



SAS® 9.1 What's new for Programmers ?

John Raddon

Education Business Manager

SAS Education UK & Ireland

INTRODUCTION

With the advent of SAS® 9.1 there is a whole new world of functionality available to SAS programmers. This paper provides an overview of some of the new features in SAS® 9.1.

SAS® 9.1 represents a revolutionary change in terms of architecture and interaction with the SAS system when compared to previous SAS versions. The limited scope of this paper means that this structural difference cannot be dealt with here. There are however some Base SAS Language and Procedure enhancements, which will quickly find their way into the offices of SAS programmers around the world. Samples of those new and enhanced System Options, Functions and Procedures will be covered in this paper.

In addition to explaining about the new functionality the compatibility between previous versions of SAS and SAS® 9.1 will also be discussed.

WILL IT STILL WORK? OR COMPATIBILITY ISSUES IN THE SPOTLIGHT

SAS PROGRAMS

Your existing SAS version 8 programs will run in SAS® 9.1 with no changes.

GENERAL COMPATIBILITY

- Cross Environment Data Access (CEDA) supports access to files from different platforms, as well as between different SAS versions (transparent to the user)
- SAS® 9.1 may directly process Version 7 and 8 native files, by using the appropriate Libname Engine
- This applies also – with restrictions – to Version 6 files
- SAS 8 applications may also access SAS® 9.1 files

For some scenarios it might be necessary or preferable to use a more traditional migration strategy. To facilitate this approach the MIGRATE procedure, one of the new procedures in SAS® 9.1, can be used.

OVERVIEW OF THE MIGRATE PROCEDURE

One of the goals for SAS® 9.1 was to introduce a procedure that will ease the conversion of SAS data sets and catalogs. The MIGRATE procedure is the tool for transitioning from older releases to the current release.

It will migrate:

- A library directly to SAS® 9.1 beginning with Release 6.12
- A 32-bit library to a 64-bit library within the same operating system
- Data sets including the indexes, integrity constraints and audit trails

The Power to Know®



- Generation data sets
- Data sets retaining all attributes:
 - Compressed files that remain compressed
 - Encrypted files that remain encrypted
 - Passwords
- Data sets with deleted observations in place
- Views, catalogs, item stores and MDDBs
- The SAS internal date and time of when the member was created and last modified.

Note: Please note that the MIGRATE procedure should not be used to migrate across machine architectures as results cannot be guaranteed!

The MIGRATE procedure uses the CV2VIEW procedure to convert SAS/Access Views for Oracle, DB2 (OS/390) or Sybase.

Data step views that contain their source code are migrated to SAS® 9.1 and are automatically recompiled when first used. A message is written to the log to indicate data step views that do not contain the source code. Copy these views manually and recompile them.

SQL views are migrated without issue.

Catalogs should be migrated using the CPORT and CIMPORT procedures.

Item stores cannot be migrated between 32-bit and 64-bit platforms.

Stored compiled programs must be copied and recompiled.

General form of the MIGRATE procedure

```
PROC MIGRATE IN=libref OUT=libref <options>;  
RUN;
```

IN= option identifies the source library or the library to migrate from.

OUT= option identifies the target library or the library to migrate to.

Data set options are not allowed in the IN= or OUT= value.

The MIGRATE procedure automatically converts 32-bit members of the IN= library to 64-bit members in the OUT= library

SAS SYSTEM OPTIONS

SAS® 9.1 has a series of new system options, which help to control the environment and the behaviour of the software. The following are a selection of these options, which might be useful for the programmer.



DTRESET System Option

- DTRESET** specifies that SAS update the date and time in the titles of the SAS Log and the listing file
- NODTRESET** specifies that SAS should use the date and time you invoked the SAS system. This is the default

DSNFERR System Option

- DSNFERR** specifies that SAS issue an error message and stop processing if a reference is made to a SAS data set that does not exist. This is the default
- NODSNFERR** specifies that SAS ignore the error message and continue processing if a reference is made to a SAS data set that does not exist. The data set reference is treated as if `_NULL_` had been specified.

DATESTYLE System Option

The DATESTYLE option identifies the order of month, day and year. The default is LOCALE.

General form of the DATESTYLE option

option DATESTYLE = MDY MYD YMD YDM DMY DYM LOCALE
--

LOCALE System Option

The LOCALE option specifies that local conventions are used for such features as formatting dates. The new default LOCALE= setting is English.

Therefore the default DATESTYLE option is MDY.

VALIDFMTNAME System Option

Controls the length of informat and format names that can be used when creating new SAS data sets and format catalogs.

General form of the VALIDFMTNAME= option

OPTION VALIDFMTNAME=LONG FAIL WARN
--

- LONG** specifies that informat and format names can be up to 32 alphanumeric characters. Names must begin with A-Z or an underscore. LONG is the default.
- FAIL** writes an error message to the log if you attempt to use or create a format or informat name that is longer than eight characters.
- WARN** writes a warning message to the log to remind you that the format you used or created cannot be used with releases prior to SAS® 9.1.



SAS FUNCTIONS

In SAS® 9.1 there are a series of new functions available for character manipulation. Some examples are:

The CAT Function suite

The CAT function concatenates character strings without removing leading or trailing blanks. It fulfils the same purpose as the || symbol in previous releases.

The CATS function concatenates character strings and removes leading and trailing blanks.

The CATT function concatenates character strings and removes trailing blanks only.

General form of the CAT, CATS and CATT functions

`CAT<ST> (string-1, string-2, ... string-n)`

String specifies a character constant, variable or expression.

The default length of the new variable will be 200.

The CATX function concatenates character strings, removes leading and trailing blanks and inserts separators

General form of the CATX function

`CATX(separator, string-1, string-2, ... string-n)`

Separator specifies a character string that is used as a delimiter or separator between concatenated strings

String specifies a character constant, variable or expression.

The default length of the new variable will be 200.

The PROPCASE Function

The PROPCASE function returns a string in proper (mixed) case.

General form of the PROPCASE function

`PROPCASE(string, <delimiter(s)>)`

The default delimiters are blank, forward slash, hyphen, open parenthesis, period and tab.

For SAS users with knowledge of Perl, SAS® 9.1 has now also a series of functions allowing the use of Perl Regular Expressions to extract, manipulate or search for character patterns.



REVISED PROCEDURES

Parallel Processing

With SAS 9 comes the introduction of parallel processing. Parallel processing means that processing is handled by multiple CPUs simultaneously. This technology provides performance gains for two types of SAS processing:

- threaded I/O which aids applications that can process faster than the data can be delivered to the application. When an application cannot keep the available CPUs busy the application is said to be I/O-bound
- threaded application processing which aids applications that receive data faster than they can perform the necessary processing. These applications are referred to as CPU-bound.

Multi-Threaded Processing

Not all SAS processes are suitable candidates for parallel processing.

Processes suitable for threading are:

- Sorting
- Grouping
- Summarising

Threaded Procedures

- PROC REG
- PROC GLM
- PROC LOESS
- PROC DMREG
- PROC DMINE
- PROC SORT
- PROC SUMMARY/MEANS
- PROC SQL
- PROC REPORT
- PROC TABULATE

PROC FORMAT Enhancements

With the FORMAT procedure, it is now possible to create formats and informats with longer names.

Format names for character and numeric formats can have a maximum length of 32, which includes the \$ for character names.

Informat names for

- Character informats can have a maximum length of 30
- Numeric informats can have a maximum length of 31



New Statistics for the TABULATE Procedure

To be closely aligned with the MEANS procedure, the TABULATE procedure now supports the following statistics:

- upper confidence limits (UCLM)
- lower confidence limits (LCLM)
- skewness
- kurtosis

The ALPHA= option specifies the confidence level to compute the confidence limits for the mean. Alpha corresponds to 1 minus the confidence level.

```
ALPHA=value
```

Value between 0 and 1

The default is .05, which results in a 95% confidence interval

You must specify the statistic keyword LCLM or UCLM (or both) to compute confidence limits

New Options for the FREQ Procedure

CONTENTS= enables you to customise hyperlinks within HTML contents file.

```
TABLES variables / CONTENTS=string;
```

The CONTENTS= option

- is used in the TABLES statement
- is valid only with crosstabulation tables
- affects only the HTML contents file and not the HTML body file.

Note: The CONTENTS= option also exists in the REPORT and TABULATE procedures

ZEROS includes observations with zero weight values. The ZEROS option is used in the WEIGHT statement

```
WEIGHT variable / ZEROS;
```

Zero weight levels are included in

- chi-square goodness-of-fit test for one-way tables
- binomial computations for one-way tables
- computation of Kappa statistics for two-way tables
- frequency and crosstabulation tables.

By default, PROC FREQ does not process observations that have zero weights. Such observations do not contribute to the total frequency count, and any resulting zero-weight row or column causes many of the tests and measures of association to be undefined.



EXAMPLE

This Example shows one of the new options on the FREQ procedure

```
ods html file='d:\temp\freq.html'
      contents='d:\temp\freqc2.html'
      frame='d:\temp\freqf2.html';
proc freq data=orion.customer ;
  table country*gender /
    contents="View Country by Gender Report";
  table country*customer_type_id /
    contents="View Country by Customer Type
    Report";
run;
ods html close;
```

Screenshot of resulting html frameset.

The screenshot shows a web browser displaying the SAS Output Frame. The browser window is titled "SAS Output Frame - Microsoft Internet Explorer". The address bar shows the file path "D:\Temp\freqf2.html". The page content is titled "The SAS System" and "The FREQ Procedure". It displays a "Table of Country by Gender" with columns for Country, Gender, and Total. The table includes data for Andorra, United Arab Emirates, Austria, Australia, and Belgium. A "Table of Contents" is visible on the left side of the browser window.

Country(Customer Country)	Gender(Customer Gender)		Total
	Female	Male	
Andorra	9	11	20
	0.01	0.01	0.02
	45.00	55.00	
	0.02	0.02	
United Arab Emirates	0	20	20
	0.00	0.02	0.02
	0.00	100.00	
	0.00	0.04	
Austria	630	772	1402
	0.70	0.86	1.56
	44.94	55.06	
	1.56	1.55	
Australia	1800	2200	4000
	2.00	2.45	4.45
	45.00	55.00	
	4.47	4.43	
Belgium	720	882	1602
	0.80	0.98	1.78

NEW PROCEDURES

PROC MIGRATE already covered earlier is joined by another new procedure PROC DOCUMENT.

The PROC DOCUMENT Procedure

The DOCUMENT Procedure enables you to:

- transform a report without having to rerun the analysis or repeat the database query.
- have more control over the structure of your output.
- display your output to any ODS output format without executing your SAS programs again.
- navigate the current file location and list entries.

The Power to Know.



- open and list ODS documents.
- manage output.
- store the ODS output objects in raw form. The output is kept in the original internal representation as a data component plus a table definition.

FOR MORE INFORMATION

Interested programmers are encouraged to book on the *SAS 9 Changes and Enhancements* seminar, which covers the topics mentioned in this paper in more detail. It also provides additional information on the more technical aspects of the SAS® 9.1 Business Intelligence Architecture.

CONCLUSION

For many companies SAS® 9.1 is a major step towards deploying the right information to the right people at the right time. This can be achieved through the Business Intelligence Architecture, which forms the basis of the enterprise wide use of SAS® 9.1. But in addition to the step change this advance in SAS technology allows, there are many gradual changes, enhancements and benefits which can be implemented and exploited immediately after moving to SAS® 9.1.

This paper shows how the new Base SAS Language and Procedure enhancements, which make use of the new capabilities of SAS® 9.1, can be used by SAS programmers to increase their productivity as well as the efficiency of their programs.



REFERENCES

Parts of this paper were taken from *SAS 9 Changes and Enhancements* seminar by Pete Snelling, Guenter Grimm and John Spooner, SAS UK Education Department. This Seminar can be booked through the UK Education Website (<http://www.sas.com/uk/education>).

CONTACT INFORMATION

John Raddon
Education Business Manager
Wittington House
Henley Road
Medmenham
Marlow
Buckinghamshire
SL7 2EB
Telephone: +44 (0) 1628 404 242
Mobile: +44 (0) 7979 366 145
Facsimile: +44 (0) 1628 404 448
Email: John.Raddon@suk.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright ©2003, SAS Institute Inc. All rights reserved.