What's New

Overview

New and enhanced features in Base SAS save you time, effort, and system resources by providing faster processing and easier data access and management, more robust analysis, and improved data presentation.

- By using new SAS system options that enable threading and the use of multiple CPUs, the following SAS procedures take advantage of multi-processing I/O: SORT, SQL, MEANS, TABULATE, and REPORT.
- The LIBNAME statement now supports secure access to SAS libraries on a WebDAV server.
- You can now use longer, easier-to-read names for user-created formats and informats. See "Rules for Words and Names in the SAS Language" in *SAS Language Reference: Concepts*.
- Two pre-defined component objects for the DATA step enable you to quickly store, search, and retrieve data based on lookup keys.
- The FILENAME statement now supports directory services, multiple FTP service commands, and Secure Sockets Layering (SSL).
- The Application Response Measurement (ARM) system enables you to monitor the availability and performance of transactions within and across diverse applications.
- The Perl regular expression (PRX) functions and CALL routines use a modified version of Perl as a pattern-matching language to enhance search-and-replace operations on text.
- New character functions search and compare character strings in addition to concatenating character strings.
- There are several new descriptive statistic functions and mathematical functions.
- New formats, informats, and functions support international and local values for money, datetime, and Unicode values. All data set options, formats, informats, functions, and system options that relate to national language support are documented in the new *SAS National Language Support (NLS): User's Guide*.
- A new ODS statement enables you to render multiple ODS output formats without re-running a PROC or a DATA step. See the *SAS Output Delivery System: User's Guide*.

Note:

- This section describes the features of Base SAS that are new or enhanced since SAS 8.2.
- z/OS is the successor to the OS/390 operating system. SAS 9.1 (and later) is supported on both OS/390 and z/OS operating systems and, throughout this document, any reference to z/OS also applies to OS/390, unless otherwise stated.

△

SAS System Features

Application Response Measurement (ARM)

Application Response Measurement (ARM) enables you to monitor the availability and performance of transactions within and across diverse applications. The SAS ARM interface consists of the implementation of the ARM API as ARM macros and an ARM agent. An ARM agent generates calls to the ARM macros. New ARM system options enable you to manage the ARM environment and to log internal SAS processing transactions. See "Monitoring Performance Using Application Response Measurement (ARM)" in *SAS Language Reference: Concepts*, "ARM Macros" on page xx, and "System Options" on page xxi.

Cross-Environment Data Access (CEDA)

CEDA processes SAS files that were created on a different host. This is especially useful if you have upgraded from a 32–bit platform to a 64–bit platform. Messages in the SAS log notify you when CEDA is being used to process a SAS file. See "Processing Data Using Cross-Environment Data Access (CEDA)" in *SAS Language Reference: Concepts*.

DATA Step Object Attributes and Methods

SAS now provides two pre-defined component objects for use in a DATA step: the hash object and the hash iterator object. These objects enable you to quickly and efficiently store, search, and retrieve data based on lookup keys.

The DATA step component object interface enables you to create and manipulate these component objects by using statements, attributes, and methods. You use the DATA step object dot notation to access the component object's attributes and methods.

The hash and hash iterator objects have one attribute, fourteen methods, and two statements associated with them. See Appendix 1, "DATA Step Object Attributes and Methods," on page 1765.

Engines

- The default BASE engine in SAS supports longer format and informat names, thread-enabled procedures such as the SORT and SUMMARY procedures, and more than 32,767 variables in a SAS data set.
- The metadata LIBNAME engine enables you to use metadata in order to access and augment data that is identified by the metadata. The metadata engine

retrieves information about the target SAS data library from metadata objects in a specified SAS Metadata Repository on the SAS Metadata Server. The metadata engine provides a consistent method for accessing many data sources. That is, SAS provides different engines that have different options, behavior, and tuning requirements. By taking advantage of metadata, the necessary information that is required to access data can be created in one central location so that applications can use the metadata engine to access different sources of data, without having to understand the differences and details of each SAS engine. See the *SAS Metadata LIBNAME Engine: User's Guide*.

- The XML LIBNAME engine imports and exports a broader variety of XML documents. The XMLMAP= option specifies a separate XML document that contains specific XMLMap syntax. The XMLMap syntax, Version 1.2, tells the XML engine how to interpret the XML markup in order to successfully import an XML document. See the *SAS Metadata LIBNAME Engine: User's Guide*.
- The new SASEDOC LIBNAME engine enables you to bind output objects that persist in an ODS document. See the *SAS Output Delivery System: User's Guide*.
- The new SAS Information Maps LIBNAME Engine provides a read-only way to access data that is generated from a SAS Information Map and to bring it into a SAS session. After you retrieve the data, you can run almost any SAS procedure against it. See the *Base SAS Guide to Information Maps*.
- The new character variable padding (CVP) engine expands character variable lengths, using a specified expansion amount, so that character data truncation does not occur when a file requires transcoding. Character data truncation can occur when the number of bytes for a character in one encoding is different from the number of bytes for the same character in another encoding, such as when a single-byte character set (SBCS) is transcoded to a double-byte character set (DBCS). See the *SAS National Language Support (NLS): User's Guide*.

Indexing

When creating an index that requires sorting, SAS tries to sort the data by using the thread-enabled sort. By dividing the sorting task into separately executable processes, the time that is required to sort the data can be reduced. See the topic "Creating an Index" in Understanding SAS Indexes in *SAS Language Reference: Concepts*.

Integrity Constraints

Variables in a SAS data file can now be part of both a primary key (general integrity constraint) and a foreign key (referential integrity constraint). However, there are restrictions when defining a primary key constraint and a foreign key constraint that use the same variables. See the topic "Overlapping Primary Key and Foreign Key Constraints" in Understanding Integrity Constraints in *SAS Language Reference: Concepts*.

Restricted System Options

System administrators can restrict system options from being modified by a user. You can use the RESTRICT option in the OPTIONS procedure to list the restricted options. The implementation of restricted options is specific to the operating environment. For details about how to restrict options, see the configuration guide for your operating environment. For information about listing restricted options, see the OPTIONS procedure in the *Base SAS Procedures Guide*.

SAS Utility Macro

The SAS utility macro, %DS2CSV, is available now in Base SAS. This macro converts SAS data sets to comma-separated values (CSV) files. Prior to SAS 9.1, this macro was available only for SAS/IntrNet users.

Universal Unique Identifiers

A Universal Unique Identifier (UUID) is a 128-bit identifier that consists of date and time information, and the IEEE node address of a host. UUIDs are useful when objects such as rows or other components of a SAS application must be uniquely identified. For more information, see "Universal Unique Identifiers" in *SAS Language Reference: Concepts*.

SAS Language Elements

Descriptions of the new and enhanced language elements for national language support can be found in "What's New for SAS 9.0 and 9.1 National Language Support" in the *SAS National Language Support (NLS): User's Guide*.

Data Set Options

- The following data set options are new:

  OBSBUF=
  determines the size of the view buffer for processing a DATA step view.

  SPILL=
  specifies whether to create a spill file for non-sequential processing of a DATA step view.

- The following data set options are enhanced:

  BUFNO=
  supports the same syntax as the BUFNO= system option in order to specify the number of buffers to be allocated for processing a SAS data set.

  BUFSIZE=
  supports the same syntax as the BUFSIZE= system option in order to specify the permanent buffer page size for an output SAS data set.

  FIRSTOBS=
  supports the same syntax as the FIRSTOBS= system option in order to specify which observation SAS processes first.

  OBS=
  supports the same syntax as the OBS= system option in order to specify when to stop processing observations.

Formats

- The maximum length for character format names is increased to 31. The maximum length for numeric format names is increased to 32.

- Several formats have been enhanced with default and range values.
- The following formats are new:

  MMYY
  writes date values in the form *mm*M<*yy*>*yy*, where M is the separator and the year is written in either 2 or 4 digits.

  PERCENTN
  produces percentages, using a minus sign for negative values.

  YYMM
  writes date values in the form <*yy*>*yy*M*mm*, where the year is written in either 2 or 4 digits and M is the separator.

  YYQ
  writes date values in the form <*yy*>*yy*Q*q*, where the year is written in either 2 or 4 digits, Q is the separator, and *q* is the quarter of the year.

  YYQR
  writes date values in the form <*yy*>*yy*Q*qr*, where the year is written in either 2 or 4 digits, Q is the separator, and *qr* is the quarter of the year expressed in Roman numerals.
- The PVALUE format now returns missing values that are specified by the MISSING= system option.

Functions and CALL Routines

New functions and CALL routines include character, mathematical, descriptive statistical, and special functions, and character-string matching functions that can use PERL expressions.

- The following character functions are new:

  ANYALNUM
  searches a character string for an alphanumeric character and returns the first position at which it is found.

  ANYALPHA
  searches a character string for an alphabetic character and returns the first position at which it is found.

  ANYCNTRL
  searches a character string for a control character and returns the first position at which it is found.

  ANYDIGIT
  searches a character string for a digit and returns the first position at which it is found.

  ANYFIRST
  searches a character string for a character that is valid as the first character in a SAS variable name under VALIDVARNAME=V7, and returns the first position at which that character is found.

  ANYGRAPH
  searches a character string for a graphical character and returns the first position at which it is found.

  ANYLOWER
  searches a character string for a lowercase letter and returns the first position at which it is found.

ANYNAME
searches a character string for a character that is valid in a SAS variable name under VALIDVARNAME=V7, and returns the first position at which that character is found.

ANYPRINT
searches a character string for a printable character and returns the first position at which it is found.

ANYPUNCT
searches a character string for a punctuation character and returns the first position at which it is found.

ANYSPACE
searches a character string for a white-space character (blank, horizontal tab, vertical tab, carriage return, line feed, or form feed), and returns the first position at which it is found.

ANYUPPER
searches a character string for an uppercase letter and returns the first position at which it is found.

ANYXDIGIT
searches a character string for a hexadecimal character that represents a digit and returns the first position at which that character is found.

CAT
concatenates character strings without removing leading or trailing blanks.

CATS
concatenates character strings and removes leading and trailing blanks.

CATT
concatenates character strings and removes trailing blanks only.

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

CHOOSEC
returns a character value that represents the results of choosing from a list of arguments.

CHOOSEN
returns a numeric value that represents the results of choosing from a list of arguments.

COMPARE
returns the position of the left-most character by which two strings differ, or returns 0 if there is no difference.

COMPGED
compares two strings by computing the generalized edit distance.

COMPLEV
compares two strings by computing the Levenshtein edit distance.

COUNT
counts the number of times that a specific substring of characters appears within a character string that you specify.

COUNTC
: counts the number of specific characters that either appear or do not appear within a character string that you specify.

FIND
: searches for a specific substring of characters within a character string that you specify.

FINDC
: searches for specific characters that either appear or do not appear within a character string that you specify.

IFC
: returns a character value that matches an expression.

IFN
: returns a numeric value that matches an expression.

LENGTHC
: returns the length of a character string, including trailing blanks.

LENGTHM
: returns the amount of memory (in bytes) that is allocated for a character string.

LENGTHN
: returns the length of a non-blank character string, excluding trailing blanks, and returns 0 for a blank character string.

NLITERAL
: converts a character string that you specify to a SAS name literal (n-literal).

NOTALNUM
: searches a character string for a non-alphanumeric character and returns the first position at which it is found.

NOTALPHA
: searches a character string for a non-alphabetic character and returns the first position at which it is found.

NOTCNTRL
: searches a character string for a character that is not a control character and returns the first position at which it is found.

NOTDIGIT
: searches a character string for any character that is not a digit and returns the first position at which that character is found.

NOTFIRST
: searches a character string for an invalid first character in a SAS variable name under VALIDVARNAME=V7, and returns the first position at which that character is found.

NOTGRAPH
: searches a character string for a non-graphical character and returns the first position at which it is found.

NOTLOWER
: searches a character string for a character that is not a lowercase letter and returns the first position at which that character is found.

NOTNAME
searches a character string for an invalid character in a SAS variable name under VALIDVARNAME=V7, and returns the first position at which that character is found.

NOTPRINT
searches a character string for a non-printable character and returns the first position at which it is found.

NOTPUNCT
searches a character string for a character that is not a punctuation character and returns the first position at which it is found.

NOTSPACE
searches a character string for a character that is not a white-space character (blank, horizontal tab, vertical tab, carriage return, line feed, or form feed), and returns the first position at which it is found.

NOTUPPER
searches a character string for a character that is not an uppercase letter and returns the first position at which that character is found.

NOTXDIGIT
searches a character string for a character that is not a hexadecimal digit and returns the first position at which that character is found.

NVALID
checks a character string for validity for use as a SAS variable name in a SAS statement.

PROPCASE
converts all words in an argument to proper case.

PRXCHANGE
performs a pattern-matching replacement.

PRXPOSN
returns the value for a capture buffer.

SCANQ
returns the n^{th} word from a character expression and ignores delimiters that are enclosed in quotation marks.

STRIP
returns a character string with all leading and trailing blanks removed.

SUBPAD
returns a substring that has a length you specify, using blank padding if necessary.

SUBSTRN
returns a substring that allows a result with a length of 0.

- The following descriptive statistics functions are new:

  GEOMEAN
  returns the geometric mean.

  GEOMEANZ
  returns the geometric mean without fuzzing the values of the arguments that are approximately 0.

  HARMEAN
  returns the harmonic mean.

  HARMEANZ
  returns the harmonic mean without fuzzing the values of the arguments that are approximately 0.

  IQR
  returns the interquartile range.

  LARGEST
  returns the k^{th} largest non-missing value.

  MAD
  returns the median absolute deviation from the median.

  MEDIAN
  computes median values.

  MODZ
  returns the remainder from the division of the first argument by the second argument; uses 0 fuzzing.

  PCTL
  computes percentiles.

  RMS
  returns the root mean square.

  SMALLEST
  returns the k^{th} smallest non-missing value.

- The following External Files function is new:

  DCREATE
  creates an external directory.

- The following macro functions are new:

  SYMEXIST
  indicates the existence of a macro variable.

  SYMGLOBL
  indicates whether a macro variable has global scope in the DATA step during DATA step execution.

  SYMLOCAL
  indicates whether a macro variable has local scope in the DATA step during DATA step execution.

- The following mathematical functions are new:

  BETA
  returns the value of the beta function.

  COALESCE
  returns the first non-missing value from a list of numeric arguments.

COALESCEC
returns the first non-missing value from a list of character arguments.

LOGBETA
returns the logarithm of the beta function.

- The following probability function is new:

LOGCDF
returns the logarithm of a left cumulative distribution function.

- The following quantile function is new:

QUANTILE
returns the quantile from the specified distribution.

- The following special function is new:

UUIDGEN
returns the short or the binary form of a Universal Unique Identifier (UUID).

- The following state and ZIP code function is new:

ZIPCITY
returns a city name and the two-character postal code that corresponds to a ZIP code.

- The following trigonometric function is new:

ATAN2
returns the arc tangent of two numeric variables.

- The following truncation functions are new:

CEILZ
returns the smallest integer that is greater than or equal to the argument; uses 0 fuzzing.

FLOORZ
returns the largest integer that is less than or equal to the argument; uses 0 fuzzing.

INTZ
returns the integer portion of the argument; uses 0 fuzzing.

ROUND
rounds the first argument to the nearest multiple of the second argument, or to the nearest integer when the second argument is omitted.

ROUNDE
rounds the first argument to the nearest multiple of the second argument, and returns an even multiple when the first argument is halfway between the two nearest multiples.

ROUNDZ
rounds the first argument to the nearest multiple of the second argument; uses 0 fuzzing.

- The following variable information functions are new:

  VVALUE
  returns the formatted value that is associated with the variable that you specify.

  VVALUEX
  returns the formatted value that is associated with the argument that you specify.

- Using Perl regular expression (PRX) functions and CALL routines is new. The following PRX functions are new. For more information, see "Pattern Matching Using SAS Regular Expressions (RX) and Perl Regular Expressions (PRX)" on page 276.

  PRXMATCH
  searches for a pattern match and returns the position at which the pattern is found.

  PRXPAREN
  returns the last bracket match for which there is a match in a pattern.

  PRXPARSE
  compiles a Perl regular expression that can be used for pattern-matching a character value.

  CALL PRXCHANGE
  performs a pattern-matching substitution.

  CALL PRXDEBUG
  enables Perl regular expressions in a DATA step to send debug output to the SAS log.

  CALL PRXFREE
  frees unneeded memory that was allocated for a Perl regular expression.

  CALL PRXNEXT
  returns the position and length of a substring that matches a pattern and iterates over multiple matches within one string.

  CALL PRXPOSN
  returns the start position and length for a capture buffer.

  CALL PRXSUBSTR
  returns the position and length of a substring that matches a pattern.

- The following CALL routines are new:

  CALL ALLPERM
  generates all permutations of the values of several variables.

  CALL CATS
  concatenates character strings and removes leading and trailing blanks.

  CALL CATT
  concatenates character strings and removes trailing blanks only.

  CALL CATX
  concatenates character strings, removes leading and trailing blanks, and inserts separators.

  CALL COMPCOST
  sets the costs of operations for later use by the COMPGED function.

CALL LOGISTIC
returns the logistic value of each argument.

CALL MISSING
assigns a missing value to specified character or numeric variables.

CALL RANPERK
randomly permutes the values of the arguments and returns a permutation of *k* out of *n* values.

CALL RANPERM
randomly permutes the values of the arguments.

CALL SCAN
returns the position and length of a given word in a character expression.

CALL SCANQ
returns the position and length of a given word in a character expression, and ignores delimiters that are enclosed in quotation marks.

CALL SOFTMAX
returns the softmax value for each argument.

CALL STDIZE
standardizes the values of one or more variables.

CALL STREAMINIT
specifies a seed value to use for subsequent random number generation by the RAND function.

CALL SYMPUTX
assigns a value to a macro variable and removes both leading and trailing blanks.

CALL TANH
returns the hyperbolic tangent of each argument.

CALL VNEXT
returns the name, type, and length of a variable that is used in a DATA step.