PharmaSUG 2025 - Paper SD-371

SAS Program for Backup Zipping

Tong Zhao, LLX Solutions, LLC

ABSTRACT

In everyday work, after completing a study or identifying outdated documents and datasets in folders, we often create zip archives to save memory and space. However, manually zipping dozens of folders individually—while retaining their original names—can be time-consuming. This SAS program automates the process by replacing folders with identically named zip files in the directory, all executed through a single batch run. This paper demonstrates the program's usage and provides a step-by-step explanation of the code. Additionally, it highlights how to integrate command-line tools into SAS, serving as a brief tutorial for those interested in combining external software with SAS programming.

INTRODUCTION

In daily work, we often manage dozens of folders and files. As projects progress, archiving outdated documents and datasets into compressed files becomes essential for saving storage space. While this can be done manually, automating the process significantly improves efficiency. This paper presents a program designed to automatically zip multiple folders with a single click, streamlining the backup process and saving considerable time and effort.

The program leverages the 7-zip (Pavlov, 2024) application for compression. Typically, 7-zip can be operated either manually (via graphical interface) or through command-line prompts. While the command prompt allows sequential execution of commands, it lacks flexibility for batch editing or simultaneous command modifications. To address this limitation, the solution uses SAS to generate and execute 7-zip commands programmatically. This approach enables easy editing of commands within the SAS environment and batch execution of multiple commands with a single click.

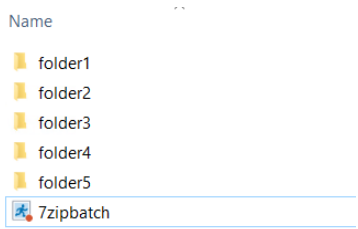
This paper details the SAS program's implementation for automated backup zipping and serves as a practical example of integrating external software (e.g., 7-zip) with SAS for enhanced functionality.

APPLICATION AND USAGE

PREPARATION

The program *7zipbatch* requires 7-Zip (Pavlov, 2024) software to be installed on the host system. 7-Zip is an open-source compression utility available for download online. While this implementation utilizes 7-Zip specifically, the program architecture supports integration with alternative compression software through minor code modifications.

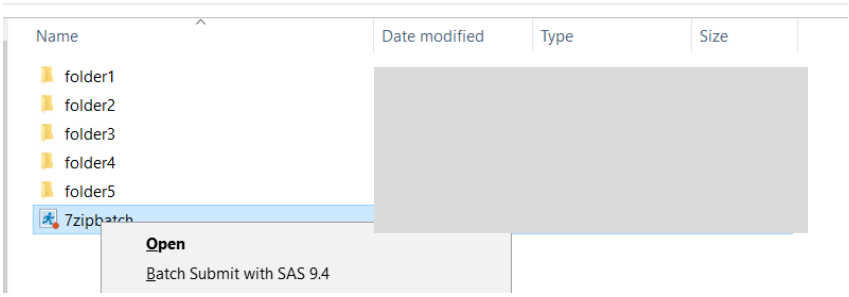
To execute the program, place the SAS program file *7zipbatch* in the parent directory containing the target folders.



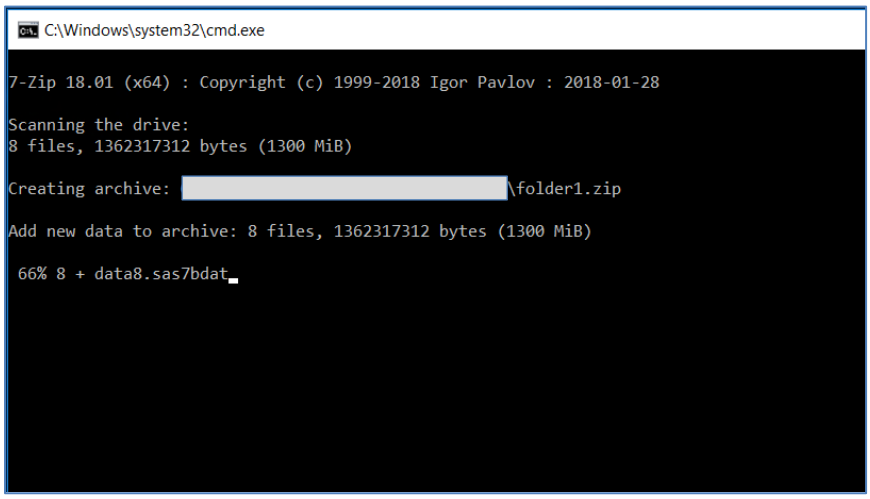
Display 1. Preparation

RUN THE PROGRAM

Batch run the program file *7zipbatch*. A command line interface window will open, and real-time progress will be displayed in the command window. The window will close automatically upon process completion.



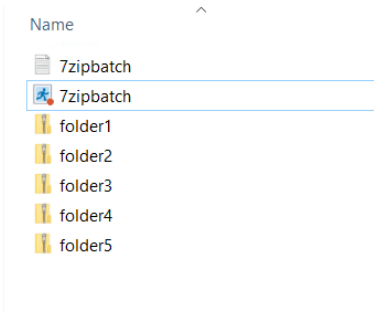
Display 2. Run the program



Display 3. Command prompt window showing the progress

OUTPUT OF THE PROGRAM

Upon successful execution, the program *7zipbatch* will generate compressed archives (.zip format) for each target folder and automatically remove the original uncompressed folders, while retaining the original directory structure, file names and file paths.



Display 4. Output of the program

PROGRAM DESCRIPTION

SET DIRECTORY

The program offers two flexible methods for directory specification. The program will automatically detect and use its current working directory. This is the default configuration requiring no user intervention, and it is ideal for standard use cases where the program is placed in the target folder. Users can also explicitly define a custom directory path within the code which provides precise control over the target location.

```
*Set directory;
/*Automatically detect the current directory*/
data _null_;
   sys = symget('sysprocessname');
   if sys = 'DMS Process' or getoption('DMS') = 'DMS' then path =
sysget('SAS_EXECFILEPATH');
   else if sys =: 'Program' then path = getoption('SYSIN');
   srcdir = substr(path,1,findC(path,'\',-length(path))-1);
   call symputx('dir',strip(srcdir),'G');
run;

/*Manually set the directory*/
*%let dir = Z:\xxx\Data\Raw;

%put &dir;
```

Program 1. Set directory

GET FOLDER LIST

The program automatically compiles a complete list of all subdirectories. The inventory is stored in the *dirlist* data set as shown in Display 5. The *dirlist* data set may be modified to include/exclude specific folders or to prioritize processing order.

```
*Get subfolder list;
filename DIRLIST pipe "dir ""&dir"" /a:d /b";

data dirlist;
   infile dirlist length=reclen;
   input buffer $varying200. reclen;
run;

data dirlist;
set dirlist;
   if buffer = 'xxx' then delete; *Modify as needed;
run;
```

Program 2. Get folder list

VIEWTABLE: Work.Dirlist

	buffer
1	folder1
2	folder2
3	folder3
4	folder4
5	folder5

Display 5. Folder list in data set dirlist

CREATE CODE LIST

The below program generates command-line instructions in output data set *code*.

```
*Create code list;
data code;
set dirlist;
length code1 code2 code $1000.;
code1 = cat("7z.exe a -w '&dir\",strip(buffer),"' -tzip '&dir\",strip(buffer),"\*'");
code1 = strip(tranwrd(code1, "'", '"'));
code2 = cat("RD /S /Q '&dir\", strip(buffer), "'");
code2 = strip(tranwrd(code2, "'", '"'));
code = catx(' && ',code1, code2);
run;
```

Program 3. Create code list

VIEWTABLE: Work.Code

	buffer	code1	code2	code
1	folder1	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder1" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder1*"	RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder1"	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder1" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder1*" && RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder1"
2	folder2	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder2" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder2*"	RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder2"	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder2" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder2*" && RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder2"
3	folder3	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder3" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder3*"	RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder3"	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder3" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder3*" && RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder3"
4	folder4	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder4" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder4*"	RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder4"	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder4" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder4*" && RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder4"
5	folder5	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder5" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder5*"	RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder5"	7z.exe a -w "C:\Users\tong.zhao\Desktop\code\New folder\folder5" -tzip "C:\Users\tong.zhao\Desktop\code\New folder\folder5*" && RD /S /Q "C:\Users\tong.zhao\Desktop\code\New folder\folder5"

Display 6. Commands in data set code

CODE SYNTAX

Commands follow the syntax as shown below.

```
7z.exe a -w [input_folder] -tzip [output.zip] && RD /S /Q [input_folder]
```

Program 4. Code syntax

The `7z.exe` statement creates the zip files. Key parameters:

- `a` - Add to archive
- `-w` - Set the working dictionary
- `-tzip` - Specify ZIP format

The `RD` statement removes the original folders. Key parameters:

- `/S` - Remove all subdirectories and files
- `/Q` - Quiet mode (no confirmation prompts)

The `&&` operator ensures sequential execution (compression completes before deletion).

EXECUTE COMMANDS

The program below executes system commands directly from SAS, leveraging SAS's `call system ()` function and `x` command.

```
*Execute commands;
options noxwait;
data _null_;
set code;
   do;
      x "cd C:\Program Files\7-Zip"; *<-- this is to use 7zip.exe;
      call system(code); *<-- this is to execute the commands stored in variable code;
   end;
run;
```

Program 3. Execute commands

CONCLUSION

The presented SAS program automates folder compression and archival through seamless integration with 7-Zip's command-line interface. The solution combines SAS's data manipulation strengths with system-level file operations for efficient bulk processing.

REFERENCES

Igor Pavlov. 2024. "7-Zip." https://www.7-zip.org

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tong Zhao
LLX Solutions, LLC
tong.zhao@llxsolutions.com