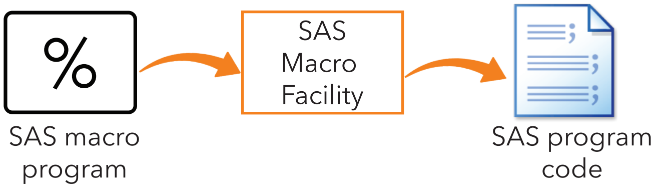
SAS® Macro Language Processing

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

SAS Macro Facility



Creating Macro Variables

Syntax	Description
%GLOBAL *macro-variable-1 ...macro-variable-n*;	Creates a macro variable that is available during the execution of the entire SAS session.
%LET *variable=value*;	Creates a macro variable and assigns it a value.
%LOCAL *macro-variable-1 ...macro-variable-n*;	Creates a macro variable that is available only during the execution of the macro where it is defined.

Defining a Macro

```
%MACRO macro-name <(parameter-list)>;
  macro-text
%MEND <macro-name>;
```

The *parameter-list* can be:
<positional-parameter-1, ...positional-parameter-n> or
<keyword-1=value-1, ...keyword-n=value-n>

Calling a Macro

```
%macro-name
%macro-name(positional-parameter-1,
               ...positional-parameter-n)
%macro-name(keyword-1=value-1, ....keyword-n=value-n)
```

Referencing a Macro Variable

Use the name of the macro variable with an ampersand.
&macro-variable;

Macro Character Functions

Syntax	Description
%INDEX (*source*, *string*)	Determines the position of the first character of a string within another string.
%SCAN (*argument*, *n*, *<charlist <,modifiers>>*)	Searches the argument and returns the *n*th word.
%SUBSTR(*argument*, *position*, *<,length>*)	Produces a substring of character string (*argument*) by extracting the specified number of characters (*length*) beginning at the specified starting position.
%UPCASE(*character-string*| *text-expression*)	Converts lowercase characters in the argument to uppercase.

SAS Functions with Macro Variables

Syntax	Description
%EVAL(*arithmetic* or *logical expression*)	Evaluates arithmetic and logical expressions using integer arithmetic.
%SYSEVALF(*expression <,conversion-type>*)	Evaluates arithmetic and logical expressions using floating-point arithmetic.
%SYSFUNC(*function (argument-1 <...argument-n>) <,format>*)	Executes SAS functions or user-written functions in the macro facility.

Troubleshooting Macro Variable References

Enables you to write your own messages to the SAS log.
%PUT *text*;

Deletes the specified variables from the macro global symbol table.
%SYMDEL *macro-variable-1 <...macro-variable-n></option>*;

SAS® Macro Language Processing

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

Masking Special Characters

Syntax	Description
%STR(*argument*)	Hides the usual meaning of a semicolon(;) so it appears as constant text.
%NRSTR (*character-string*)	Hides the usual meaning of an ampersand (&) or a percent sign (%) so they appear as constant text.
%SUPERQ (*argument*)	Masks all special characters and mnemonic operators at macro execution but prevents further resolution of the value.
%BQUOTE(*character-string* | *text-expression*)	Masks special characters and mnemonic operators in a resolved value at macro execution.
%QUPCASE(*character-string* | *text-expression*)	Converts values to uppercase and returns a result that masks special characters and mnemonic operators.
%QSUBSTR(*argument*, *position* <,*length*>)	Produces a substring of a character string.
%QSCAN(*argument*, *n* <,*charlist*<, *modifiers*>>)	Searches for a word and masks special characters and mnemonic operators.
%QSYSFUNC(*function*(*arguments*) <,*format*>)	Executes functions and masks special characters and mnemonic operators.

Conditional Processing

```
%IF expression %THEN text;
<%ELSE text>;

%IF expression %THEN %DO;
    text and/or macro language statements;
%END;
%ELSE %DO;
    text and/or macro language statements;
%END;

%DO index-variable=start %TO stop <%BY increment>;
    text
%END;
```

Options

```
OPTIONS MCOMPILENOTE= NONE | NOAUTOCALL | ALL;
OPTIONS MPRINT | NOMPRINT;
OPTIONS MLOGIC | NOMLOGIC;
OPTIONS MAUTOSOURCE | NOAUTOSOURCE;
```

Creating Macros in SQL

```
PROC SQL NOPRINT;
  SELECT column1<,column2,...>
    INTO :macro-variable-1<,:macro-variable-2,...>
            <TRIMMED>
    FROM table-1 | view-1
    <WHERE expression>
    <other clauses>;
QUIT;
```

DATA Step Interface

```
CALL SYMPUTX(macro-variable-name, value
                  <,symbol-table>);

PUT(source,format.);
```

Advanced Macro Techniques

```
%INCLUDE file-specification </SOURCE2>;
DOSUBL(text-string);
```

Default Autocall Library

%LOWCASE(*argument*)

%QLOWCASE(*argument*)

%LEFT(*argument*)

%TRIM(*argument*)

%CMPRES(*argument*)

%DATATYP(*argument*)