SAS® GLOBAL FORUM 2020 #SASGF

Paper 4821-2020

Organize and Manage Files by SAS® Programming

Weibo Wang, TELUS Communications; Heyang Li, Montclair State University

ABSTRACT

In daily work, we often save different types of files on physical servers or hard disks. Statistics show that 7.5 percent of an organization's documents are lost entirely, while another 3 percent are misfiled. As time goes by, the servers or hard disks would get cluttered. We may want to regularly organize, archive or delete some old files to release space. Many applications can do these, SAS® software being one of them.

SAS is not only an analytics, business intelligence and data management software, but also a powerful tool that can make your housekeeping work more efficient. Together with SAS® Management Console Scheduler or UNIX crontab, you can set up automated process to regularly scan your physical server or hard disk. The process can send you a report of all the files in a directory. It can delete files of a size over a threshold or having not been modified for some days. It can remove duplicate files. It can archive files. It can split and merge files. In a nutshell, it can do everything that a file management software can do.

INTRODUCTION

This paragraph uses When people say using SAS® software to manage your disk space, using SQL delete, Drop table, Delete procedure or Proc Datasets to remove dataset may at once come to mind. However, there are some functions in SAS® that give you immense potentials. They are X command, CALL SYSTEM ('command') and %SYSEXEC command. These methods enter operating environment mode and enable you to submit a Windows command or a UNIX command without ending your SAS® session. Then not only can you delete SAS® dataset, you may also copy, move, delete, and zip all kinds of files.

Some other useful functions are DOPEN, DREAD, DCREATE and DLCREATEDIR. DOPEN opens a directory. DREAD retrieves file size, creation time, and last modified date of an external file. DCREATE and DLCREATEDIR can create files and directories. With these functions, operation on files can be deliberate and precise. We can:

1. Cull legacy, big files.

2. Organize files by certain categories.

3. Copy, rename and move files.

4. Zip files.

5. Create a metadata index for all files.

6. Search files contain certain string.

For example, you can delete files that are over 1G and haven't been modified by 24 months; you can move all files that have 'Upsell' in file name to the Upsell folder; you can archive all the files with '2019Q3' in file name and having '.csv' as suffix; you can rename files following certain naming convention; you can organize files by date, by owner, by portfolio; you can create an index file for all files; you can search the logs contains string 'Error', etc.. This is an advantage of SAS® software over other file management software.

After you know what you want to do with those files, you can automate the session with SAS® Scheduler or UNIX crontab. In this paper, the authors will present some examples of how they use SAS to manage their disk space.

PART 1: READ IN FILE NAMES

Usually there are many sub-folders under one directory. With the following X command, we can extract all the folder and subfolder names under one directory, creating a temporary text file that contains all the folder names for SAS to read. X command can be used to delete this temporary file afterwards. CALL SYSTEM ('command') and %SYSEXEC command can do the same but this paper will only focus on X command as there’re plenty of examples about how to use the other commands.

```
*******************For UNIX Server ****************************************;
**************** Grab all subfolders from UNIX server. ********************;
x 'find /home/yourfolder -type d -ls >/home/sasdata/path.txt';
```

Usually Then we can input file names into dataset for further manipulation.

```
data pathname;
  infile '/home/user/path.txt' truncover;
  input   name $200.;
   start=index(name,'/');
   pname=substr(name,start);
run;
******************** For Windows ******************************************;
************ Grab all subfolders on your windows hard disk.****************;
x 'dir c:\yourfolder /s /b /o:n /a:d >c:\path.txt';
data pathname;
  infile 'c:\path.txt' truncover;
  input   name $200.;
run;
```

The following part of the paper will only focus on the UNIX side. Readers can test on Windows system if they are interested.

At the end of your program, you can delete the temporary file:

```
**************** Delete the temporary file.********************************;
x "rm &yourfolder./path.txt";
```

After inputting all the folder paths, we can use the following macro to extract the attributes of all the files. The end part of the macro will append all the datasets into one dataset “list_of_all_files”. With this dataset, we can find duplicate files, oversized files, old files or files with specific suffix for further actions.

```
%macro scanfolder;
  proc sql noprint;
    select count(*) into : folder_counts from pathname;
  quit;
  %let n=&folder_counts;
  proc sql noprint;
    select pname into : PName1 -: PName&n from pathname;
  quit;
    %do i= 1 %to &n;
    %let dir = %unquote(&&PName&i);
  data scan&i (drop = _:);
   format foldername $200.;
    _rc = filename("dRef", "&dir.");
    _id = dopen("dRef");
    _n = dnum(_id);
      do _i = 1 to _n;
        name = dread(_id, _i);
        _rc = filename("fRef", "&dir./" || strip(name));
        _fid = fopen("fRef");
        foldername="&dir";
        size = finfo(_fid, "File Size (bytes)");
        dateModify = finfo(_fid, "Last Modified");
        _rc = fclose(_fid);
        output;
      end;
    _rc = dclose(_id);
  run;
  %end;
  data list_of_all_files;
  set scan1
   %do j=2 %to &n;
   scan&j %end;;
  run;
%mend scanfolder;
%scanfolder
```

PART 2: SOME EXAMPLES

When you have the final report, you can take different actions tailored to various goals.

DELETE OLD FILES

You may want to delete some big and old files, let’s say some SAS datasets that are over 1G and haven’t been modified for 60 months. After retrieving those from the report dataset, you can use the following macro to delete them.

```
data files_to_delete;
set list_of_all_files;
  format fname $500.;
  fname= strip(foldername)||'/'||strip(name);
                   /* This step is to retrieve file names with the path. */
  where size >=  1024*1024*1024
                                          /* The file size unit is Byte. */
  and dateModify <= today()-60*30
  and index(name, ‘sas7bdat’) ;
run;
%macro deletefile;
  proc sql;
    select count(*) into : k from files_to_delete;
  quit;
  %let k=&k;
  proc sql noprint;
    select fname into: FName1 -: FName&k from files_to_delete;
  quit;
      %do l=1 %to &k;
      x "rm &&FName&l.";
      %end;
%mend deletefile;
%deletefile
```

SEND SUMMARY OF FILES

You may want to send yourself a detailed report of the files in the directory. The following code is being used by the authors to that end.

```
options emailsys=smtp emailhost="&Emailhost" EMAILID="&email"
emailport=="&email_port";
filename mail email to=("&email") subject= "File Report"
content_type="text/html";
data _null_;
      file mail;
ods _all_ close;
ods ESCAPECHAR='^';
ods html body=mail style=minimal;
      put '<html>';
      put '<head>';
      put '<meta content="text/html; charset=ISO-8859-1" http-
equiv="content-type">';
title "Report Date: &sysdate.";
proc report data=final_report nowd headskip headline missing
formchar(2)='_' spacing=2 split='~';
      column name size LastModified foldername;
      define name / "File Name" width=20 left flow;
      define size / "File Size(B)" width=20 left flow;
      define LastModified / "File Modify Date" left flow;
      define foldername / "File Location" width=20 left;
run;
options missing='-';
title;
footnote;
ods html close;
ods _all_ close;
run;
```

You can add several additional conditions that only show duplicate files, only show files generated in a certain time period or only show files of a specific portfolio to customize your report.

ARCHIVE FILES

If you want to zip and archive your input files, SAS code or other files periodically, (for example, you would like to zip all the csv report files from the third Quarter,) you can refer to the following code written by the authors.

```
data files_to_zip;
set list_of_all_files;
      format fname $500.;
      zname= strip(foldername)||'/'||strip(name);
                  /* This step is to retrieve file names with the path. */
      where index(name, ‘Q3’)
      and index(name, ‘.csv’) ;
run;

%macro ZipFile;
%local files_to_zip zip_command zipfile;
data _null_;
      call symput('zipfile',"/home/archive/Q3._data.zip");
run;
      %let files_to_zip=N;
      %let zip_command=zip -Dqjm &zipfile;
%macro append_zip_command(zFile);
      %if %length(&zFile)
```

```
        %then
        %do;
              %let zip_command=&zip_command &zFile;
              %put &zFile to be zipped;
              %let files_to_zip=Y;
        %end;
  %mend;
  proc sql;
  select count(*) into : p from files_to_zip;
  quit;
        %let p=&p;
  proc sql noprint;
  select zname into: FName1 -: ZName&p from files_to_zip;
  quit;
        %do q=1 %to &p;
              %append_zip_command(&&ZName&q)
        %end;

        %if &files_to_zip=Y
        %then
        %do;
              systask command "&zip_command" wait;
        %end;
        %else
        %put No files zipped;
  %mend;
  %ZipFile;
```

OTHERS

These are only several examples that can efficiently manage your disk space. You can investigate more SAS functions and conference papers for different approaches and solutions.

CONCLUSION

The authors have an automated program to scan Enterprise server, send report to team and delete big, legacy files to make room for new files. This procedure makes sure that no other automated programs fail because of no room to keep the output files. For some companies that are inputting external files to manipulate data, this procedure can also help them to archive and save those external files automatically.

REFERENCES

1 SAS Support Website,

http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000209538.htm

2 SAS Support Website,

http://support.sas.com/kb/40/934.html

3 SAS Support Website,

http://support.sas.com/documentation/cdl/en/hostwin/63285/HTML/default/viewer.htm#win-stmt-x.htm

4 SAS Support Website,

http://support.sas.com/documentation/cdl/en/hostunx/61879/HTML/default/viewer.htm#xcomm.htm

5 Jia, J., & Lin, A. (2015). Yes, SAS® can do! ---Manage external files with SAS programming [Paper no. 32622015]. In Proceedings of the SAS Global Forum, Dallas TX.

http://support.sas.com/resources/papers/proceedings15/3262-2015.pdf