SAS® GLOBAL FORUM 2015

The Journey Is Yours

Yes, SAS® Can Do!

--- Manage External Files With SAS Programming (Paper 3262 - 2015)

Justin Jia, TransUnion Canada
Amanda Lin, CIBC, Canada

April 26-29, 2015
Dallas, Texas, USA

SAS GLOBAL FORUM

Overview

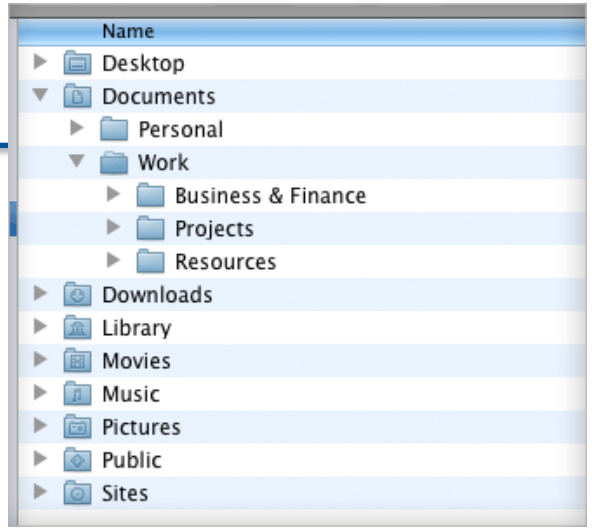
- **Introduction**
- **I. System Statements and Shell Commands**

  X command and statement / %Sysexec macro statement/ Systask command
- **II. SAS Piping Functionality**
- **III. SAS Call System Routine and System Function**
- **IV. New SAS File Management Functions**

  DCREATE function/ RENAME function/ FDELETE function/DLCREATEDIR option
- **V. Application Examples**
- **Conclusion**

SAS GLOBAL FORUM

Introduction



- A good file management system is important for efficient documentation and project management.
- It would be beneficial to automate and standardize repetitive routine tasks via SAS programming: **create, rename, move, copy, delete**.
- Proc DATASETS, though powerful, is only effective for managing SAS files.
- How can we manage non-SAS external files or directories?

I. System Statements and Shell Commands

- **X <'command '> ;**

```
X " mkdir "&path.\ WLMU 999"  ";
```

- **%Sysexec macro statement**

```
%sysexec  mkdir  "&path.\WLMU 999" ;
```

- **Systask statement**

```
 systask command    %unquote(%str(%')   mkdir
"&path.\WLMU 999" %str(%'));
```

SAS GLOBAL FORUM

Pros / Cons of System Statements/Commands

- **Advantages**

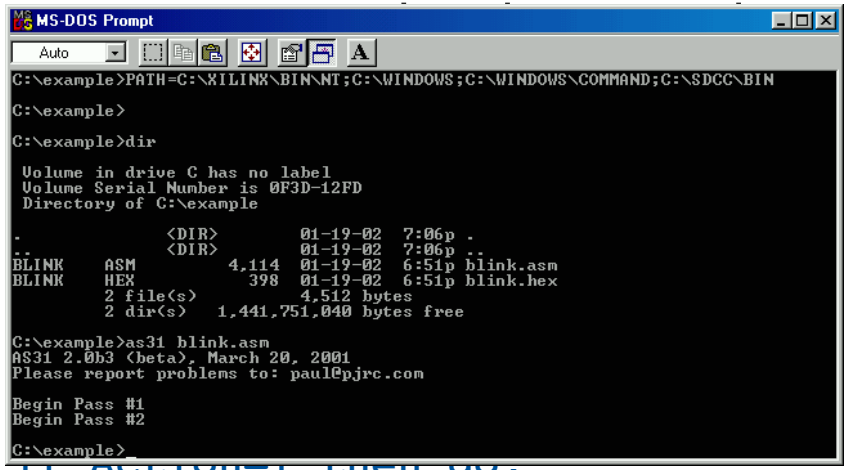
  They use simple and easy DOS or Unix commands.

  Able to perform all kinds of file management tasks (create, rename, copy, move, delete).

  They are global commands and can exist anywhere in a SAS program.

- **Disadvantages**

  They prompt annoying flashing DOS window.

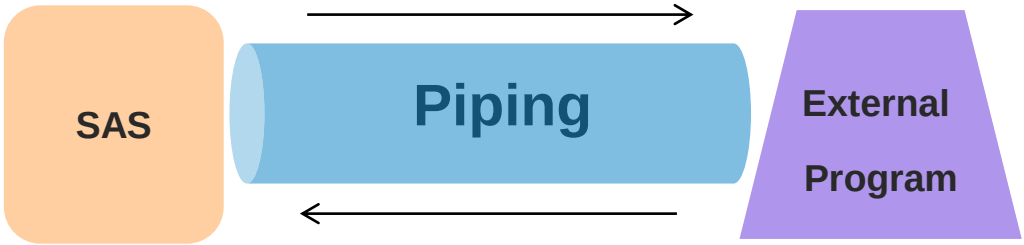



```
X "mkdir  "&path.\Try"  " ;
end;     run;
```

The X command will be executed every time and the new directory Try will be created regardless of the value of Action.

II. SAS Piping Functionality

- Piping is a channel of parallel communications between SAS and other external applications.
- The issued command will be directed to external programs, and the execution results will feed back to SAS via piping.
- We can use SAS Filename statement to invoke unnamed piping.



Unnamed Piping

FILENAME fileref PIPE
'program name /command '
<options> ;

Manage External Files via SAS Piping

```
***** Create a new directory.*****;

Filename Create  PIPE
"mkdir  &path.\By_Pipe";

data _null_ ;
infile Create;
input ;
put _infile_; *Write the
execution results to SAS
log.
run;
```

Error messages will write to SAS log for unsuccessful executions. For example, if the directory already exists:

```
16          Filename Create PIPE     "mkdir
&path.\By_Pipe ";
23          data Create  ;
24          infile Create;
25          input ;
26          put  _infile_;
27      run;

NOTE: The infile CREATE is:
  Unnamed Pipe Access Device,
  PROCESS=mkdir  D:\WLMU2015\Projects\By_Pipe,
RECFM=V, LRECL=256
Stderr output:
A subdirectory or file
D:\WLMU2015\Projects\By_Pipe already exists.
NOTE: 0 records were read from the infile CREATE.
NOTE: The data set WORK.CREATE has 0 observations
and 0 variables.
```

III. SAS Call Routine and System Function

- **Call System Routine**

  CALL SYSTEM (command);

  Command=character string/ expression or a character variable.

  It is a local command rather than a global one, it must be used in a DATA step.

  It is a callable routine, allowing conditional execution.

- **System Function**

  It issues an operating system command for execution during a SAS session.

```
data Create;
Name="pid1280";
NewDir= "&path.\"||strip(Name);
Check= fileexist(NewDir);
if Check=0 then do;
command="mkdir "||'"'||NewDir||'"';
call system(command);
RC=symget("SYSRC");
/*RC= &SYSRC;*/   Not working!
/*RC=system(command);*/
if RC="0" then put "Note: Directories
Created Successfully: " NewDir;
else put "Note: Directories Cannot Be
Created: " NewDir;
end;

else if Check=1 then put "Note:
Directory Already Exists: " NewDir;
Run;
```

IV. New SAS File Management Functions

- From SAS 9.2, some new functions are introduced for file management uses.
- They can work on both SAS files and non-SAS external files.
- They are true SAS functions with simple syntax, easy to use.
- Local and callable commands allowing conditional execution.

DCREATE Function: create a new directory.

Syntax: DCREATE (directory-name, <parent-directory>) ;

```
data _Null_;
pid="PID9999";
NewDir="&path.\"||strip(pid);
Check=fileexist(NewDir);
if Check=0 then do;
Created=dcreate(strip(pid),"&path.");
end;
else if Check=1 then put "Note:
Directory Already Exists: " NewDir;
run;
```

***Note: If successful, the DCREATE function returns a complete pathname of a new external directory; a blank text string for unsuccessful execution.**

RENAME Function

- **A new function since SAS 9.2.**
- **Powerful and capable of working on both SAS files and external files, directories.**
- **Easy and convenient to use.**
- **It returns a return code of zero for successful execution, and non-zero value for unsuccessful execution.**

RENAME Function: rename a SAS file, external file or directory.

Syntax:

RENAME(old-name, new-name , < type>);

Old-Name: the current name of a file or directory. It supports library reference use.

New-Name: the new name of a file or directory.

Type: specifies the element type, the possible values are DATA, VIEW, CATALOG, ACCESS, FILE.

SAS GLOBAL FORUM

Rename files and directories

```
libname Test"&path.\Test"; *define a libref.

data _null_;
/***rename a SAS data set using libref.;***/
RC1=rename("Test.AAA", "BBB", "data");

/*rename a SAS data set using the full pathname.;*/
RC2=rename("&path.\Test\AAA.sas7bdat", "&path.\Test\BBB.sas7bdat", "file");

/***rename an external file.;/
RC3=rename("&path.\Test\Sales.pdf", "&path.\Test\2012 Sales.pdf", "file");

/***rename a Windows directory.;/
RC4=rename("&path.\Test", "&path.\New_Test", "file");

/***rename a Unix directory.;/
RC5=rename("/&path./Test", "/&path./New_Test", "file");
Run;
```

Use Rename Function to move and delete files.

```
RENAME(old-name, new-name, <type>);
```

We can use RENAME function to move and delete files if we specify the New Name with a path different from that of the Old Name.

This is a creative use of RENAME function discovered by the paper authors.

```
data _null_;

/**move an external file to a different
directory and change its name.; */

RC3=rename("&path.\Test\Sales.pdf",
"&path.\New Test\2012 Sales.pdf", "file");

/***move a directory along with its
contents to a different directory, and
also change its name. ;*/

RC6=rename("&path.\Test\AAA", "&path.\New
Test\BBB", "file");

/***delete an external file by moving it
to a specified Trash directory.;*/

RC4=rename("&path.\Test\Sales.pdf",
"&path.\Trash\Sales.pdf", "file");

Run;
```

SAS GLOBAL FORUM

FDELETE Function and DLCREATEDIR Option

Syntax: FDELETE(fileref|directory);

A new function since SAS 9.2, it can permanently delete SAS or non-SAS files and empty directories, must use a fileref rather than a physical name.

It returns 0 for a successful operation and other value for an unsuccessful operation.

```
filename DEL "&path.\Test\Sales.pdf";

data _Null_;
RC=fdelete("DEL");
if RC=0 then put "Note: Files deleted
successfully. ";
else put "Files cannot be deleted.";
Run;
```

New global system option in SAS 9.3. When put in effect, the LIBNAME can automatically create a new directory and then assign the libref, if it does NOT exist.

The default NODLCREATEDIR system option can cancel this effect.

```
options dlcreatedir;
libname AAA "&path.\AAA";

NOTE: Library AAA was created.
NOTE: Libref AAA was successfully
assigned as follows:
Engine: V9
Physical Name:
D:\WLMU2015\Projects\AAA.
```

Summary of Advantages and Disadvantages

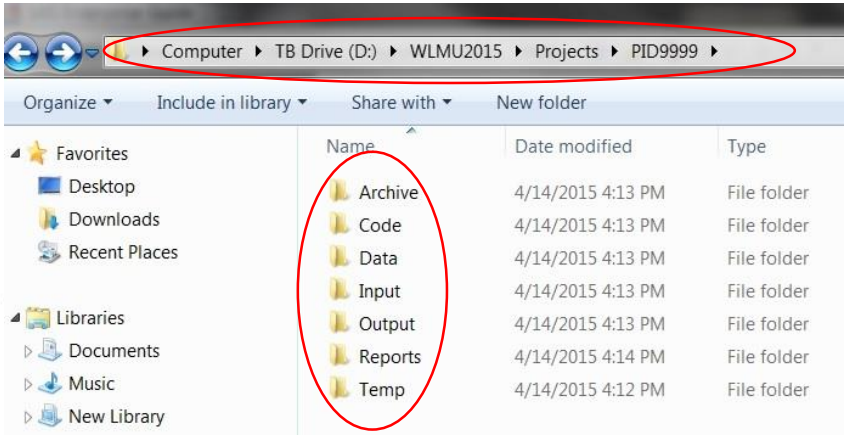
	Advantages	Disadvantages
I. Systems Statements and Shell Commands	1. They use simple and easy DOS, Unix or other OS commands. 2. They are global commands and can exist anywhere in a SAS program. 3. They are able to perform all kinds of file management tasks (create, rename, copy, move, delete).	1. They prompt annoying flashing DOS window. 2. It is not easy to monitor the execution status of the submitted command (no return code). 3. They cannot perform conditional execution since they are global.
II. SAS Piping Functionality	1. It uses simple DOS, Unix or other OS commands, no fleeting DOS window. 2. We can look into SAS log to monitor the execution status. 3. It is able to perform all kinds of file management tasks (create, rename, copy, move, delete).	1. It requires more SAS coding than other methods. 2. It cannot perform conditional execution since it is global.
III. SAS Call Routine and System Function	1. They use simple DOS, Unix or other OS commands. 2. We can look into the return code to monitor the execution status. 3. They are able to perform all kinds of file management tasks (create, rename, copy, move, delete). 4. They are local and callable commands, allowing conditional execution.	1. They are local and must exist in a SAS DATA step. 2. They trigger the annoying fleeting DOS window.
IV. New SAS File Management Functions	1. They are true SAS functions with simple syntax, easy to use. 2. No fleeting DOW window. 3. They can work on both SAS files and non-SAS external files. 4. They are local and callable, allowing conditional execution. 5. We can look into the return code to monitor the execution status.	1. They are local and must exist in a SAS DATA step. 2. They cannot perform all the file management tasks. 3. Only few functions are available for use in SAS.

V. Application Example I: Create Routine Folder Structure

For each project, we need create below folder structure.

Below Excel file controls the conditional creation of wanted directories.

No	Folder_Name	Create
1	Archive	Y
2	Code	Y
3	Data	Y
4	Input	Y
5	Output	N
6	Reports	Y




```
%macro Create(path=, pid=);
options mprint symbolgen ;

Filename List "physical path\List.xls";
proc import datafile=List out=List DBMS=Excel
replace; getnames=Yes;  guessingrows=1000;  run;
data Create;
set List;
Project_Folder="&path.\"||"&pid.";
Folder_Name=upcase(Folder_Name);
if Create="Y" then do;
NewDir=Project_Folder||"\"||strip(Folder_Name);
Sub_Check=fileexist(NewDir);
if Sub_Check=0 then do;
Created=dcreate(strip(Folder_Name),
Project_Folder);
if not missing(Created) then put "Note: Directory
created successfully: " NewDir;
else put "Note: Directory cannot be created: "
NewDir;
end;
if N=1 then do;
Check=fileexist(Project_Folder);
if Check=0 then do;
put "Note: Project folder does not exist "
Project_Folder;
Created=dcreate(strip("&pid."), "&path.");
if not missing(Created) then put "Note: Project
folder created successfully: " Project_Folder;
else put "Note: Project folder cannot be created:
" Project_Folder;
end; end;
else if Sub_Check=1 then put "Note: Directory
already exists: " NewDir;
run;
%mend;
```

V. Application Example II: Archive Out-dated Files.

If a file is older than a given duration(in months), it will be moved to a specific Archive folder by a SAS macro.

```
%macro Archive(path=, pid=, duration=);
options mprint symbolgen ;

Filename Files PIPE %unquote(%str(%') dir
"&path.\&pid.\Reports" /b %str(%'));

data List;
infile Files lrecl=256 truncover;
input  Full_Name $256.;
Report_Name=strip(scan(Full_Name, 1, "."));
Suffix=strip(scan(Full_Name, 2, "."));
N=length(Report_Name);
MTH_Year=strip(substr(Report_Name, N-7));
DD_MTH_Year="15"||substr(MTH_YEAR, 1,
3)||substr(MTH_YEAR, 5, 4);
Age=INTCK("Month", input(DD_MTH_Year, date9.),
today());
run;

data Move;
length Old_Name New_Name $100;
set List;
if Age > &Duration then do;
Old_Name="&path.\&pid.\Reports\"||strip(Full_N
ame);
New_Name="&path.\&pid.\Archive\"||strip(Full_N
ame);
RC=rename(Old_Name, New_Name, 'file');
if RC=0 then put "Note: Files moved to Archive
folder: " Old_Name ;
else put "Note: Files cannot be moved: "
Old_Name  ;
end;
run;
%mend;
```

SAS GLOBAL FORUM

Conclusion

Managing external files and directories plays an important part in data analysis and business analytics.

A good file management can help to streamline project management and improve work efficiency.

It will be both efficient and beneficial to automate and standardize the file management processes by using SAS programming.

Acknowledgement

- Special thanks to TransUnion Canada and SAS Canada for the financial support.
- Thank you so much!

SAS® GLOBAL FORUM

April 26-29
Dallas, TX