Paper 24-27

Optimizing SAS® Version 8 in a Windows Environment – from the User Interface to Automated Document Production

Roger D. Muller, Ph.D., Elanco Animal Health, Indianapolis, IN

ABSTRACT

The Microsoft Windows and the SAS® version 8 environments have been fine-tuned to work harmoniously with each other and provide for efficient program development and quality report writing experience by taking advantage of features that each product provides. Project organization is provided by a common file structure across projects. The structure provides for SAS code, SAS output, SAS data sets, and collateral files such as E-mail, Microsoft's Powerpoint®, Excel®, Access® and Word® files, graphic files, etc. on a project-by-project basis. The structure optimizes the utilization of the SAS Explorer. Output from SAS is moved in a semi-automated technique to Microsoft Office documents and enhanced and indexed by VBA macros. High quality graphics from SAS/GRAPH® are placed in Office documents as EMF (enhanced metafiles).   The ODS facility in SAS is used for creating RTF files that are incorporated in Microsoft Word while proc printto is used to prepare text files.

INTRODUCTION

The default software installation for SAS has many options that may be enabled to improve the interaction of the user with both SAS and the Microsoft Office series of products. This paper provides a guide to improving both the Windows and SAS environments and provides a number of working techniques that organize SAS and other files on a project basis. The projects being undertaken are typical of those of a statistician or a report-writing analyst – retrieve data from databases, make calculations and summarizations, write text reports and create graphics. The examples in this paper were prepared on a computer having Windows 2000, SAS version 8.2, and Microsoft Office XP (2002) with Visual Basic for Applications 6.3.

LAUNCHING SAS

SAS should be launched in a manner that provides for convenient access to all data files, report files, program code etc. To point SAS to specific storage folders requires that multiple icons be available with appropriate identification and launch instructions. These icons could be installed in a number of different ways: (1) scattered across the desktop, (2) in a folder on the desktop, (3) in the start programs menu, or (4) preferably in a tool bar (sample completed in Figure 1). To build a toolbar, place the mouse in the system tray to the right of any toolbars that might exist and hit the right mouse button, select add a toolbar and proceed to build the structure you desire (Figure 2). The user will need knowledge on building shortcuts at this point.

The SAS command (target) to be placed on the shortcut for each launch icon is rather lengthy (Figure 3). A complete command statement in the target field for the minimum launch is to specify the configuration folder and the initial launch folder (Figure 4).

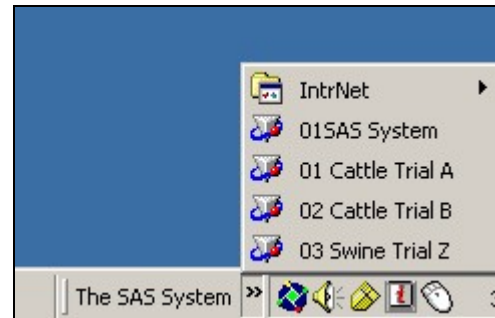**Figure 1. Finished Launch Toolbar for SAS Projects.**

Figure 2. Building a Toolbar (Following a Right Click in the System Tray).
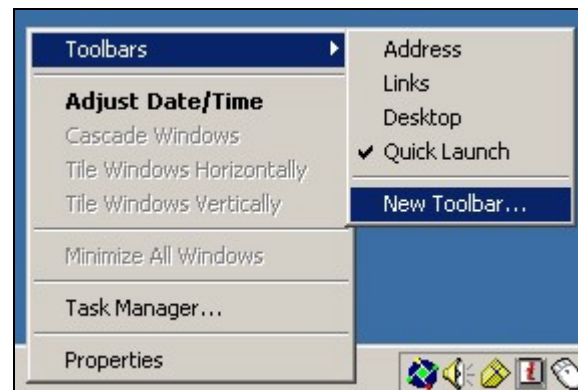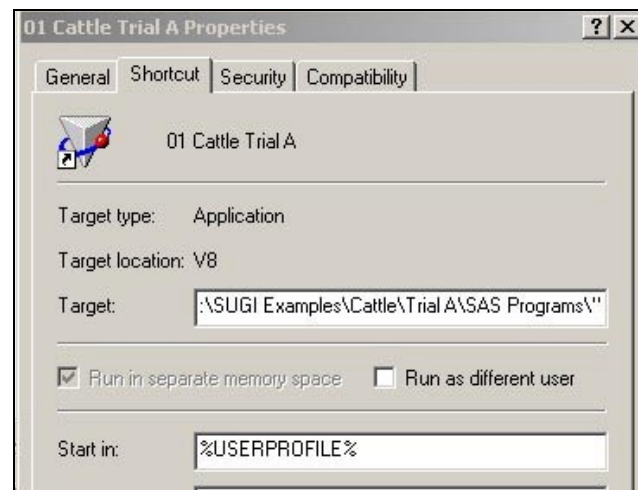

Figure 3. SAS Project Launch Shortcut.

Figure 4. Full Text for Target in Shortcut (Specifying Configuration and Initial Folder).

```
"C:\Program Files\SAS Institute\SAS\V8\sas.exe" -CONFIG "c:\Program Files\SAS
Institute\SAS\V8\SASV8.CFG" -SASINITIALFOLDER "D:\SUGI
Examples\Cattle\Trial A\SAS Programs\"
```

Once SAS has been launched, the folder for storage of SAS programming code is shown in the lower right of the SAS application work space (SASINITIALFOLDER). All file open and file save operations will be directed to this folder by default (Figure 5).

Figure 5. Indicator of Initial Folder (Current Folder) in SAS Application Work Space.

THE SAS EXPLORER

Modifications made to the SAS Explorer Window are noted in Figure 6. **Important: these changes must be made from the SAS menus while the Explorer is selected. The SAS menus change depending on which window is selected**. Enhancements include: (A) Changing the view from Icons to Details, (B) Enabling View Show Tree, (C) Adding Favorite Folders to the display and (D) adding Results and My Computer to the display.

Figure 6. SAS Explorer Window With Numerous Modifications From the Default Installation
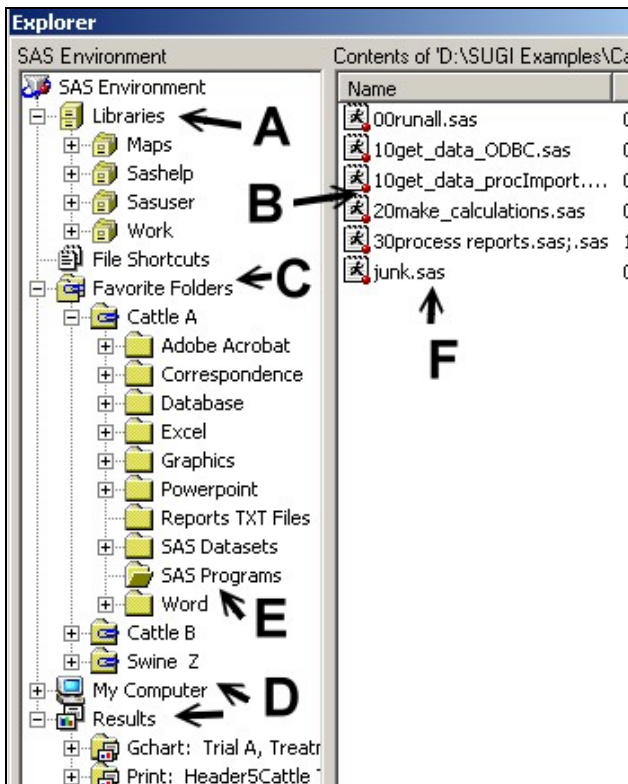
Details and Show Tree are enabled from the View menu when the SAS Explorer window is selected. Adding Favorite Folders,

My Computers and Results is accomplished by: (a) selecting the SAS Explorer window, (b) selecting Tools Options Explorer from the Menu bar (c) selecting "Initialization" in the window that appears (Figure 7, point 1) and (d) changing the description of the registered types from "Off" to "On" (Figure 7, point 2).

Figure 7. Setting SAS Explorer Options to Show Favorite Folders and Results.
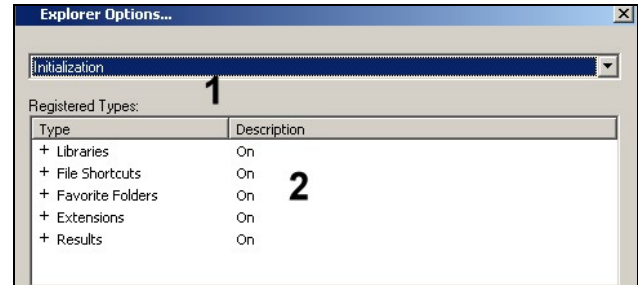
ADDITIONAL SAS EXPLORER OPTIONS

SAS datasets in libraries such as work can be viewed by double clicking on them in the SAS Explorer. A View Table facility allows you to look at the data table or the metadata (variable names, lengths, formats, labels, etc). Unfortunately (or fortunately depending on your viewpoint), the View Table display comes up showing variable labels rather than variable names as column headings   To change this setting, (a) select the Explorer window, (b) from the menu bar, select Tools Options Explorer Members Table and Edit (Figure 8).   The command for the default action (&open) needs to be changed by adding "colheading=name" as show in Figure 9.

Figure 8. Changing Explorer Options to View SAS Datasets With Variable Names Rather than Labels.
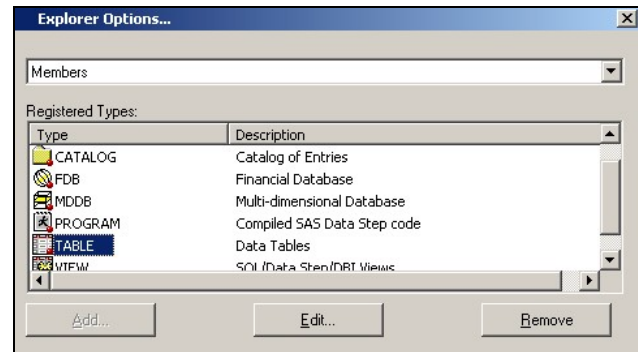
Figure 9. Changing Viewtable Command (Continued from Figure 8).


```
VIEWTABLE %8b.'%s'.DATA colheading=name
```

THE PROJECT FILE STRUCTURE

The directory structure common to all projects is shown in Figure 6, point F. Note that at a higher level, the project has a structure organized according to the business. As an example, subdirectories are organized by species (cattle, swine, etc) and by trial within species (Trial A, Trial B, etc).

Within each project, there are locations for a wide variety of documents – correspondence, Excel spreadsheets, SAS code, SAS datasets, Word documents, graphics metafiles, etc. Note that within any folder in the SAS Explorer, you should be able to double click on a file (e.g. a "ppt" file in the Powerpoint folder) and launch the associated application. If you cannot do this, the file associations from within SAS may not be communicating with the Windows file associations and you need to contact SAS technical support.

SAS code for the project is stored in the "SAS Programs" folder. This should be the same folder listed at the bottom of the SAS Application Work Space. All file "opens" and "saves" will be directed to this folder.   Note the sequential naming conventions used for the files in this directory. File "10_get_data_proc_import.sas" contains the code to retrieve data from a database table via proc import. Other files are sequentially numbered in a logical order to reach the conclusion of the programming effort. These files are edited and submitted individually. The lead program "00runall.sas" contains a "%include filename" statement for each file in the project and can be used to submit all of the code for the project when it is completed. This file numbering and submittal process greatly reduces the amount of code that needs to be repeatedly resubmitted and repeatedly rerun when code is being developed. Frequently programmers place all code in one file and put up with the frustration of waiting for reprocessing, or spend considerable time commenting out or interactively selecting code to either run or not run.

ACCESSING DATA FROM DATABASES

Two methods of accessing data from common Windows databases such as Microsoft Access are easily used:

1. Use of **the File Import** capability when a SAS editor view is open. Simply use this to interactively access your data and bring it in. The code that is generated by this technique can be stored for future use. An example follows:

```
PROC IMPORT OUT= WORK.weight
            DATATABLE= "Weight"
            DBMS=ACCESS2000 REPLACE;
     DATABASE="D:\SUGI
Examples\Cattle\Trial
A\Database\Trial_A.mdb";
RUN;
```

2. Use of the **Open Database Connectivity (ODBC)** utility in Microsoft Windows. This facility serves as an interface for a wide variety of database and software packages. Using this is a little more complicated than "file import" in that an ODBC file reference must be established on the machine on which SAS is running to provide the directions for finding the data. The icon to initiate this is found in various locations in the different flavors of the Windows operating systems. In Windows 2000 it is located by following this sequence from Start Programs in the lower left: – Start – Settings – Control Panel – Administrative Tools – Data Sources (ODBC). The window shown in Figure 10 will appear.   Your objective is to add your data source to this panel such as has been done for "Cattle Trial A". The example of completing the succeeding screens is shown in Figure 11.

Figure 10. ODBC Adminstrator Screen Managing Defined Data Sources.
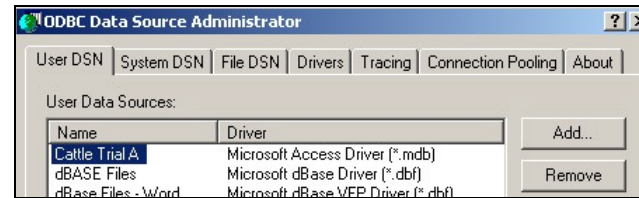
Figure 11. Defining the ODBC Data Source.
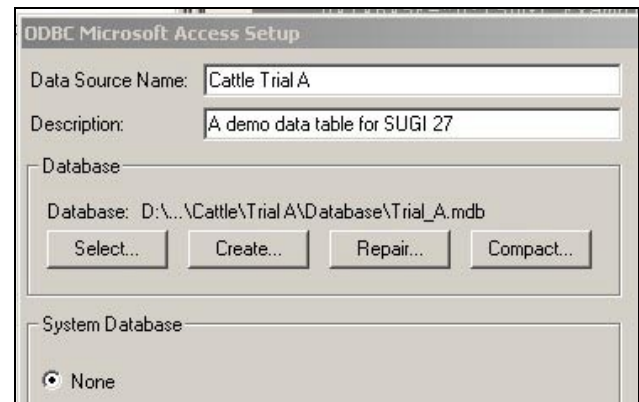
3. To use this ODBC definition in SAS and build a SAS dataset, use the following:

```
Proc SQL;
  connect to odbc (dsn="Cattle Trial
A");
  create table work.weight as
    select * from connection to odbc
       (select * from Weight);
  quit;
```

ROUTING TEXT OUTPUT TO FILES FOR INSERTION IN MICROSOFT WORD

Reports written by SAS typically are text files containing tabular data. Frequently these documents become part of the body or attachments to word processor files.   To enable further processing in the Microsoft Word environment, key words are dropped in the SAS output file to facilitate post-processing in Word via VBA macros. These actions might be used to control appearance and indexing functions. As an example, the key words "Header5" inserted at the beginning of titles in the following SAS code will be searched for by Word VBA macros that will result in selecting the line, assigning it to Header5 in the Word document and then deleting the word "Header5" in the title.   To create output that is easily incorporated into Microsoft Word, two techniques are used:

1. **ODS.** SAS Version 8 has the ODS facility that permits output to be routed to RTF (rich text format) files. An example of the SAS code required to do this follows. This output may suffice for some applications. I have found that some of the formatting exceeds that which is

appropriate for my work (titles become Word page headers, etc.). Note that style=styles.minimal and bodytitle reduce these problems. Note also that templates can be customized for the RTF file (see references).

```
ods rtf file='d:\SUGI
Examples\Cattle\Trial A\Reports TXT
Files\report7.rtf'
style=styles.minimal bodytitle;
*style=styles.default is more
graphical;
*bodytitle prevents titles in SAS from
becoming Word headers;
proc print data=final;title
'Header5Cattle Trial A, Listing of
Weight Table';
title2 'RTF File'; run;
proc sort data=final;by trial_no
treatment;run;
proc means data=final;by trial_no
treatment;
var gain  gain_per_day;
title 'Header5Cattle Trial A, Means by
Treatment';title2 'RTF File';run;
ods rtf close;
run;
```

2. **Proc Printto and Txt Files.** An old favorite, proc printto can be very useful in routing output to simple text files that can be inserted into Word documents. You have a lot of control over page layout, appearance, etc. SAS code to show the routing of the output to an external file and the insertion of keywords that will eventually be used to post-process the file follows:

```
options   FORMCHAR= "|----|+|---+=|-
/\<>*" ; **this is the second formchar
in the default configuration file;
*it greatly improves and simplifies
text output;
proc printto print='d:\SUGI
Examples\Cattle\Trial A\Reports TXT
Files\report.txt';
proc print data=final;title
'Header5Cattle Trial A, Listing of
Weight Table';run;
proc sort data=final;by trial_no
treatment;run;
proc means data=final;by trial_no
treatment;
var gain  gain_per_day;
title 'Header5Cattle Trial A, Means by
Treatment';
run;
proc printto; run;
```

ROUTING GRAPHICS OUTPUT TO FILES FOR INSERTION IN MICROSOFT WORD, POWERPOINT, ETC.

A very significant but little known graphics driver was added to SAS Version 8 which greatly improves the appearance of SAS Graph output placed in Microsoft Office documents. The EMF driver (enhanced metafile) produces files that are significantly improved over the WMF files (windows metafiles) which have been available for a long time. Not only can the EMF files be inserted into Office documents, they can be ungrouped and edited in Powerpoint.

An example of code which builds an emf file that will later be used in Word follows:

```
filename grafout "D:\SUGI
Examples\Cattle\Trial A\Graphics" ;
goptions device=emf gsfname=grafout
gsfmode=replace;

goptions ftext="Arial" htext=2
htitle=2.5 hsize=4 vsize=3.5;
goptions  hpos=50 vpos=40 transparency;

proc gchart data=final;
vbar treatment/sumvar=gain_per_day
type=mean discrete name="trt_mean"
outside=mean;
label Treatment="Test Drug (ppm)";
title .j=C  "Cattle Trial A, Means for
Gain Per Day";
title2 .j=C "Graph Developed Using SAS
EMF Driver";
format gain_per_day 5.2;
run;
```

INSERTING TEXT AND GRAPHICS FILES IN MICROSOFT WORD, USE OF VBA MACROS TO FORMAT

An example of a SAS-generated EMF graphics file inserted in a Word document is presented in Figure 12. Note that the graph is using sharp crisp Arial fonts and does not have the jagged appearance associated with alternative bitmap formats. The graph is scalable in size and can be edited in Powerpoint.

Figure 12. An Example of a SAS Graph EMF File Inserted in a Word Document.

Cattle Trial A, Means for Gain Per Day
Graph Developed Using SAS EMF Driver

gain_per_day MEAN

3.00 — 2.56   2.42

2.00 —

1.00 —

0.00 —

00   10

Test Drug (ppm)
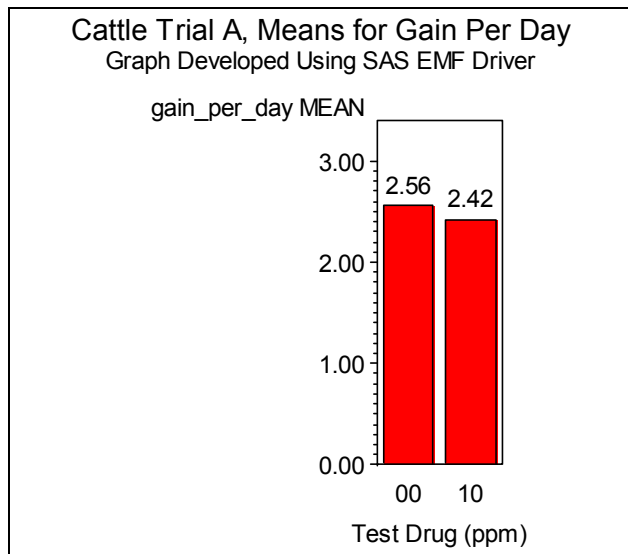
Inserting either a text or an RTF file in a Word document is accomplished by opening the Word document (preferably with a customized template for your institute/project), positioning the cursor at the appropriate insertion point, and then, from the Word menu bar select "Insert File" and find the txt or RTF file to insert. Following this, you may wish to run a Word VBA macro to perform formatting tasks (Tools Macro and make the selection). Once you have the formatting changes made in Word, it is very easy to generate a document table of contents by indexing the document to the degree of depth you desire via the specification of the headers (from the Word menu bar – Insert – Index and Tables – Table of Contents – Show Levels).

VBA MACROS
Microsoft furnishes VBA with all Office software. Implementation on a site-by-site basis will vary with local administrators having the power to enable/disable its use. An example of the VBA macro code that will post-process either of the files produced above (TXT or RTF) when placed in Word follows. The code searches for the term "Header5" in a line. If the term is found, the line is selected and set to header5 formatting as defined in the Word template. The term "Header5" is then stripped out of the line. The following VBA macro code was developed with Office XP (2002) with version 6.3 of VBA. Word macros are easy to understand once the user is familiar with the use of document templates, has mastered the concept of where macros are stored (templates), how they are edited (in the VBA editor) and how they are run. Programming in VBA is a more difficult matter. It is suggested that the beginning user turn on the macro record capability of Word and then examine the resultant macro code generated to get started.

```
Sub FixHeader5()
'  VB Macro to set header 5 and strip code
 With Selection
      .Find.Forward = True
      .Find.ClearFormatting
      .Find.Wrap = wdFindContinue
      .Find.MatchCase = False
      .Find.Text = "Header5"
     Do While (.Find.Execute)
         .MoveRight Unit:=wdCharacter, Count:=1,
Extend:=wdExtend
          .EndKey Unit:=wdLine, Extend:=wdExtend
```

```
         .Style = ActiveDocument.Styles("Heading 5")
         .Move Unit:=wdParagraph, Count:=1
    Loop
 End With

 'now replace text
 With Selection
     .Find.Forward = True
     .Find.ClearFormatting
     .Find.Wrap = wdFindContinue
     .Find.MatchCase = False
     .Find.Text = "Header5"
    Do While (.Find.Execute)
      .Find.Replacement.Text = ""
     .Find.Execute Replace:=wdReplaceAll, Forward:=True, _
       Wrap:=wdFindContinue
       .Move Unit:=wdParagraph, Count:=1
    Loop
 End With
End Sub
```

KEYS AND HELP

Function key assignments can be reviewed by hitting F9. When frequent submissions of SAS code are made, function key assignments to clear the SAS log and listing files are extremely handy. These are noted as point A in Figure 14. To assign these, simply type the commands into the screen that comes up when F9 is hit. Other extremely useful keys are marked B in Figure 14. Their functions and usage are adequately explained in numerous other SAS manuals and publications.

Help can be obtained in several ways: (a) use of the F1 key, (b) use of the Help menu pull down on the menu bar, (c) typing commands in the command window. Syntax help is frequently needed when coding SAS procedures. To get help on procedure syntax, simply type the procedure name after the word help in the command window. Do not type the word "proc" or "procedure". An example for obtaining syntax help on proc means is shown in Figure 13. Make sure to hit the return key after the command "help means" has been typed.

Figure 13. Obtaining Help for Proc Means from Command Window (Hit Return).

e Edit View Tools Solutions Window Help

✓ help means
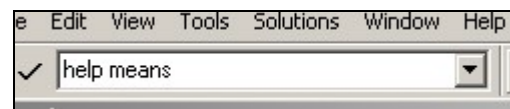
A NOTE ABOUT EDITORS
The Enhanced SAS editor is the preferred editor for editing SAS code within the SAS environment. To use the enhanced editor click on View – Enhanced Editor while having an edit window selected. If you wish to use the simpler editor – select View – Program Editor. While the enhanced editor is not nearly as sophisticated as some other commercial editors, there are a number of options that can be enabled. To show line numbers, select Tools – Options – Enhanced Editor and click the box for show line numbers. To issue line commands, select a line and then go to the command window and issue commands preceded by a colon. As an example, to repeat a line 3 times, select the line and then type ":r3" in the command window. This does not work on multiple line selections (i.e. does not have the power of block commands). Drag and drop of selected text is also enabled to move text.

Figure 14. Function Keys (A) Suggested Assignments and (B) Most Useful Defaults .

CONCLUSION

Adjustments to default SAS settings and the use of various techniques described in this paper lead to a much more satisfactory user experience when using SAS in the Windows environment. The techniques organize projects, provide for efficient iteration of code during the code development cycle and result in high quality documents published in the Microsoft Office environment.

REFERENCES

Hull, B., 2001. "Now There Is an Easy Way to Get to Word, Just Use PROC TEMPLATE, PROC REPORT, and ODS RTF", *Proceedings of the 26tth SAS User's Group*, paper 163.

McNeill, S., 2001. "Changes & Enhancements for ODS by Example (through Version 8.2)", *Proceedings of the 26tth SAS User's Group*, paper 2.

SAS Institute Inc*., SAS Companion for the Microsoft Windows Environment*, Version 8, Cary, NC, SAS Institute Inc., 1999. pp 555.

SAS Institute Inc., "The Template FAQ, Introduction to the RTF destination". Available at http://www.sas.com/rnd/base/topics/templateFAQ/Template_rtf.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Roger D. Muller, Ph.D.
Elanco Animal Health
500 East 96th St, Parkwood 4, Suite 125
Indianapolis, IN 46240
Work Phone: 317/276-3651
Fax: 317/276-6116
Email: rdmuller@lilly.com
Web: www.elanco.com