

# AUTOCALL MACROS- A QUICK OVERVIEW

Vinod Panangattiri Parambil, Roche Products Limited, UK

## 1 – AUTOCALL MACRO

- ❑ **Autocall macro** allows you to use the same SAS macro code for multiple SAS programs by storing the macro in a particular location.
- ❑ By default a set of autocall macros are included in a library, called “**Autocall Library**” within the SAS system supplied by SAS Institute
- ❑ The autocall library must be included in the list of search in order to make use of the **autocall macros**
- ❑ The following SAS options are required to access the autocall macros:

- ❖ **MAUTOSOURCE** : tells SAS to activate the autocall facility
- ❖ **SASAUTOS** : tells SAS where to look for the macros

```
options mautesource;
options sasautos = Library #1;
```

**NOMAUTOSOURCE** option specifies that the autocall facility is not available

- ❑ The following autocall macros are discussed:

- ❖ **%CMPRES**
- ❖ **%LEFT**
- ❖ **%LOWCASE**
- ❖ **%TRIM**
- ❖ **%VERIFY**

## 2 – %CMPRES

- ❑ **%CMPRES** macro returns the argument passed to it in an unquoted form with multiple blanks compressed to single blanks and with leading and trailing blanks removed

- ❑ Syntax: `%cmpres(argument);`

- ❑ Example:

```
proc sql noprint;
  select count(type) into : num
  from sashelp.vcolumn
  where libname = "SASHELP" and memname = "CLASS" and type = "num";

  select count(type) into : char
  from sashelp.vcolumn
  where libname = "SASHELP" and memname = "CLASS" and type = "char";
quit;

%let x = %str(SASHELP.CLASS contains &num numeric and &char character variables);
%let y = %cmpres(&x);

%put ***x = &x ***;
%put ***y = &y ***;
```

The following lines are written to the log file:

```
***x = SASHELP.CLASS contains      3 numeric and      2 character variables***
***y = SASHELP.CLASS contains 3 numeric and 2 character variables***
```

## 3 – %LEFT

- ❑ **%LEFT** macro returns the argument passed to it without any leading blanks in an unquoted form

- ❑ Syntax: `%left(argument);`

- ❑ Example:

```
proc sql noprint;
  select nobobs into : nobobs
  from sashelp.vtable
  where libname = "SASHELP" and memname = "CLASS";
quit;

%let x = %str(SASHELP.CLASS contains &nobobs observations);
%let y = %str(SASHELP.CLASS contains %left(&nobobs) observations);

%put ***x = &x ***;
%put ***y = &y ***;
```

The following line is written to the log file:

```
***x = SASHELP.CLASS contains      19 observations ***
***y = SASHELP.CLASS contains 19 observations ***
```

## 4– %LOWCASE

- ❑ **%LOWCASE** macro returns the argument passed to it unchanged except that all upper case alphabetic characters are changed to their lower case equivalents

- ❑ Syntax: `%lowcase(argument);`

- ❑ Example:

```
proc sql noprint;
  select distinct (name) into : vars separated by " "
  from sashelp.vcolumn
  where libname = "SASHELP" and memname = "CLASS";
quit;

%let x = &vars;
%let y = %lowcase(&vars);

%put ***x = &x***;
%put ***y = &y***;
```

%put writes the following lines to the log file:

```
***x = AGE HEIGHT NAME SEX WEIGHT***
***y = age height name sex weight***
```

## 5– %TRIM

- ❑ **%TRIM** macro returns the argument passed to it without any trailing blanks in an unquoted form

- ❑ Syntax: `%trim(argument);`

- ❑ Example:

```
data _null_;
  a = 1;
  call symput ('x', a);
run;

%put ***&x***;
%put ***%trim(&x)***;
```

The following lines are written to the log file:

```
***      1***
***1***
```

## 6– %VERIFY

- ❑ **%VERIFY** macro returns the position of the first character in the argument that is not in the target value
- ❑ If every character in the argument is also in the target value, then the **%VERIFY** returns a value of 0

- ❑ Syntax: `%verify(argument,target);`

- ❑ Example:

```
%let scanlist = ABC123;
%let text1 = ABC_1;
%let text2 = AC23;

%put ***%verify(&text1, &scanlist)***;
%put ***%verify(&text2, &scanlist)***;
```

The following lines are written to the log file:

```
***4***
***0***
```