Hints & Tips on Using SAS®

As featured in the SAS UK newsletter 'In the Know - 2005'

The Power to Know.

Table of Contents

Introduction

This paper presents a collection of hints and tips that have featured in the SAS UK newsletter ‘ In the Know’ in 2005. The hints and tips included are a mix of customer contributions and reader suggestions together with SAS articles based on queries received by UK Customer Support.

For more free hints and tips delivered to your desktop on a bi-monthly basis, subscribe to ‘In the Know’ at:

http://www.sas.com/uk/newsletter/subscription.

Did You Know that You Can Now Access PC Files on a Windows Machine Directly from a Unix Machine?

The SAS/ACCESS® Interface to PC Files Software on UNIX has been significantly enhanced in SAS 9.1.

It is now possible to access a wider range of PC file formats from UNIX, provided that these files reside on a windows machine. This is made possible by the PC File Server, which resides on a network windows machine.

You can use the LIBNAME statement, the IMPORT and EXPORT procedures, and the Pass-Through Facility to access these new PC file formats.

When the PC files reside on UNIX, however, SAS/ACCESS for PC Files Software instead behaves as it did in SAS 8, working with a limited range of PC file formats via the Import/Export wizard and Import/Export procedures. In SAS 9.1, locally-residing JMP® files can also be accessed in this way.

To access Microsoft® Access database file (.mdb), Microsoft Excel workbook file (.xls) or any ODBC supported data sources, SAS/ACCESS for PC Files Software on Unix requires the PC File Server (pcfserver.exe) running on Windows but no local SAS installation is required.

The PC File Server (pcfserver.exe) component is shipped on the Client Side Components CD Volume1. The setup.exe can be found in a location similar to the following:

\client1cd\pcfilesrv\setup.exe

Unless otherwise specified, the PC File Server will be installed in C:\Program Files\SAS\PC File Server on your Windows machine. The setup.exe will install the following files:

- PCFServer.exe - the SAS PC File Server executable
- Several support files

You can start the PC File Server by going to Start -> Programs -> SAS -> PC File Server 9.1.

This will launch the SAS PC File Server. The Server Options section of the window contains connection information that can be configured if required. For further information, visit the SAS Online Doc CD for Configuring the PC Files Server.

Here is some example code directly accessing a local Excel spreadsheet with a PC File Server running and creating a temporary SAS dataset on a Unix machine from the Excel spreadsheet.

```
/* run the following code from your Unix machine */
libname localxls pcfiles type=Excel port=8621
server=machinename
path = “Path to local XLS file\Name_of_file.xls” ;
data work.New_Dataset_from_Excel ;
set localxls.”sheetname$”n;
run;
proc print data=work.New_Dataset_from_Excel;
run;
```

Did You Know that the Shortcut Key *Ctrl +]* Searches for a Matching Bracket in your Code?

(Suggested by Andy Cooper, Direct Line Group Limited)

There are a whole host of shortcut keys that can be used to make your life easier within SAS. In this article we look at some of the most useful shortcut keys that can be used within the Enhanced Editor window.

Category	Command	Keyboard Shortcut
Abbreviation	Bring up word tip	Alt + F1 + No Selection
	Hide the current word tip	Esc
Code Folding	Collapse all folding blocks	Alt + Ctrl + Number pad -
	Expand all folding blocks	Alt + Ctrl + Number pad +
Command/Macro Support	Execute the last recorded macro	Ctrl + F1
Edit	Copy selection	Ctrl + C
	Cut selection	Ctrl + X
	Paste from clipboard	Ctrl + V
Help	Get Help for a SAS procedure	place the cursor within a procedure name and press F1
	Context Help	F1
Navigation	Move cursor to matching brace/parentheses	Ctrl + [Ctrl +]
	Move cursor to matching DO/END keyword	Alt + [Alt +]
	Move cursor to next case change	Alt + Right
	Move cursor to previous case change	Alt + Left
Selection	Select all	Ctrl + A
Selection Operations	Clean up white space	Ctrl + Shift + W
	Comment the selection with line comments	Ctrl + /
	Undo the Comment	Ctrl + Shift + /
	Convert the selected text to lowercase	Ctrl + Shift + L
	Convert the selected text to uppercase	Ctrl + Shift + U

For a complete shortcut keys list, please search the SAS Online Doc CD for:

Default Key Definitions under Windows
Keyboard Shortcuts within Print Preview
Keyboard Shortcuts within the Enhanced Editor

Keyboard Shortcuts within the SAS Main Window

The tables above list the default key definitions for the primary SAS application windows (such as Program Editor, Log, and Output) and the Enhanced Editor window.

Any other key combinations that are not listed in the tables are either reserved by Windows or has a definition that you cannot change within SAS.

How Do I Capitalise the First Letter of a Word and Lower Case all the Other Letters in a Word?

(Written by Ifor Lewis, HFC Bank Ltd)

By using Macros and the SAS datastep functions UPCASE, LOWCASE and SUBSTR you can convert your text strings to proper case making your text more readable.

This mimics a function which is available in Oracle - INITCAP - to capitalise the first letter of a word and lower case all the other letters in a word. In my job this is useful, particularly for names and addresses when creating mailing files and sending data outside of our company.

It is designed to run within a dataset and I have included an example dataset and usage example:

```
%macro initcap(col_name);
     /* Copyright Iffy 2004 */
     do i=1 to length(&col_name) - 1;
          if substr(&col_name,i,1) in (' ','-',"'")
               then substr(&col_name,i+1,1) =
upcase(substr(&col_name,i+1,1 ));
          else substr(&col_name,i+1,1) = lowcase(substr(&col_
name,i+1,1));
     end;
     substr(&col_name,1,1) = upcase(substr(&col_name,1,1));
     drop i;
%mend;

/* Example dataset */
data temp;
     length firstname $5 fullname $25;
     infile datalines delimiter=',';
     input firstname $ fullname $;
     datalines;
dave,david johnson
john,joHn herrington-PROOPS
STEVE,steven barrington edwards
andY,andrew brown
jo,jo o'brien
;
run;

/* Usage - within a datastep */
data temp;
     set temp;
     %initcap(firstname);
     %initcap(fullname);
run;
```

In addition to the customer's suggestion, please find below alternative SAS datastep examples when using Base SAS 8.2. If you have the module SAS Data Quality Cleanse licensed in SAS 8.2, you will have access to the function PROPERCASES that automatically does this for you. Beginning with Base SAS®9 there is a new function PROPCASE which will convert all words in an argument to proper case.

http://ftp.sas.com/techsup/download/sample/datastep/char.html

http://www2.sas.com/proceedings/sugi25/25/cc/25p079.pdf

How Can I Change the Default Action When I Double Click on a SAS® File in Windows Explorer?

During the SAS System setup process, all files associated with SAS are registered in the Windows registration database. Actions for each of the file types are also registered. Right mouse clicking in the Windows Explorer on a file type will display all of the registered actions for a specific file type. When you double click on a file type, the default action for that file type is performed. The application invoked is determined by what is registered for that file type.

For SAS, the application invoked will be either SASOACT.EXE, The SAS System Viewer, or SAS.EXE itself (batch submit).

SASOACT.EXE will then see if an existing OLE automation session of SAS is running. If not, it will then invoke an OLE automation session of SAS. Next, the initial command:
“afa c=sashelp.dispatc.actions.frame” is sent to SAS.

Next, the dispatch window is made the active window and SASOACT.EXE passes the following command to the dispatch window: Action=”Open” Datatype=”SASFile” Filename=””. Finally, the Dispatch SCL program invokes the appropriate method to perform the end result. These methods are setup by The SAS Desktop contained with the SAS/BASE module.

To look at your settings, go to Windows Explorer, choose Tools ‘ Folder Options, then select the File Types tab. Scroll down to the SAS extension and click Advanced. Select the OPEN action and click Edit. You should see syntax similar to the following:

“C:\PROGRA~1\SASINS~1\SAS\V8\CORE\SASEXE\SASOACT.EXE” action=Open datatype=SASFilefilename=”%1” progid=SAS.Application.8

The %1 will be replaced by the Windows operating system with the actual filename you have chosen. Below are some examples showing the syntax in different versions of SAS.

“C:\PROGRA~1\SAS\SAS9~1.1\CORE\SASEXE\SASOACT.EXE” action=Submit datatype=SASFile filename=”Test.sas”

“C:\PROGRA~1\SASINS~1\SAS\V8\CORE\SASEXE\SASOACT.EXE” action=Open datatype=Data filename=”Houses.sd2”

Below is a brief summary of the command line arguments that can be used:

SASOACT - will parse the following command line arguments.

ACTION - SAS File Types and Actions for valid actions for the registered data types.

DATATYPE - SAS File Types and Actions for valid data types.

FILENAME - Fully qualified filename, for which the action is to be performed on. In the Explorer environment, the operating system provides the fully qualified filename.

ENTRY - Two level entry name, when datatype=Catalog, this parameter can be used to perform actions on individual catalog members.

INITCMD - This is the first command that will be executed once SAS is invoked. The default initial command is “afa c=sashelp.dispatc.actions.frame”. If initcmd is omitted, the default initcmd is invoked.

WINDOWNAME - After the initial command is executed a window is made active to send the action/datatype/filename string to. The default windowname is dispatch. If windowname is omitted, the dispatch window will be made the active window.

OPTIONS - SASOACT only searches for three specific options. Otherwise, the remaining arguments in the options statement is passed on through unchanged to the active window in SAS. The following three OPTIONS have special significance.
-NEW - SASOACT will always invoke a new OLE automation session of SAS. Otherwise, it will first check to see if an existing OLE automation session of SAS is running. If it is, then all commands will be sent to that session. Else, a new session will be invoked.
-DNDPRN - Printer driver, a parameter needed to support PrintTo or drag and drop to non-default printers.
-DNDPORT - Printer port, a parameter needed to support PrintTo or drag and drop to non-default printers.

PROGID - This is populated from the Windows Registry under the Registry structure:
HKEY_CLASSES_ROOT ‘ SAS.Application (Default) = “SAS.Application”
The current version settings can be found in the following location and the command it invokes.
CurVer (Default) = SAS.Application.901 (for version 9)
command (Default) = C:\PROGRAM FILES\SAS\SAS 9.1\SAS.EXE -CONFIG C:\PROGRAM FILES\SAS\SAS 9.1\SASV9.CFG

How Do You Retrieve Data from Indexed Datasets for a Handful of Members Only?

(Written By Jim Jeffrey, BCA)

The following code determines how many records you wish to retrieve data for and then creates the same amount of macro variables, each one containing the key value that you wish to retrieve. These macro values are then queried in a where statement.

```
%macro get_custs;
proc sql noprint;
select count(*) into :num_custs from sample_file;
%let num_custs = &num_custs;
select cust_ref into :cust1 - :cust&num_custs from sample_
file;
quit;
data retrieved_customers;
set indexed_master_file;
where cust_ref in (%do i=1 %to &num_custs; "&&cust&i" %end;);
run;
%mend get_custs;
%get_custs;
```

How Can I Find a Time Format that Does Not Remove the Leading Zero e.g. Time5 (output printed as 09:36 instead of 9:36)?

(Suggested by Stephen Hoar, Lloyds TSB)

There is currently no time format that puts leading zeros automatically in time values. But there are a number of ways of achieving this, below are some examples.

Example 1:
Create your own custom format using the following syntax. Then format your data values using the user defined format.

```
proc format ;
picture military other = '%0H:%0M:%0S'  (datatype=time) ;
run ;

data test;
x= '9:36't ;
format x military8. ;
put x=;
run ;

x=09:36:00
```

Example 2:
You can also use the TODw.d format which does write leading zeros, but the original value must be a SAS datetime value. To convert your time values to datetime values use the DHMS function. In the DHMS function insert the value: 0 date, 0 hours, 0 minutes, and then the SAS time value as the number of seconds.

Then format the new variable with the TODw.d format and you will have the correct time, including the leading zeros.

```
data test;
x='9:36't;
y=dhms(0,0,0,x);
format y tod8.;
put y=;
run;

y=09:36:00
```

Have a look at the following Technical Support website for further examples of using Date & Time functions and formats:

http://support.sas.com/techsup/sample/functions.html

When Using DDE to Output Data into an Excel File, How Do I Get Character Variables containing Spaces to Output into One Cell?

(Requested by Mark Ray, Barclaycard)

SAS expects to see a TAB character placed between each variable that is communicated across the DDE link. Similarly, SAS places a TAB character between variables when data are transmitted across the link. When the NOTAB option is placed in a FILENAME statement that uses the DDE device-type keyword, SAS accepts character delimiters other than tabs between variables.

The NOTAB option also can be used to store full character strings, including embedded blanks, in a single spreadsheet cell. For example, if a link is established between SAS and the Excel application, and a SAS variable contains a character string with embedded blanks, each word of the character string is normally stored in a single cell. To store the entire string, including embedded blanks in a single cell, use the NOTAB option.

Here is some sample code, a tab (‘09’x) is added between each variable, as in the following PUT statement so that column1 contains ‘Mark Ray’ and column2 contains ‘Jim Goodnight’.

```
filename test dde 'excel|sheet1!r2c1:r2c2' notab;
data string;
   file test;
   a='Mark Ray';
   b='Jim Goodnight';
   put a $15. '09'x b $15.;
run;
```

For additional Base SAS sample code, please have a look at the following website: http://ftp.sas.com/techsup/download/sample/samp_lib/hostsampBase_SAS_Sample_Programs.html

How Do I Use the SAS® Explorer Window to Create My Own Pop-up Menu Actions?

The SAS Explorer Options dialog box enables you to change the actions available for any data type, as well as customise the contents of the Explorer dialog box. You can use the window to edit or hide members, catalog entries, host files, or metadata servers. You can also add or remove host file types.

An example of this would be to double click on a Format catalog in the SAS explorer window and get a description of the format displayed in the Output window.

You will find that the SAS Explorer Options interface will look slightly different between SAS 8 and SAS®9 although the functionality is very similar.

Select the SAS Explorer window then from the pull down menu select Tools ‘Options ‘Explorer.

In SAS®9 select the ‘Entries’ tab, for SAS 8 select ‘Catalog Entries’ from the drop down list.

Scroll down to the catalog entry type of FORMAT and select EDIT. In the Format options window select ADD.

Type the following syntax in the boxes provided.

Action: Display FORMAT in the output window
Action Command: GSUBMIT “PROC FORMAT library=%8b page; /*%32b*/ select %32b; RUN; QUIT;”

For character FORMATS use the following syntax:

Action: Display character FORMAT in the output window
Action Command: GSUBMIT “PROC FORMAT library=%8b page; /*%32b*/ select $%32b; RUN; QUIT;”

Note: Character formats require a $ sign.
Informats require a @ sign.
%8b = valid SAS library name (8-byte SAS name allowed)
%32b = valid SAS member name (32-byte SAS name allowed)

Important: As in the example above the ‘%..’ substitutions are separated by SAS code so be careful to substitute the parameters in the correct order. Enter the %8b string for the library name, the commented %32b string to keep the place value for the catalog name and the second %32b for the catalog entry.

For host files, enter the following string for filename substitution into the command: ‘%s’ = quoted string.

E.g. GSUBMIT ‘filename temp ‘%s’;

For library members, enter the following string for library name, member name, and entry type substitution into the command, respectively:

E.g. KEYS %8b.%32b.%32b

For catalog entries types, enter the following string for library name, catalog name, catalog entry name, and entry type substitution into the command, respectively:

E.g. BUILD %8b.%32b.%32b.%8b

What are the Most Efficient Ways of Loading a SAS Dataset into an Oracle Table and Vice Versa?

You can read or write to an Oracle table from SAS on Windows using any of the following SAS modules:

SAS/ACCESS Interface to Oracle, accesses the database via the Oracle Client software.

SAS/ACCESS Interface to ODBC, accesses the database via the Oracle ODBC datasource.

SAS/ACCESS Interface to OLEDB, accesses the database via the Oracle OLEDB provider.

Each module supports reading and writing to the Oracle database from SAS using either of the following methods:

1) SAS/ACCESS Libname Engine statement.
2) Proc SQL Pass Through Facility.

For information on efficiency and example code, please have a look at the following SAS Technical documents:

Using the SAS/ACCESS Libname Technology to Get Improvements in Performance and Optimizations in SAS/SQL Queries:
http://support.sas.com/techsup/technote/ts661.pdf

Accessing an Oracle Database from SAS on Microsoft Windows:
http://support.sas.com/techsup/technote/ts703.html

SAS/ACCESS Guidelines for Connecting to Relational Database Systems in the UNIX Environment:
http://support.sas.com/techsup/technote/ts518.html

SAS/ACCESS Guidelines for Connecting to ORACLE Databases in the UNIX Environment:
http://support.sas.com/techsup/technote/ts518d.html

Processing ORACLE Dates with SAS/ACCESS in the UNIX Environment:
http://support.sas.com/techsup/technote/ts566d.html

How Do I Access the Contents of an Enterprise Guide 3.0 Project Outside of Enterprise Guide?

(Suggested by Robert Meekings, Legal Services Commission)

During a recent upgrade we were unable to access projects created with Enterprise Guide 3. Whilst waiting for the software to be re-installed, we needed access to these files and discovered that the following process would accomplish this.

Step 1: Create a copy of your Enterprise Guide project.

Step 2: Rename the copy of the project with a .zip extension, e.g. Test.zip

Step 3: Double-click to open Test.zip with your favorite zip archive software.

You should now be able to access and see the contents of your Enterprise Guide project as folders in the directory specified.

SAS UK
Wittington House, Henley Road,
Medmenham, Marlow
Bucks SL7 2EB
Tel: (44) 1628 486933
Fax: (44) 1628 483203
www.sas.com/uk

SAS International
PO Box 10 53 40
Neuenheimer Landstr. 28-30
D-69043 Heidelberg, Germany
Tel: (49) 6221 4160
Fax: (49) 6221 474850

SAS World Headquarters
SAS Campus Drive
Cary, NC 27513, USA
Tel: (1) 919 677 8000
Fax: (1) 919 677 4444
www.sas.com

0557UK0905