**SAS MACROS**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO VARIABLE BY LET** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%let variable =value;

options symbolgen mprint mlogic merror serror ;

**data** a;

input name $ age;

cards;

hh 34

jj 45

kk 67

;

**run**;

%let name=('hh');

**%macro** ***abc***;

data b;

set a ;

where name=&name;

run;

proc print data=b;

run;

**%mend** abc;

%***abc***;

%include 'E:\Office\SAS\_Chakra\_Training\SAS Macros\tes\external.sas' / source2;

%***abc***

**proc** **sort** data=c out=bydays;

by name;

**run**;

%let site=DALLAS;

**data** a;

input name $ age;

cards;

hh 34

jj 45

kk 67

;

**run**;

**proc** **print** data=a;

title "REVENUES FOR &site TRAINING CENTER";

**run**;

**data** aa;

input name $ begin\_date date9.;

cards;

hh 12mar2009

jj 14jan2018

kk 22mar2009

;

**run**;

**proc** **print** data=aa;

where begin\_date between

**"%substr(&sysdate9,3)"d** and

**"&sysdate9"d**;

title "All Courses Held So Far This Month";

title2 "(as of &sysdate9)";

**run**;

**proc** **print** data=aa;

where begin\_date between

**"01%substr(&sysdate9,3)"d** and

**"&sysdate9"d**;

title "All Courses Held So Far This Month";

title2 "(as of &sysdate9)";

**run**;

**proc** **print** data=aa;

where begin\_date between

**"14%substr(&sysdate9,3)"d** and

**"&sysdate9"d**;

title "All Courses Held So Far This Month";

title2 "(as of &sysdate9)";

format begin\_date date9.;

**run**;

**data** a;

input name $ age location $;

cards;

hh 34 dallas

jj 45 canada

kk 67 australia

;

**run**;

**proc** **print** data=a;

where upcase(location)="&site";

title "REVENUES FOR &site TRAINING CENTER";

**run**;

%put \_automatic\_;

%put \_user\_;

%put \_global\_;

%put \_local\_;

%put \_all\_;

options symbolgen;

title "Today is &sysdate9.";

title "Today is &systime.";

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO FUNCTIONS** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%let variable =value;

%let k = 1;

%let tot = &k + 1;

%put &tot;

%let tot = %eval(&k + 1);

%put &tot;

%let tot = %sysevalf(2.1 + 1.1);

%put &tot;

options symbolgen;

%let prog=data new; x=**1**; **run**;

&prog

proc print;

**run**;

/\*Method 1\*/

options symbolgen;

%let prog=%str(data new; x=**1**; **run**;);

&prog

proc print;

**run**;

options symbolgen;

%let prog=%str(data new; x=**1**; **run**;);

%put &prog;

/\*Method 2\*/

%let prog=data new%str(;) x=1%str(;)run%str(;);

%put &prog;

/\*Method 3\*/

%let s=%str(;);

%let prog=data new&s x=1&s run&s;

%put &prog;

**proc** **print**;

title "Joan’s Report";

**run**;

options symbolgen;

%let text=Joan%’s Report;

title "&text";

**run**;

%put The value of TEXT is: &text;

options symbolgen;

%let text=%nrstr(Joan%’s Report);

title "&text";

**run**;

%put The value of TEXT is: &text;

**data** team1;

input position : $8. player : $12.;

datalines;

shortstp Ann

pitcher Tom

frstbase Bill

;

**run**;

**proc** **print** data=team1;

title "Report Produced on sysfunc(today(),mmddyy10.)";

**run**;

**proc** **print** data=team1;

title "Report Produced on %sysfunc(today(),mmddyy10.)";

**run**;

**proc** **print** data=team1;

title "Report Produced on %sysfunc(today(),worddate.)"; /\*gives error msg use below function \*/

**run**;

options nodate;

**proc** **print** data=team1;

title "Report Produced on %sysfunc(left(%sysfunc(today(),worddate.)))";

**run**;

/\*Executes functions and masks special characters and mnemonic operators

& % ' " ( ) + - \* / < > = ¬ ^ ~ ; , # blank

AND OR NOT EQ NE LE LT GE GT IN

\*/

options nodate;

**proc** **print** data=team1;

title "Report Produced on %sysfunc(left(%qsysfunc(today(),worddate.)))";

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO IF** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*macro statement: %if\*/

/\*%if %then\*/

**%macro** tq84\_if(val\_one, val\_two);

%if &val\_one gt &val\_two %then

%put "&val\_one > &val\_two";

**%mend** tq84\_if;

%***tq84\_if***(**42**, **10**);

/\*%if %then %else\*/

**%macro** tq84\_if\_then\_else(val\_one, val\_two);

%if &val\_one gt &val\_two

%then %put "&val\_one > &val\_two";

%else %put "&val\_one <= &val\_two";

**%mend** tq84\_if\_then\_else;

%***tq84\_if\_then\_else***(**42**, **10**);

%***tq84\_if\_then\_else***(**42**, **42**);

%***tq84\_if\_then\_else***(**42**, **99**);

/\*%if %then %else %if %then\*/

**%macro** num\_to\_foo\_bar\_baz(num);

%if &num eq **1** %then 'foo';

%else %if &num eq **2** %then 'bar';

%else 'baz';

**%mend** num\_to\_foo\_bar\_baz;

%put %num\_to\_foo\_bar\_baz(1);

%put %num\_to\_foo\_bar\_baz(2);

%put %num\_to\_foo\_bar\_baz(3);

/\*%if %then %do\*/

**%macro** greater(a, b);

%if &a gt &b

%then

%do;

%put a is greater than b;

%put b is less than a;

%end;

%else

%do;

%put a is less or equal than b;

%put b is greater or equal than a;

%end;

**%mend** greater;

%***greater***(**10**, **20**);

%***greater***(**20**, **20**);

%***greater***(**30**, **20**);

/\*Since %eval is only capable of integer arithmetic, an %if statement cannot handle something like %if 5.3 + 2.1 > 4.9:\*/

**%macro** ***tq84\_if***;

%if **39** + **3** = **42**

%then %put indeed, 39 + 3 is 42;

%else %put surprisingly, 39 + 3 is not 42;

/\* The following %if statement does not work; it

will cause the following error message:

ERROR: A character operand was found in the %EVAL

function or %IF condition where a numeric

operand is required.

The condition was:

38.9 + 3.1 = 42

ERROR: The macro TQ84\_IF will stop executing.\*/

%if **38.9** + **3.1** = **42**

%then %put indeed, 38.9 + 3.1 is 42;

%else %put surprisingly, 38.9 + 3.1 is not 42;

**%mend** tq84\_if;

%***tq84\_if***

**%macro** isFooBarOrBaz(txt);

%if %upcase(&txt) eq FOO or

%upcase(&txt) eq BAR or

%upcase(&txt) eq BAZ %then

yes

%else

no

**%mend**;

%put %isFooBarOrBaz(Bar); /\* yes \*/

%put %isFooBarOrBaz(xyz); /\* no \*/

%put %isFooBarOrBaz(BAZ); /\* yes \*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MORE EXMPLES** **MACRO IF** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%let z=3;

**data** test;

x=**1**;

y=**1**;

**run**;

**%macro** diff (v) ;

%if &z=&v %then %do;

proc print data=test;

run;

%end;

**%mend** diff;

%***diff*** (**3**)

/\*check below\*/

%***diff*** (**5**)

**data** test;

if x=**1** ;

**run**;

/\*will not work correctly.\*/

**data** test;

%if x=**1** ;

**run**;

/\*will not work correctly.\*/

**%macro** ***try***;

data test;

%if x=**1**;

run;

**%mend**;

%***try***

**DATA** test;

infile cards delimiter=',';

length codes state $**2** vote **8**;

input codes state vote;

cards;

V, WA, 1235

N, WA, 3210

V, CA, 18990

N, CA, 21222

;

**data** why;

**run**;

**%macro** show(param);

data why;

set test;

if &param=codes then do;

if codes='Y' then do;

vote=vote\***100**;

end;

end;

else if &param=state then do;

if state='CA' then vote=vote\***100**;

end;

else vote=**.**;

proc print data=why;

run;

**%mend**;

%***show***(code)

/\*check below\*/

%***show***(codes)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **CALL SYMPUT AND** **SYMGET** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Returns the value of a macro variable to the DATA step during DATA step execution.

SYMGET(argument)

\*/

options symbolgen;

/\*

CALL SYMPUT(macro-variable, value);

\*/

**data** team1;

input position : $8. player : $12.;

call symput(position,player);

datalines;

shortstp Ann

pitcher Tom

frstbase Bill

;

%put &position.;

%PUT &shortstp &pitcher &frstbase;

**data** team1;

input position $;

datalines;

shortstp

pitcher

frstbase

;

**data** k;

set team1;

call symput('posit',position);

**run**;

%put &posit.;

**data** team2;

input position : $12. player $12.;

call symput('POS'||left(\_n\_), position);

datalines;

shortstp Ann

pitcher Tom

frstbase Bill

;

%put &POS1. &POS2. &POS3.;

**data** c;

input holiday mmddyy.;

call symput('holdate',trim(left(put(holiday,worddate.))));

datalines;

070497

;

**run**;

%put &holdate.;

**data** x;

x='December';

call symput('var',x);

**proc** **print**;

title "Report for &var";

**run**;

**data** char1;

input c $;

call symput('char1',c);

call symput('char2',trim(c));

datalines;

x

;

**run**;

%put char1 = \*\*\*&char1\*\*\*;

%put char2 = \*\*\*&char2\*\*\*;

**data** \_null\_;

x=**1**;

call symput('num1',x);

call symput('num2',left(x));

call symput('num3',trim(left(put(x,**8.**)))); /\*preferred technique\*/

**run**;

%put num1 = \*\*\*&num1\*\*\*;

%put num2 = \*\*\*&num2\*\*\*;

%put num3 = \*\*\*&num3\*\*\*;

**data** aa;

input i k;

cards;

1 1

2 4

3 9

;

**run**;

**data** \_null\_;

set aa;

call symput('var' || put(i, Z1.), k);

**run**;

%PUT &var1 &var2 &var3;

**%MACRO** ***test***;

%DO i=**1** %TO **3**;

%PUT Variable NO. &i: name=var&i, value=&&var&i;

%END;

**%MEND** test;

%***test***

/\*

1) The name of a macro variable, enclosed in quotes but with no ampersand, as follows:

myvar = SYMGET('macrovar');

So, here the DATA step variable MYVAR takes on the value of the macro variable &MACROVAR.

MYVAR will be a character variable, and unless specified otherwise, it will have a length of 200.

2) The name of a DATA step character variable, specified without quotation marks,

which contains the names of one or more macro variables.

The correct functioning of the statement below requires that the value of the

DATA step variable CHARVAR is the name of a macro variable at each iteration of the DATA step:

myvar = SYMGET(charvar);

So, in each iteration of DATA, the value of the DATA step variable CHARVAR supplies the name of a

macro variable – and the value of that macro variable gets assigned to the DATA step variable MYVAR.

3) A character expression that constructs a macro variable name.

The following statement assumes that there are macro variables &V1, &V2, &V3,. . . &Vx

for each iteration of the DATA step:

myvar = SYMGET('v'||LEFT(\_N\_));

\*/

**Example 1 : Creating a single macro variable**

\*\*\*\*\*\*\*\*\*\*\*\*\*Creating a macro variable using call symput;

**data** \_null\_;

set sashelp.class;

if \_N\_ = **1** then do;

call symput('nvar', name);

end;

**run**;

%put &nvar;

\*\*\*\*\*\*\*\*Get macro variable value in a dataset using symget;

**data** want;

var1=symget('nvar');

**run**;

**Example 2 : Creating multiple macro variables**

**data** \_null\_;

set sashelp.class;

call symput('nvars' || strip(\_n\_), name);

**run**;

%put &nvars1 &nvars2 &nvars3;

\* Number of rows;

**data** \_null\_;

if **0** then set sashelp.class nobs=n;

call symput ('nrows',n);

**run**;

\*Get macro variable value in a dataset;

**data** want (drop = i);

do i=**1** to &nrows.;

var1=symget(cats('nvars',i));

output;

end;

**run**;

**data** dusty;

input dept $ name $ salary @@;

datalines;

bedding Watlee 18000 bedding Ives 16000

bedding Parker 9000 bedding George 8000

bedding Joiner 8000 carpet Keller 20000

carpet Ray 12000 carpet Jones 9000

gifts Johnston 8000 gifts Matthew 19000

kitchen White 8000 kitchen Banks 14000

kitchen Marks 9000 kitchen Cannon 15000

tv Jones 9000 tv Smith 8000

tv Rogers 15000 tv Morse 16000

;

**proc** **means** data=dusty noprint;

class dept;

var salary;

output out=stats sum=s\_sal;

**run**;

**proc** **print** data=stats;

var dept s\_sal;

title "Summary of Salary Information";

title2 "For Dusty Department Store";

**run**;

**data** \_null\_;

set stats;

if \_n\_=**1** then call symput('s\_tot',s\_sal);

else call symput('s'||dept,s\_sal);

**run**;

**data** new;

set dusty;

pctdept=(salary/symget('s'||dept))\***100**;

pcttot=(salary/&s\_tot)\***100**;

**run**;

**proc** **print** data=new split="\*";

label dept ="Department"

name ="Employee"

pctdept="Percent of \*Department\* Salary"

pcttot ="Percent of \* Store \* Salary";

format pctdept pcttot **4.1**;

title "Salary Profiles for Employees";

title2 "of Dusty Department Store";

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CALL SYMPUTX \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

Assigns a value to a macro variable, and removes both leading and trailing blanks.

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

\*/

**data** \_null\_;

x=**1**;

call symputx('num1',x);

call symput('num2',left(x));

call symput('num3',trim(left(put(x,**8.**)))); /\*preferred technique\*/

**run**;

%put num1 = \*\*\*&num1\*\*\*;

%put num2 = \*\*\*&num2\*\*\*;

%put num3 = \*\*\*&num3\*\*\*;

**data** \_null\_;

call symputx(' items ', ' leading and trailing blanks removed ',

'lplace');

call symputx(' x ', **123.456**);

**run**;

%put items=!&items!;

%put x=!&x!;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **SERIES OF MACRO VARIABLE**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%let teach=teach3;

%let teach1=Hallis, Dr. George;

%let teach2=Wickam, Dr. Alicee;

%let teach3=Forest, Mr. Peter;

%let CRS =crs1;

%let crs1=3;

%put &&&&teach&&&crs;

&&&&teach&&&crs ->&&teach&crs1 ->&teach3 -> Forest, Mr. Peter

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO REFERENCE** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**tex***t&variable*

*&variable***text**

**tex***t&variable***text**

or referencing adjacent macro variables

*&variable&variable*

%let lib=perm;

%let graphics=g;

%let year=90;

%let month=jan;

%let var=sale;

libname perm "E:\Office\SAS\_Chakra\_Training\SAS Macros\tes";

**data** perm.y&year&month;

input week sale day;

cards;

2 100 1

3 300 2

;

**run**;

**proc** &**graphicschart** data=perm.y&year&month;

hbar week / sumvar=&var;

**run**;

**proc** &**graphicsplot** data=perm.y&year&month;

plot &var\*day;

**run**;

**proc** &graphics.chart data=perm.y&year&month;

hbar week / sumvar=&var;

**run**;

**proc** &graphics.plot data=perm.y&year&month;

plot &var\*day;

**run**;

**proc** &graphics.chart data=&lib.y&year&month;

hbar week / sumvar=&var;

**run**;

**proc** &graphics.plot data=&lib.y&year&month;

plot &var\*day;

**run**;

**proc** &graphics.chart data=permy90jan;

hbar week / sumvar=&var;

**run**;

**proc** &graphics.chart data=&lib.**.y**&year&month;

hbar week / sumvar=&var;

**run**;

**proc** &graphics.plot data=&lib.**.y**&year&month;

plot &var\*day;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO VARIABLE BY SQL** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**data** all1;

input name $;

cards;

kk

jj

hh

;

**run**;

**proc** **sql** ;

select \* from all;

**quit**;

%let name=totfee1;

**proc** **sql** noprint;

select name

into :totfee1

from all1;

**quit**;

%put &totfee1;

**proc** **sql** noprint;

select name

into :totfee1 separated by ','

from all1;

**quit**;

%put &totfee1;

**data** all;

input sal fee;

cards;

100 300

300 100

500 200

;

**run**;

**proc** **sql** noprint;

select sum(sal) format=dollar10. ,

count(fee)

into :totfee, :count1 sepearted by ','

from all;

**quit**;

%let totfee=&totfee; /\* to remove any leading and trailing blanks \*/

%put &totfee;

%put &count1;

**proc** **sql** noprint;

select sal ,

fee

into :totf seperated by ',', :coun seperated by ','

from all;

**quit**;

%put &totf;

%put &coun;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **POSITIONAL AND KEYWORD PARAMETERS**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*Example 1: Using the %MACRO Statement with Positional Parameters \*/

**data** srhigh;

input school $ district $ enrollmt;

cards;

jones CA 123

cambridge MN 234

;

**run**;

**PROC** **PRINT** DATA=SRHIGH;

VAR SCHOOL DISTRICT ENROLLMT;

SUM ENROLLMT;

**RUN**;

**%macro** prnt(var,sum);

proc print data=srhigh;

var &var;

sum &sum;

run;

**%mend** prnt;

%***prnt***(school district enrollmt, enrollmt)

/\*Example 2: Using the %MACRO Statement with Keyword Parameters\*/

**%macro** prnt1(var=school district enrollmt, sum=enrollmt);

proc print data=srhigh;

var &var;

sum &sum;

run;

**%mend** prnt1;

%***prnt1***()

**%macro** prnt1(var=, sum=);

proc print data=srhigh;

var &var;

sum &sum;

run;

**%mend** prnt1;

%***prnt1***(var=school district enrollmt, sum=enrollmt)

**data** yearend;

input expenses division;

cards;

123 1

345 2

456 3

;

**run**;

**%macro** finance(yvar=expenses,xvar=division);

proc plot data=yearend;

plot &yvar\*&xvar;

run;

**%mend** finance;

%***finance();***

**PROC** **PLOT** DATA=YEAREND;

PLOT EXPENSES\*DIVISION;

**RUN**;

%***finance***(xvar=year)

**PROC** **PLOT** DATA=YEAREND;

PLOT EXPENSES\*YEAR;

**RUN**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **MACRO VARIABLE %GLOBL AND %LOCAL**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**%GLOBAL Statement**

**Creates macro variables that are available during the execution of an entire SAS session.**

|  |
| --- |
| Syntax |

|  |
| --- |
| ****%GLOBAL**** *macro-variable-1 <...macro-variable-n>*; |

%macro vars(first=1,last=);

%global gfirst glast;

%let gfirst=&first;

%let glast=&last;

var test&first-test&last;

%mend vars;

When you submit the following program, the macro VARS generates the VAR statement and the values for the macro variables used in the title statement.

proc print;

%vars(last=50)

title "Analysis of Tests &gfirst-&glast";

run;

PROC PRINT;

VAR TEST1-TEST50;

TITLE "Analysis of Tests 1-50";

RUN;

|  |
| --- |
| **Syntax** |
|  |

|  |
| --- |
| ****%LOCAL**** *macro-variable-1 <...macro-variable-n>*; |

%let variable=1;

%macro routine;

%put \*\*\*\*\* Beginning ROUTINE \*\*\*\*\*;

%local variable;

%let variable=2;

%put The value of variable inside ROUTINE is &variable;

%put \*\*\*\*\* Ending ROUTINE \*\*\*\*\*;

%mend routine;

%routine

%put The value of variable outside ROUTINE is &variable;

Submitting these statements writes these lines to the SAS log:

\*\*\*\*\* Beginning ROUTINE \*\*\*\*\*

The value of variable inside ROUTINE is 2

\*\*\*\*\* Ending ROUTINE \*\*\*\*\*

The value of variable outside ROUTINE is 1

**libname mylib 'SAS-data-library';**

**filename prtlast catalog**

**'mylib.macsrc.prtlast.source';**

**%include prtlast / source2;**

**proc sort data=perm.courses out=byfee;**

**by fee;**

**run;**

**%prtlast**

**FILENAME** *fileref* **CATALOG** '*library.catalo*g' ;

**OPTIONS** MAUTOSOURCE SASAUTOS=*fileref* ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Example: Identify words in a list supplied as a macro parameter. Start with the first word and continue as long as more words are found. If no first word is found, display a message

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

libname macrolib 'E:\test\zi';

options mstored sasmstore=macrolib;

**%macro** words(text,root=w,delim=%str( ))/store source;

%local i word;

%let i=1;

%let word=%scan(&text,&i,&delim);

%do %while (&word ne );

%global &root&i;

%let &root&i=&word;

%let i=%eval(&i+1);

%let word=%scan(&text,&i,&delim);

%end;

%global &root.num;

%let &root.num=%eval(&i-1);

**%mend** words;

**%macro** words(text,delim=%str( ));

%local i word;

%let i=1;

%let word=%scan(&text,&i,&delim);

%do %while (&word ne );

%put Word number &i is: &word;

%let i=%eval(&i+1);

%let word=%scan(&text,&i,&delim);

%end;

%if &i=**1** %then %put Text is blank.;

**%mend** words;

%***words***()

/\*

Accessing Stored Compiled Macros

In order to access a stored compiled macro, you must

assign a libref to the SAS library that contains a Sasmacr catalog in which the macro was stored

set the system options MSTORED and SASMSTORE=libref

call the macro.

Only one permanent catalog containing compiled macros can be accessed at any given time.

\*/

libname macrolib 'E:\test\zi';

options mstored sasmstore=macrolib;

%***words***(This is a test)

%put Number of Words (wnum): &wnum;

%put Word Number 1 (w1): &w1;

%put Word Number 2 (w2): &w2;

%put Word Number 3 (w3): &w3;

%put Word Number 4 (w4): &w4;

**proc** **catalog** cat=macrolib.sasmacr;

contents;

title "Stored Compiled Macros";

**quit**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Another Example\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

filename grp5mac "E:\Office\SAS\_Chakra\_Training\SAS Macros\tes";

filename prj5Amac "E:\Office\SAS\_Chakra\_Training\SAS Macros\tes";

options mautosource sasautos=(prj5amac, grp5mac, sasautos);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MACRO WORK LIBRARY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** ***compiled\_macro*** ;

%put Now executing compiled\_macro ;

**%mend**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MACRO PERMANENT LIBRARY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

libname macrocmp "E:\Office\SAS\_Chakra\_Training\SAS Macros\tes";

options mstored sasmstore = macrocmp;

**%macro** ***compiled\_macro*** /store des = "Meaningful macro description";

%put Now executing compiled\_macro ;

**%mend**;

**proc** **catalog** catalog=macrocmp.sasmacr;

contents;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOCAL MACRO WITH PARAMETER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*1234.5---- Roman12 ----MCCXXXIV (M=1000 CC=100, XXX=10\*/

/\*2000---- Roman12 ----MCCXXXIV (MM=2000\*/

**%macro** roman(start=**1**,stop=**10**,incr=**1**);

%local i;

data \_null\_;

%do i=&start %to &stop %by &incr;

value=&i;

put

"Roman form of &i is " value roman12.;

%end;

run;

**%mend** roman;

options mprint;

%***roman***(start=**2000**,stop=**2006**,incr=**2**)

**data** all;

input course\_number course\_code $ course\_title $**9**-**33** days fee location $ Begin\_date $**49**-**57** Teacher $**59**-**75** name $**77**-**96** company $**98**-**123** @**128** paid $;;

cards;

1 c001 basic telecommunications 3 795 seattle 23oct2000 Hallis, Dr.George albritton,Mr.bryan special services y

2 c001 structured query language 4 1150 dallas 04dec2000 Wickam, Dr.Alice chodnoff,Mr.norman NBA insurance y

3 c003 local area networks 3 650 Boston 08jan2001 Forest, Mr.peter clark,Mr.rich assoc. of realtors y

4 c004 database design 2 375 seattle 22jan2001 Tally, Ms.judy crace,Mr.ronch von crump seafood y

5 c005 computer aided design 5 1600 dallas 26feb2001 Hallis, Dr.George dellmonache,Ms.susan Us express corp. y

6 c006 basic telecommunications 3 795 Boston 02apr2001 Berthan, Ms.judy dixon,Mr.matt southern edison y

7 c001 medium telecommunications 3 793 dallas 21may2001 Hallis, Dr.George edwards,Mr.charles gorman tire corp. n

8 c002 High telecommunications 3 791 Boston 11jun2001 Wickam, Dr.Alice edwards,Mr,sonia animal hospital y

9 c003 basic telecommunications 3 792 seattle 16jul2001 Forest, Mr.peter elsins,ms.marisa F. sss inc. y

10 c004 medium telecommunications 3 790 Dallas 13aug2001 tally, Ms.julia griffin,mr.lantz quantum corporation y

11 c005 High telecommunications 3 795 Boston 17sep2001 Tally, Ms.julia hall,ms.sharon cetadyne technologies y

12 c006 local area networks 3 640 seattle 01oct2001 Berthan, Ms.julia haubold,ms.ann reston railway n

13 c001 High area networks 3 630 Boston 12nov2001 Hallis, Dr.George hodge,ms.rita wilbur right airforce base n

14 c002 High area networks 3 620 seattle 03dec2001 Wickam, Dr.Alice knight,Ms.susan k&p products y

15 c003 High area networks 3 610 Dallas 07jan2002 Forest, Mr.Peter koleff,mr.jim emulate research y

16 c004 High area networks 3 660 Boston 21jan2002 Tally, Ms.Julia laiken,mr.jim lorus toy co. y

17 c005 local area networks 3 670 seattle 25feb2002 Hallis, Dr.George mcgillivary,Ms.kathy allied wood corporation n

18 c006 local area networks 3 690 Dallas 25mar2002 Berthan, Ms.Judy merenstein,mr.w. dunnely consultants y

;

**run**;

**data** schedule;

input course\_number course\_code $ location $ Begin\_date $**17**-**25** Teacher $**27**-**44** ;

cards;

1 c001 seattle 23oct2000 Hallis, Dr.George

2 c001 dallas 04dec2000 Wickam, Dr.Alice

3 c003 Boston 08jan2001 Forest, Mr.peter

4 c004 seattle 22jan2001 Tally, Ms.judy

5 c005 dallas 26feb2001 Hallis, Dr.George

6 c006 Boston 02apr2001 Berthan, Ms.judy

7 c001 dallas 21may2001 Hallis, Dr.George

8 c002 Boston 11jun2001 Wickam, Dr.Alice

9 c003 seattle 16jul2001 Forest, Mr.peter

10 c004 Dallas 13aug2001 tally, Ms.julia

11 c005 Boston 17sep2001 Tally, Ms.julia

12 c006 seattle 01oct2001 Berthan, Ms.julia

13 c001 Boston 12nov2001 Hallis, Dr.George

14 c002 seattle 03dec2001 Wickam, Dr.Alice

15 c003 Dallas 07jan2002 Forest, Mr.Peter

16 c004 Boston 21jan2002 Tally, Ms.Julia

17 c005 seattle 25feb2002 Hallis, Dr.George

18 c006 Dallas 25mar2002 Berthan, Ms.Judy

;

**run**;

**data** courses;

input course\_code $ course\_title $**6**-**31** days fee;

format fee dollar5.;

cards;

c001 basic telecommunications 3 795

c002 structured query language 4 1150

c003 local area networks 3 650

c005 database design 2 375

c006 computer aided design 5 1600

;

**run**;

**data** student;

input student\_name $**1**-**21** student\_company $**22**-**50** @**52** paid $;

cards;

albritton,Mr.bryan special services y

amigo,Mr.bail assoc. of realtors n

chodnoff,Mr.norman NBA insurance y

clark,Mr.rich assoc. of realtors y

crace,Mr.ronch von crump seafood y

dellmonache,Ms.susan Us express corp. y

dixon,Mr.matt southern edison y

edwards,Mr.charles gorman tire corp. n

edwards,Mr,sonia animal hospital y

elsins,ms.marisa F. sss inc. y

griffin,mr.lantz quantum corporation y

hall,ms.sharon cetadyne technologies y

haubold,ms.ann reston railway n

hodge,ms.rita wilbur right airforce base n

knight,Ms.susan k&p products y

koleff,mr.jim emulate research y

laiken,mr.jim lorus toy co. y

mcgillivary,Ms.kathy allied wood corporation n

merenstein,mr.w. dunnely consultants y

pancoast,Ms.jane chase information technology n

summer,mr.kenneth nbainsurance y

washington,mr.robert federal landmarks y

wurzelbacher,mr.phil a1's discount clothing y

;

**run**;

**%macro sites;**

**%local i;**

**proc sort data=perm.schedule(keep=location)**

**out=values nodupkey;**

**by location;**

**run;**

**data \_null\_;**

**set values end=last;**

**call symput('site'||left(\_n\_),trim(location));**

**if last then call symput('count',\_n\_);**

**run;**

**%do i=1 %to &count;**

**%put SITE&i is &&site&i;**

**%end;**

**%mend sites;**

**%sites**

**data** dates;

input date $;

datalines;

10nov97

11nov97

12nov97

;

**data** reptdata;

input date $ var1 var2;

datalines;

10nov97 25 10

10nov97 50 11

11nov97 23 10

11nov97 30 29

12nov97 33 44

12nov97 75 86

;

**%macro** rept(dat,a,dsn);

proc chart data=&dsn;

title "Chart for &dat";

where(date="&dat");

vbar &a;

run;

**%mend** rept;

**data** \_null\_;

set dates;

call execute('%rept('||date||','||'var1,reptdata)');

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ERROR IN BELOW AS % is missing in then\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**data** yearend;

input expenses division;

cards;

123 1

345 2

456 3

345 2

499 7

348 5

433 4

;

**run**;

**%macro** ***printit***;

%if &syslast ne \_NULL\_ then %do;

proc print data=\_last\_(obs=**5**);

title "Last Created Data Set Is &syslast";

run;

%end;

**%mend**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*% ADDED in then\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** ***printit***;

%if &syslast ne \_NULL\_ %then %do;

proc print data=\_last\_(obs=**5**);

title "Last Created Data Set Is &syslast";

run;

%end;

**%mend**;

%***printit***

**%macro** ***printit***;

%if &syslast ne \_NULL\_ %then %do;

proc print data=\_last\_(obs=**5**);

title "Last Created Data Set Is &syslast";

run;

%end;

**%mend**;

options mprint mlogic symbolgen;

%***printit***

**data** x;

y=**1**;

**run**;

%***printit***

/\*

The MONYYw. format writes SAS date values in the form mmmyy or mmmyyyy, where

mmm

is the first three letters of the month name.

yy or yyyy

is a two-digit or four-digit integer that represents the year.  
put date monyy5.;

MAY12

put date monyy7.;

MAY2012

\*/

**%macro** ***reports***;

proc sort data=all out=current;

where put(begin\_date,monyy7.)=

"%substr(&sysdate9,3,7)"

and begin\_date ge **"&sysdate9"d**;

by begin\_date location course\_title;

run;

proc print data=current noobs n;

by begin\_date location course\_title;

var name company paid;

title "Course Registration as of &sysdate";

run;

%if &sysday=Friday %then %do;

proc sort data=all out=current;

where put(begin\_date,monyy7.)=

"%substr(&sysdate9,3,7)"

and begin\_date le **"&sysdate9"d**;

by begin\_date location course\_title;

run;

proc means data=current maxdec=**0** sum;

by begin\_date location course\_title;

var fee;

class paid;

title "Revenue for Courses as of &sysdate9";

run;

%end;

**%mend** reports;

**%macro** ***reports***;

%include 'file-containing-daily-report-code';

%if &sysday=Friday %then %do;

%include 'file-containing-weekly-report-code';

%end;

**%mend** reports;

options mprint;

%***reports***

**%macro** ***daily***;

proc sort data=perm.all out=current;

where put(begin\_date,monyy7.)=

"%substr(&sysdate9,3,7)"

and begin\_date ge **"&sysdate9"d**;

by begin\_date location course\_title;

run;

proc print data=current noobs n;

by begin\_date location course\_title;

var name company paid;

title "Course Registration as of

&sysdate9";

run;

**%mend** daily;

**%macro** ***weekly***;

proc sort data=perm.all out=current;

where put(begin\_date,monyy7.)=

"%substr(&sysdate9,3,7)"

and begin\_date le **"&sysdate9"d**;

by begin\_date location course\_title;

run;

proc means data=current maxdec=**0** sum;

by begin\_date location course\_title;

var fee;

class paid;

title "Revenue for Courses as of &sysdate9";

run;

**%mend** weekly;

**%macro** ***reports***;

%***daily***

%if &sysday=Friday %then %do;

%***weekly***

%end;

**%mend** reports;

%***reports***

**%macro** ***prtlast***;

%if &syslast ne \_NULL\_ %then %do;

proc print data=&syslast(obs=**5**);

title "Listing of &syslast data set";

run;

%end;

%else

%put No data set has been created yet.;

**%mend**;

%include 'external-file' / source2;

**proc** **sort** data=perm.courses out=bydays;

by days;

**run**;

%***prtlast***

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DELETE A MACRO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Deletes the specified variable or variables from the macro global symbol table.**

****%SYMDEL**** *macro-variable-1 <...macro-variable-n>*</option>;

***options***

NOWARN

suppresses the warning message when an attempt is made to delete a non-existent macro variable

%let city = hyd;

%symdel city;

**%macro** ***mm***;

%Put &syscc;

**%mend**;

**proc** **catalog** c=work.sasmacr kill;

**run**;

**quit**;

**proc** **catalog** c=work.sasmacr kill;

**run**;

**quit**;