r3c5 r3c6  
r1c7 r1c8 r1c9 r2c7 r2c8 r2c9 r3c7 r3c8 r3c9  
r4c1 r4c2 r4c3 r5c1 r5c2 r5c3 r6c1 r6c2 r6c3  
r4c4 r4c5 r4c6 r5c4 r5c5 r5c6 r6c4 r6c5 r6c6  
r4c7 r4c8 r4c9 r5c7 r5c8 r5c9 r6c7 r6c8 r6c9  
r7c1 r7c2 r7c3 r8c1 r8c2 r8c3 r9c1 r9c2 r9c3  
r7c4 r7c5 r7c6 r8c4 r8c5 r8c6 r9c4 r9c5 r9c6  
r7c7 r7c8 r7c9 r8c7 r8c8 r8c9 r9c7 r9c8 r9c9
```

The work81 variable is an 81 byte work area. This variable temporarily stores a single row, column, or quadrant as the given structure is analyzed in the mainline program.

```
length work81 $ 81 ;  
/* 81 byte work area to contain one structure */  
array cnta{9}cnt_1-cnt_9;  
/* array to count digit occurrences */
```

Our Open Positions

Permanent Positions

Credit Risk Analyst	Minneapolis
Applications Systems Analyst	Madison, WI
Senior Quantitative Analyst	Madison, WI
Unix Systems Administrator	Madison, WI
Data Analyst	Madison, WI
Assistant Manager, Marketing	Chicago, IL
Project Manager	Madison, WI
SAS Consultant	Madison, WI

Contract Positions

Programmer Analyst	Madison, WI
Visual Basic Technical Lead	Madison, WI

To apply for one of these positions,
send your resume to
careers@sys-seminar.com
or
call Erin at (608)-278-9964, ext 313.

QUICK TIP

The abbreviated variable list `_ALL_` may be used on the BY statement in the SORT procedure to use all variables as the sort key.

```
PROC SORT DATA=SOFTSALE;  
BY _ALL_;  
RUN;
```

The sort key will be made up of all variables on the data set in the order in which they were created. This is a quick work around with the NODUPRECS option to eliminate duplicate records which may not be contiguous records with a limited sort key.



The mainline portion of the program is built on three loops. The two inner loops cycle through the 27 structures in the Sudoku - rows one through nine, columns one through nine, and quadrants one through nine. When a structure is up to bat, the current values of the nine variables it references are loaded into the work81 variable. Two subroutines analyze the work81 variable and determine whether any digits may be ruled out in the nine substrings that represent individual cells. (I will discuss the algorithms in detail in the second installment of the article.) Work81 is updated accordingly, and the new set of possible values is loaded back into the variables. After each cycle of 27 structures, the outer loop checks whether any changes have occurred in the variables, based on the value of a flag. If nothing has changed, the mainline portion terminates. Otherwise, the inner loops iterate through another cycle. At the end of every cycle, the mainline portion calls a reporting subroutine that generates an updated sudoku grid.

```
/*mainline portion*/  
link load_initial; /*load initial puzzle*/  
change='Y'; /*set change flag*/  
do until (change='N'); /*keep going until no more changes*/  
change='N'; /*change flag to no*/  
i+1; /*times looped thru*/  
do Structure='R','C','Q'; /*do rows,cols,then quads*/  
do struct_no = 1 to 9; /*do all 9 rows,cols,quads */  
link loadwork81; /*move structure to work area */  
link naked_singles; /*look for naked singles */  
link hidden_singles; /*then look for hidden singles*/  
link unloadwork81; /*move work area back to arrays */  
end; /*end of inner loop */  
/*title with interation, struct*/  
rtitle= 'Iteration ' !! put(i,3.) !! ' ' !!  
structure !! ' Structure Report';  
link report; /*print results so far */  
end; /*next structure */  
end; /*next loop if still changed */  
return; /*end of mainline */
```

The load_initial subroutine is called by the mainline portion before it enters the loops. This subroutine is responsible for reading in the sudoku and initializing variables.

Continued

```

load_initial:          /* read in puzzle, init blk cells */
do row=1 to 9;          /* read 9 rows */
  do col=1 to 9;        /* and 9 columns */
    input all81{row,col} $char1. @;
    /* input array element from file */
    if all81{row,col} in('.', ' ') then
      /* if missing or blank? */
      all81{row,col}='123456789';
      /* initialize to all 9 possibles */
    end;
  /* end of col loop */
end;

input;                 /* clear input pointer */
end;                   /* next row */

rtitle='Step1 - initial load '; /* set title */
link report;          /* print initial puzzle */
return;               /* end of load */

```

The loadwork81 subroutine, also called by the mainline portion, moves variables from the arrays into the work81 variable.

```

Loadwork81:
          /* load work with cells from r,c,q */
do cell=1 to 9; /* 9 cells, each 9 digits long */
  fposition=((cell-1)*9)+1; /* cell 1 to pos 1, 2 to 10 etc */
  if structure==Q= then /* if quad? */
    cell_in=quada(struct_no,cell); /* move cell from quad array */
  else if structure==R= then /* same for rows */
    cell_in=rowa(struct_no,cell);
  else if structure==C= then /* and cols */
    cell_in=cola(struct_no,cell);
  substr(work81,fposition,9)=cell_in; /* move cell in to work area */
end; /* next cell */

save_work81=work81; /* save work to look for changes */
return; /* end of load work

```

To Be Continued in Next Issue



What are the Advantages of a Datamart?

Standardize Core Business Data

Let SSC Help Design your Datamart

- ◆ **Consistent**-A dimensional model enforces naming conventions, standardizes measurements and attributes such as location, customer, and time
- ◆ **Scalable**-On a go-forward basis, additional resources for space allocation are readily quantifiable
- ◆ **Flexible** - Dimensions can be gracefully added for new measurement attributes
- ◆ **Standardized**-Utilize an industry-standard format that is easy to query various IT tools for quick answers to business questions

Call 1-800-997-7081 to discuss details!

Technical Interview Tips

Do Your Research-Prior to your interview, research company history and profile. Also do some research on the industry, learning about past/current trends.

Know Your Career Path-Be prepared to discuss goals and what path you would like to see your career take.

Listen-Pay attention to the questions you are being asked. Stay on track and respond with more than a “yes” or “no” answer. Be detailed in your answer but avoid rambling.

Emphasize Your Skills-Discuss your strengths and be honest about skills that you may not have; follow up with how quickly you can learn.

References-Be prepared to present at least two professional references. Contact your references prior to the interview so that they can be prepared for a phone call from the prospective employer.

Follow up-Send a thank you after your interview. If you corresponded by email, it is acceptable to send your thank you electronically. If the company is more formal, you will want to send your note by mail. Be sure to recap the highlights of your conversation, send any information that you may have needed to check on, and finally express your enthusiasm about the opportunity and how your skills will benefit the company.



New Online Training!

25 Years of SAS Training Experience

2008 Online Training Schedule

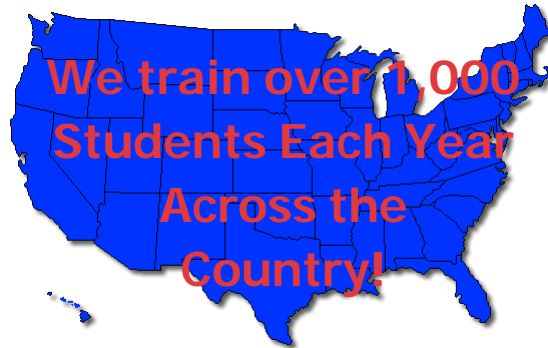
Introduction to SAS	January 28-30
SAS Enterprise Guide	January 31-February 1
SAS Macros	March 11-12
Tips, Tricks, & SAS Techniques	February 14-15
SAS Efficiencies	March 13
SAS Report Writing	February 11-12
Introduction to PROC Report	February 13
The SAS SQL Procedure	February 28
Advanced SAS	February 25-27

**For questions or registration,
Call 1-800-997-7081!**

SAS Training Solutions at Your Business Location

Complimentary Follow-up Help Desk
Competitive Prices

Customized Materials & Courses
Quality Training Nationwide



Our Newest Class...
Enterprise Guide Version 4

Interested in training...Unsure of how to coordinate?

- ◆ Have your SAS users take our training survey at www.sys-seminar.com/sscsurvey.php
- ◆ We'll compile reports to **minimize expenses & maximize results**

View our Course Catalog at www.sys-seminar.com

Call 1-800-997-7081 ext. 306 to discover your SAS training solution.

Technical Credit and Recognition



Steve First
President



Katie Ronk
Director of Operations
Editor



Jennifer First
Office Manager
Editor



Teresa Schudrowitz
Project Manager



Thomas Miron
Senior Consultant



Rosalind Gusinow
Sr. Trainer/Consultant



Kathleen Nosal
Programmer



Erin Ward
Placement Coordinator