CLASS-32

MACROS CONTD.

Piping : The output of one command becomes the input for the second command.

How to invoke DOS from SAS.

In order to create folder on desktop:

x mkdir "..... folder path ……...test1"; // 'x' command is an interface between sas and dos. It is used to invoke DOS and mkdir will make the folder named test1 on desktop.
The moment you run this command, DOS will get activated and will wait for the exit command. Type exit, test1 folder will be created at this moment on the desktop//

In order to remove the created folder test1:
x rmdir "..... folder path ……...test1"; // RMDIR will remove the folder test1//

NOXWAIT OPTION :

Option noxwait; // noxwait option is a global option which is valid for the entire session. By this option we don't have to write the exit command manually to close the terminal. Terminal will come for a fraction of seconds and will automatically go by using "noxwait option " //
x mkdir "..... folder path ……...test1";

Macro to create 5 folders on desktop:

```
Option noxwait;
%macro folder;  // name of the marco is folder //
%do i=1 %to 5;  // loop will execute 5 times //
x mkdir ".....  folder path…….a&i. " ;  // 5 folders will be created with names a1 to a5 , terminal will come 5 times and will automatically go because of noxwait option //
%end;
```

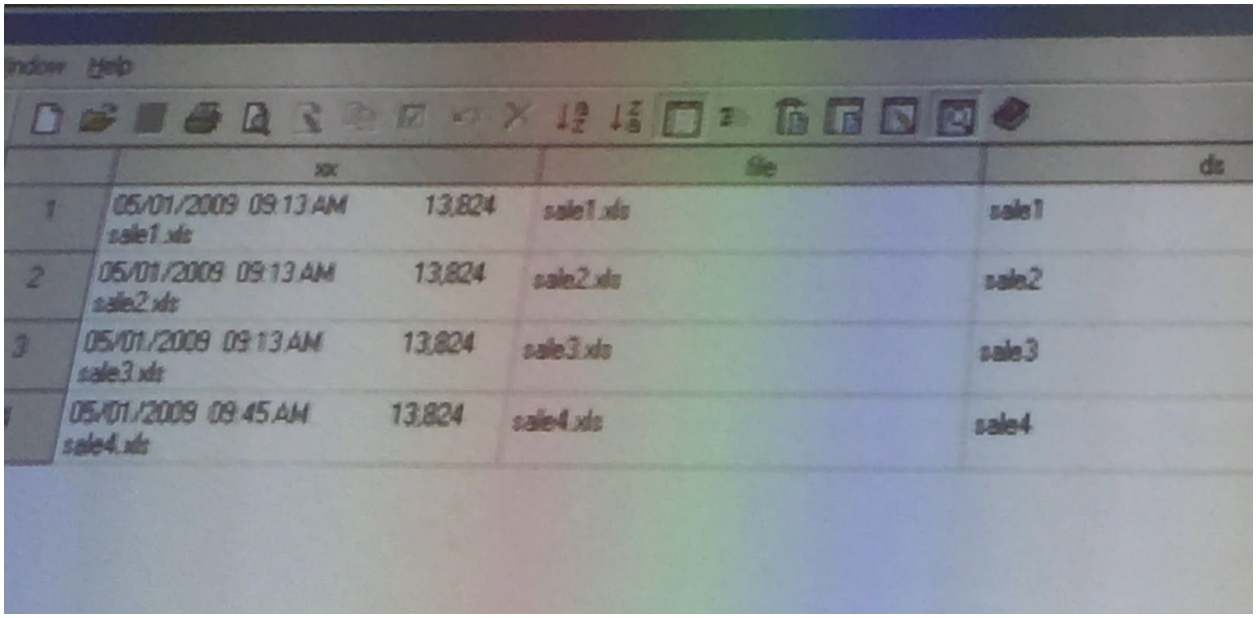
```
%mend folder;
%folder;
```

Suppose we have a folder on desktop named " all " which have multiple files in it of multiple types like excel,pdf etc. Out of those multiple files , we want to pick the excel files starting with name " sale" .

```
Filename yo pipe 'dir  ".....  folder path…….\all";  // dir command will display the list of files in the folder "all" , 'pipe' (>) command will store the 'dir' output in 'yo' //

%macro fetch;    // Fetch is the name of the macro //
Data temp;    // temp dataset is created which will have the variable 'xx' in it //
Infile yo truncover;
Input xx $1-100;    // xx variable is created of length 100 bytes//
If index(xx, '.xls") and index( xx, "sale");    // index function will look out for the particular keyword in variable, filter '.xls' and 'sale' will give the excel files starting with the name "sale"//.
file=scan(xx,-1," ");   // scanned from backside , with this we can get the only the file name eg.sale1, sale2 //
ds=scan(file,1,".");    // scan file, in file we have the values like sale1.xls etc taking ' . ' as delimiter , this will give the value sale1, sale 2 etc.. //
Run;

Proc sql noprint;
Select count(*) into: n from temp;  // count(*) will give the number of rows //
Select file into: f separated by "@"from temp;  // file have values sale1.xls… so on and separated by "@" will create a string.//
Select ds into: d separated by "@" from temp;  // ds is having the values sale1,sale2...//
Quit;
%do i=1 %to &n; // loops will execute 4 times as we have to import 4 files from sale1 to sale 4 //
%let fi= %scan(&f,&i, "@" );  // from sale1.xls@ we will get the value sale1.xls….//
%let di= %scan(&d,&i, "@" );   // from sale1@ we will get the value sale1...//

Proc import out=&di DATAFILE= "..... path name…….all\&fi."; // proc import out=sale1 , datafile = sale1.xls //
                DBMS=EXCEL REPLACE;
                SHEET= "sheet1$" ;
                GETNAMES=YES;
RUN;
%end;
```

%mend fetch;
%fetch;

Note : If it is unix based then from the first line just change command from “dir” to “ls”. Rest is same.



Batch process: We want the codes to run one after the other. Suppose in a file we have 2 folders named new1 and new 2 containing the data:

New1	New2
Data a; Set sasuser.admit; run;	Data b; Set sasuser.admit; run;

We want that dataset ‘a’ and ‘b’ should get created in work one after the other after running the code.

Filename yo pipe ‘dir “..... folder path …….all” ‘ ;
%macro fetch;
Data temp;
Infile yo truncover;
Input xx $1-100;
If index(xx, ‘.sas’) and index(xx, “new”); // we will get new1.sas and new2.sas //

```
file=scan(xx,-1," ");
Run;


Proc sql noprint;
Select count(*) into: n from temp;
Select file into: f separated by "@"from temp;
Quit;
%do i=1 %to &n;  // loop will run 2 times //
%let fi= %scan(&f,&i, "@" );

%include "..... folder path …….all\&fi.";  // %include is used to execute the code //

%end;

%mend fetch;
%fetch;
```

SCOPE OF MACRO VARIABLE:

```
%let a=Ram;  // global macro //

%macro abc;
%let b = kumar;  // local macro//

Data a;
Run;
%mend abc;

%abc;
Data &a;  // macro ram will become //
Run;

Data&b;  // b is a local macro variable. At this step it will show error , if b will resolve within a
macro only then it will take the value as 'kumar; //
Run;
```

5th way to create a macro-variable: % global statement (When internally we make macro-variable and we want it to exist globally)

```
 %let a=Ram;  // global macro //

%macro abc;
%global b;  // This will make 'b' a global macro-variable //
%let b = kumar;

Data a;
Run;
%mend abc;

%abc;
Data &a;
Run;

Data&b;  // b will take the value 'kumar; //
Run;
```

If a global macro-variable name matches with the local macro-variable name, then local overwrites the global.

```
%let a=Ram;  // global macro //

%macro abc;

%let a=kumar;  // local macro-variable//  [ Name of both local and global macro-variable is 'a' ]

Data &a; // 'a' will take the value kumar //
Run;

%mend abc;

%abc;
Data &a;  // here also 'a' will take the value kumar as local overwrites global and the name of macro-variable is same i.e. "a" //
Run;
```

6th way to create a macro-variable: % local statement

```
%let a=Ram;  // global macro //

%macro abc;
%local a;  // writing this will keep macro-variable created inside as local , even if the name is same //
%let a=kumar;  // local macro-variable//

Data &a; // 'a' will take the value kumar //
Run;

%mend abc;

%abc;
Data &a;  // a will take the value ram //
Run;
```