PharmaSUG 2025 - Paper SD-228

A SAS® System 7-zip macro that creates a zip archive and a file archive macro: versioning in the context of a private library of programs

Kevin R. Viel, Ph.D., Histonis LLC and Navitas Data Sciences

ABSTRACT

Versioning of files is essential in any project, but especially in a regulated industry, when the project has multiple resources (programmers, project managers, biostatisticians, et cetera), and in a fast-paced environment. When a programmer obtains a program from another project/delivery or in the midst of a complicated update with changing specifications, like saving a file while working on it, keeping a version of an updating file, a snapshot, is wise and helpful. While mature, well-developed versioning platforms are freely available, they may not be approved for use and have a learning curve. The **goals** of this paper is to describe a SAS® System macro that uses 7-zip, a file archiver with a high compression ratio (https://www.7-zip.org/), a macro based on it to create a zip archive and its corresponding table of contents file (CSV), and macro to demonstrate the creation of an archive. The advantage of 7-zip over the ZIP engine to the FILENAME statement is that the datetime of the target file is preserved including after extraction . One use would be for a programmer to create a local archive and, thus, a personal library of files (programs) including their statuses (final or ongoing) for a given folder (project) and annotations such as the (first) use of a macro, technique, or derivation, while providing the ability to quickly and easily find a candidate program to adopt to a new delivery or project or to roll back to an earlier version while keeping other versions of the file..

INTRODUCTION

While working on a file of importance, one frequently saves the file to protect updates against some calamity. Sometimes, we learned the costs of saving the file too infrequently through a loss of power before we were able to save the file. One can adopt a manual process that allows one to revert to an earlier version if one decides against the changes. Of course, manual approaches such as appending a preferred suffix to a file name are quite simple, but may clutter one's working directory. In shared directories or with shared files, the impact of an inappropriate update or deletion of a file can be quite severe. In such cases, a programmer or manager may wish to create and maintain a "private" library of files, an archive, with at least rudimentary versioning being optional. The **goals** of this paper are 1) to describe a SAS® System macro that uses 7-Zip, a file archiver with a high compression ratio (https://www.7-zip.org/), 2) to describe a second macro based on the first that creates a zip archive and its corresponding table of contents (ToC) file in comma-separated value (CSV) format, and 3) to describe a third macro to demonstrate the creation of an archive. A fourth macro is included for learning and exploratory purposes.

POL/SOP/WI

The author advises the reader to verify any potential process with respect to the Policies (POL), Standard Operating Procedures (SOP), and Working Instructions(WI) that govern the work of the reader. The author is not advising the reader on the ability to use any of these methods in a given environment, like at work. The author cautions that even for a file for which the user is the Owner, that archiving it on another drive, like one's "personal drive" in Windows, for example:

```
C:\Users\kviel\Documents
```

may violate policies of one's employer. If this drive is backed up (automatically), such as OneDrive in Microsoft Windows®:

```
C:\Users\kviel\OneDrive\Documents
```

then the issue may be even more profound since this cloud-based storage is accessible outside of one's work environment. This issues may also apply when one manual copies a files (copy-and-paste).

Verification with one's manager and/or IT department that using any method or code in this paper is consistent with applicable POL/SOP/WI is highly advisable, especially *in writing*.

ZIP

ZIP is a file format that compresses an original file with the resulting file often called an "archived version". Chris Hemedinger wrote three excellent blog posts[1,2,3] about using the ZIP engine to the FILENAME statement in SAS. This paper does not use this approach noting the response[1] by Mr. Hemedinger to a question:

> September 30, 2016 2:09 pm
>
> This process rewrites the file and so changes the file date/time information.

Since it preserves the Last Modified datetime stamp, the author has chosen the third party, freely available **7-Zip**[4] program. If a shop decides to not install 7-Zip, then the ZIP engine to the FILENAME statement in SAS is an excellent substitute.

MAJOR/MINOR UPDATES

This paper is **not** intended to represent a Version Control System. Either the version number or the datetime stamp will be incorporated into the name of the file in the zip archive. In this paper, a MAJOR update results in an increment of the version number to the next higher whole number (1.00 will increment to 2.00 and 3.02 will increment to 4.00). A MINOR update results in an increment of the decimal portion by 0.01 (it is assumed that 100 minor updates will not occur). A version number of 0 in the ToC indicates a "work in progress" distinguished by the datetime stamp incorporated into the name. The initial version number is 1.00. In the use of these macros and programs, the *user* deems what qualifies as MAJOR or MINOR update to a file simply by selecting a value of a macro parameter. The impetus for adding version was SAS AutoCall macros, whose name must be the name of the (.sas) file, but other file names such as adsl.sas or dm.r also do typically not reflect their versions.

MAC_U_7ZIP MACRO

Appendix 1 presents the macro MAC_U_7ZIP**. Log 1** shows the help section printed to the SAS Log when HELP = Y with a brief description of the macro parameters and their default values, if any.

```
Purpose of program:     This utility macro creates or interacts with a zip archive using 7zip.

Macro Parameter         Description
_______________         ____________________________________________________________________
command               = Corresponds to the command for the command line of 7zip.
                          Add - a; Extract - e;
                          List - l; Rename - rn;
                          Delete - d
archive_path          = Path of zip archive
archive_name          = Name of zip archive
path                  = Path of file to archive
                            Default: %str()
file                  = Name of file to archive (include extension or wildcard)
                            Default: %str()
pathfile              = The path and filename of a single file of interest (to archive).
                            Default: %str()
path_file_ds          = Name of data set with paths/files
extract               = x (Extract with full paths) command (Extracts files from an archive
                          with their full paths in the current directory, or in an output
                          directory if specified.
                        e (Extract) command (Extracts files from an archive to the current
                          directory or to the output directory.)
```

```
                          Default: x
extract_o             = The directory corresponding to the switch -0{Directory} : set Output
                        directory
                          Default: %str()
extract_c             = Extract criteria.
                          For instance: Version_001_000/adsl.sas
                                        *.sas -r
                          Default: %str()
original_name         = Name of file in the archive to change.
                          Default: %str()
rename                = New name of file
                          Default: %str()
list_ds               = Output data set for the parsed output from the 7zip listing of the
                        archive.
                          Default: _7z_list
passwd                = password for the -p switch
                          Default: %str()
switch                = Additional switches appending to the end of the command line
                          Example:  -y (Assume Yes on all queries) switch
                          Default: %str()

_____________________________________________________________________________________________

End of help
```

Log 1. The Help section of MAC_U_7ZIP.

The major focus of this macro is to write the command line analogous to what one would write "by hand" on the command line (**Figure 1**). Note that, in Windows, one can use the "Progra~1" for "Program Files" in the path and since no other folder has spaces, "C:\Progra~1\7-Zip\7z.exe -h" without the double quotation marks achieves the same result as that in Figure 1 in the Windows CMD. This is one aspect that the reader may need to change in the provided code, especially depending on environmental variables like "path". Also note that Windows is case insensitive.

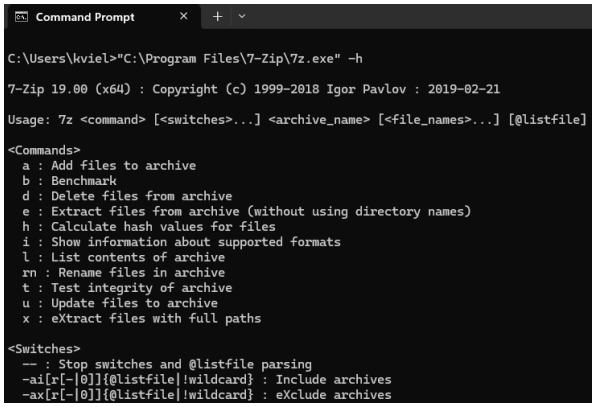


Figure 1. A partial display of the results of a call to 7Zip on Windows using the command line.

MAC_U_ FILE_ARCHIVE MACRO

Appendix 2 presents the macro MAC_U_FILE_ARCHIVE. **Log 2** shows the help section printed to the SAS Log when HELP = Y. To provide a brief summary, if the ToC exists, then the macro will create it, otherwise, the macro will read it. If the zip archive exists, then the macro will create a list of its contents, among later actions. Otherwise, the macro will create it. The macro will obtain the datetime stamps of the files using the macro MAC_U_FINFO[5], this macro is based on the SAS FINFO() function, which is also used in the SAS macro MAC_U_DIRECTORY_READ[6]. The macro checks whether the current file is already in the archive with the type (FINAL). If so or if the file of interest is somehow younger than the file

in the archive, then the macro notes this in the log and does not further act on the archive or its ToC. The macro creates the "path" by omitting the colon (":") from the drive name:

This becomes a "folder" conceptually. After the macro adds the file to the archive, it renames it, thereby keeping distinct "versions" with the same "base" name.

```
Purpose of program:      This utility macro uses 7zip to work with a zip archive, for instance
                         to create a new zip archive and add a file to it, to rename the file
                         for archival purposes, to list the contents of the zip archive, or to
                         delete a file from the archive.  This macro also creates, modifies, or
                          adds to a Table of Contents (CSV) corresponding to the zip archive.

                          This macro has two modes: FINAL = Y and FINAL = N.

                          For FINAL = Y, if the file of interest is younger than or the same age
                          as the file in
                          the archive, then it will not be replaced.

                          For FINAL = N, if the file of interest is not same age as the file in
                          the archive, then the file is added to the archive, then the name is
                          changed by appending the datetime to the filename.

                          This macro requires 7zip: https://www.7-zip.org/. The reason this
                          program was chosen over the SAS ZIP engine for the FILENAME statement
                          is that 7zip preserves the datetime of the archived then extracted
                          file. See: https://blogs.sas.com/content/sasdummy/2015/05/11/using-
                          filename-zip-to-unzip-and-read-data-files-in-sas/

Macro Parameter          Description
_________________        ____________________________________________________________________
archive_path          = The path to the zip archive of interest.
                           Default: %str()
archive_name          = The name of the zip archive of interest.
                           Default: %str()
pathfile              = The path and filename, with extention, of the file of interest.
                           Default: %str()
final                 = Whether the file of interest is the final version.  "Final" may mean
                          Validated and approved, but not the latest version.  FINAL is of
                          interest so that one may search an archive for "production" programs.
                           Default: N
library_csv           = The path and filename of the .csv file that may act as the table of
                          contents of the archive.
                           Default: %str()
_____________________________________________________________________________________________
End of help
```

Log 2. The Help section of MAC_U_ FILE_ARCHIVE.

MAC_ PGM_ARCHIVE MACRO

Appendix 3 presents the macro MAC_PGM_ARCHIVE. This is just one approach to using the macros to create a personal archive that accepts a data set of path and file names of files of interest. If that input data set includes the variable FINAL, then it controls that value of the macro parameter FINAL file-by-file. Otherwise, the macro parameter FINAL to MAC_PGM_ARCHIVE applies to each file (globally). **Appendix 4** present a macro for testing in this paper. It list the contents of the archive and the ToC for comparison purposes and is intended only for exploratory, not production (formal), purposes.

EXAMPLE

In the Windows system, we will create a text file and replicate (copy and paste) it in the same directory (**Figure 1**):

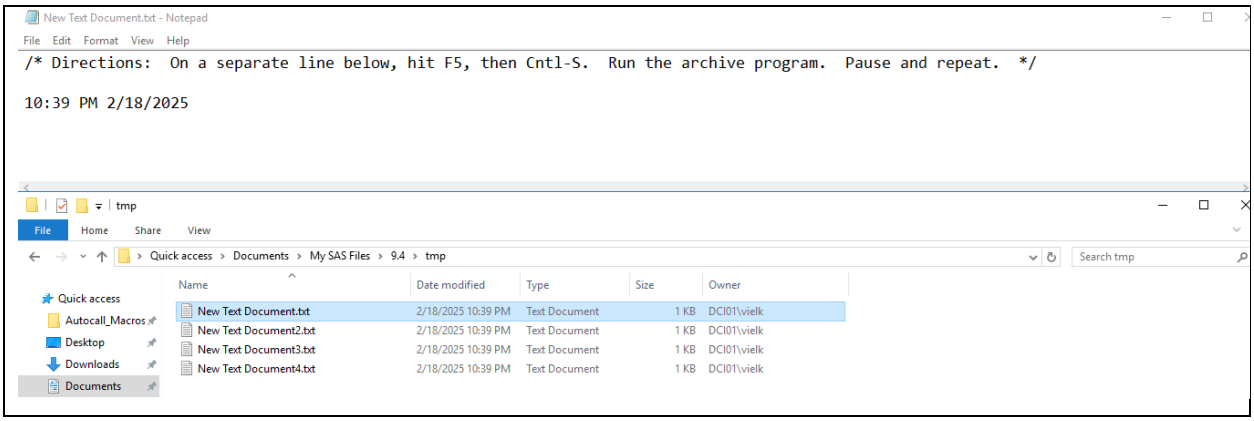


Figure 1. The initial files.

Note that in Notepad, pressing F5 inserts the time and date, for example:

12:14 PM 3/29/2025

In Windows File Explorer, copying and pasting a file does not change its Date Modified (Last Modified datetime stamp). This creates unusual metadata, however, with a “Modified” datetime stamp *BEFORE* the “Created” datetime stamp (**Figure 2**, right panel).

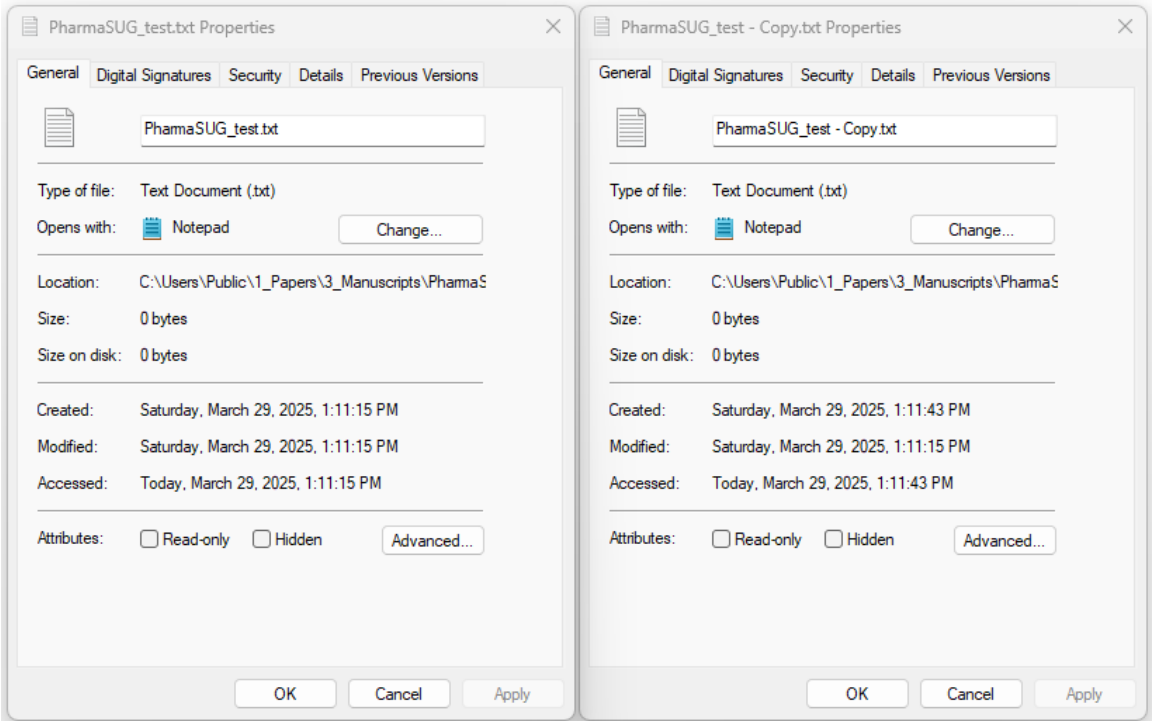


Figure 2. An example of quizzical metadata in Windows.

The screenshots and output that follow pertain to the code in **Program 1**.

```
 1 data files_archive
 2         ( drop   = final
 3                    version_type
 4         )
 5      files_archive2
 6      ;
 7
 8   length pathfile     $ 200
 9          final        $   1
10          version_type $   5
11          ;
12
13   input pathfile &&
14         final
15         version_type
16         ;
17
18   pathfile = scan( pathfile , 1 , '"' ) ;
19
20 datalines ;
21 "C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document.txt"   Y major
22 "C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document2.txt"  N major
23 "C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document3.txt"  Y minor
24 "C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document4.txt"  N major
25 ;
26 run ;
```

Program 1. Test code to demonstrate an archival process.

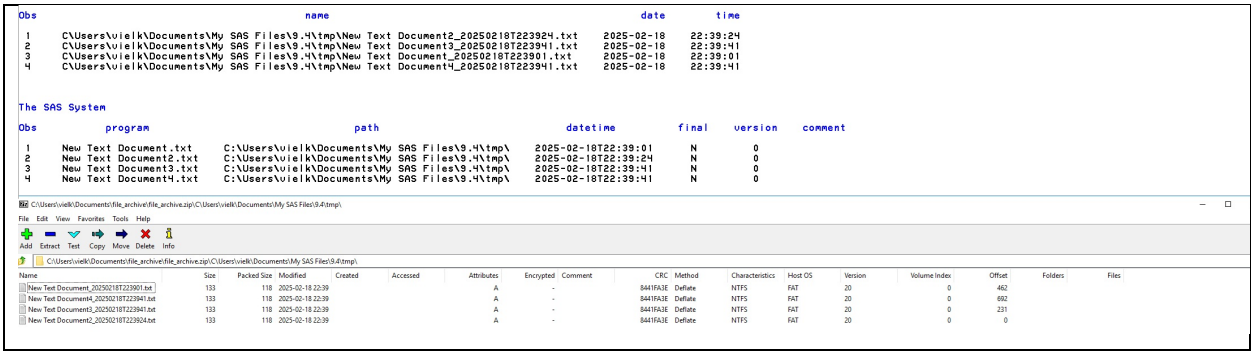
ITERATION 1 – INITIAL ARCHIVE CREATION

The initial submitted code uses the macro parameter FINAL to assign the type of archive:

```
%mac_pgm_archive
   ( final          = N ) ;

%mac_pgm_list_7z_csv ;
```

In this case, the update is neither MAJOR or MINOR, which pertains to a final version, so the value of VERSION in the ToC is 0. Instead, in the archive, the Last Modified datetime is included in the name, not the version. **Figure 3** shows the results.



Obs	name	date	time
1	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document2_20250218T223924.txt	2025-02-18	22:39:24
2	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document3_20250218T223941.txt	2025-02-18	22:39:41
3	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document_20250218T223901.txt	2025-02-18	22:39:01
4	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document4_20250218T223941.txt	2025-02-18	22:39:41

The SAS System

Obs	program	path	datetime	final	version	comment
1	New Text Document.txt	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\	2025-02-18T22:39:01	N	0	
2	New Text Document2.txt	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\	2025-02-18T22:39:24	N	0	
3	New Text Document3.txt	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\	2025-02-18T22:39:41	N	0	
4	New Text Document4.txt	C:\Users\vielk\Documents\My SAS Files\9.4\tmp\	2025-02-18T22:39:41	N	0	

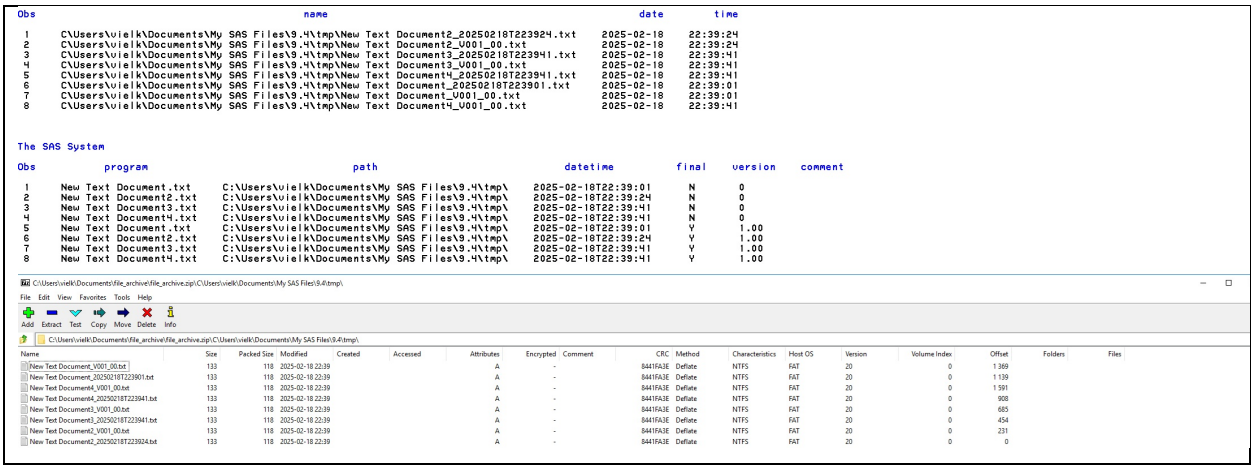
Figure 3. The initial creation of an archive with files that are not final.

ITERATION 2 – UPDATE WITH MINOR FINAL VERSIONS

From this point on, the call to the test macro MAC_PGM_LIST_7z_CSV will not be shown in the example code. The macro parameters FINAL and VERSION_TYPE are used to apply these types to every file. For a given call, this is likely the approach, that is, one is likely to only include interim or final updates.

```
%mac_pgm_archive
   ( final          = Y
   , version_type   = minor
   ) ;
```

Figure 4 shows the results of a first update of the archive after it was initially created. Note that in the case of the initial FINAL versions, the values of VERSION are 1.00, that is, VERSION_TYPE (MINOR or MAJOR) is irrelevant. The archive retains the distinct non-final versions of the files initially created with their datetime stamps and now includes the “final” versions. Deletion of such version will require the programmer to interact with the archive either programmatically (using MAC_U_7ZIP) or manually using 7-Zip in interactive mode, that is, with its GUI. Using the macro potentially provides an audit trail (of logs and/or programs). Note that the combined DATE and TIME for a given “base” name are not necessarily distinct.



Obs name date time

1 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document2_20250218T223924.txt 2025-02-18 22:39:24
2 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document2_V001_00.txt 2025-02-18 22:39:24
3 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document3_20250218T223941.txt 2025-02-18 22:39:41
4 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document3_V001_00.txt 2025-02-18 22:39:41
5 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document4_20250218T223941.txt 2025-02-18 22:39:41
6 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document_20250218T223901.txt 2025-02-18 22:39:01
7 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document_V001_00.txt 2025-02-18 22:39:01
8 C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document4_V001_00.txt 2025-02-18 22:39:41

The SAS System

Obs program path datetime final version comment

1 New Text Document.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:01 N 0
2 New Text Document2.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:24 N 0
3 New Text Document3.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:41 N 0
4 New Text Document4.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:41 N 0
5 New Text Document.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:01 Y 1.00
6 New Text Document2.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:24 Y 1.00
7 New Text Document3.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:41 Y 1.00
8 New Text Document4.txt C:\Users\vielk\Documents\My SAS Files\9.4\tmp\ 2025-02-18T22:39:41 Y 1.00

Figure 4. The first update of the archive with files that are final.

ITERATION 3 – UPDATE 2 WITH MAJOR FINAL VERSIONS

```
%mac_pgm_archive
   ( final          = Y
   , version_type   = major
   ) ;
```

Figure 5 shows multiple versions of the same base file. In some cases, those might be versions 1.00 and 2.00 or multiple “development” versions with distinct datetime stamps incorporated into the name.

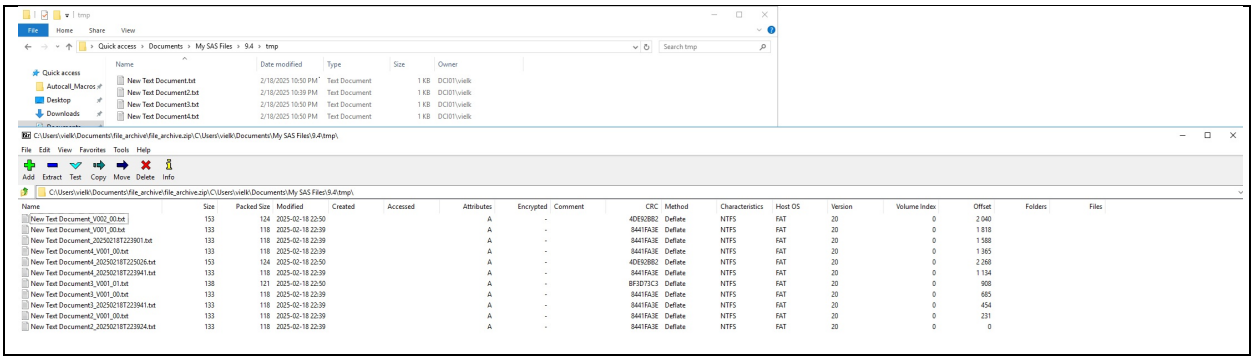


Figure 5. The next update of the archive with files that are final.

Note that not every file has been updated. This approach relies on the datetime stamps. While certain combinations will result in an addition to the archive and its ToC, in other conditions, no additions will occur, including if the some given code was submit again inadvertently.

COLLISIONS

Especially, when working with numerous files or when burning the midnight oil, a user might attempt to add a file with a given FINAL status multiple times. The user may just have forgotten the she or he already added the file or accidentally submitted the code twice. In this case, the macro simply notes the “collision” and performs no further actions:

```
WARNING:  The target file is not older than the version in ~\file_archive\file_archive.zip
WARNING:  FINAL:          Y
WARNING:  VERSION_TYPE:   major
WARNING:  Target:         C:\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document.txt
WARNING:  Archive:        C\Users\vielk\Documents\My SAS Files\9.4\tmp\New Text Document_V001_00.txt
WARNING:  Target:         2025-02-18T22:39:01
WARNING:  Archive:        2025-02-18T22:39:01
```

Note “conversions” that have occurred to the Archive version (the change in the Drive name and the file name).

COMMENTS

The ToC has a free text field, COMMENT, in which the user can annotate or otherwise add data of interest. This text remains untouched by the current macro and programs, so it will persist. The author has used this field to indicate the program (or version thereof) that used a new algorithm, such as a windowing method or imputation of a date with missing components, new code, such as a SAS GTL plot or a new macro a lab conversion macro[7] with project-specific units. The most secure way to update this field might be to use a program. Editing a CSV (text) file in a typical editor like NOTEPAD can be problematic. Using Microsoft Excel can result in undesired conversions or alterations to a cell, for instance, and requires that one remember to save the file in the CSV format.

CONCLUSION

This paper contributes to the methods of creating a personal or private archive of important files, some of which may be in development and, thus, are being updated frequently with high resource utilization. The archive of choice is a zip file because it preserves datetime stamps of the original files and saves space through compression. The particular software, 7-Zip, can also create password control, but that may not be nearly sufficient and may be more of a safety, rather than security, mechanism. This paper also contributes to the discussion concerning version types (Major/Minor), without representing a Version Control System. In this case, the VERSION is a sequence and status indicator. In addition to allowing a programmer to revert to a given pervious file (“version”) in an informal situation, like a program in the development environment that is not yet part of a production batch or delivery, the archive and ToC can also provide metrics of interest and can be used to quickly identify candidate programs that used a given technique (algorithm or definition, for instance, an alternative windowing definition or imputation of missing date/time components). This suite of macros and programs should provide an informal level of safety and a source of personal files to adopt to or reference for future deliveries or different projects.

REFERENCES

[1] Hemedinger, C. “Using FILENAME ZIP to unzip and read data files in SAS.” Available at https://blogs.sas.com/content/sasdummy/2015/05/11/using-filename-zip-to-unzip-and-read-data-files-in-sas/. Last modified May 11, 2015. Accessed March 29, 2025.

[2] Hemedinger, C. “List the contents of your ZIP files using SAS.” Available at https://blogs.sas.com/content/sasdummy/2016/10/16/filename-zip-list-file-contents/. Last modified October 16, 2016. Accessed March 29, 2025.

[3] Hemedinger, C. “Using FILENAME ZIP and FINFO to list the details in your ZIP files.” Available at https://blogs.sas.com/content/sasdummy/2017/09/08/filename-zip-details/. Last modified September 8, 2017. Accessed March 29, 2025.

[4] Pavlov, I. “7-Zip”. Available at https://www.7-zip.org/. Accessed March 29, 2025.

[5] Viel, KR. 2025. “A SAS® System PowerShell macro to report directory-level metrics by Owner (programmer).” Proceedings of the PharmaSUG 2025 Conference, San Diego, CA: PharmaSUG. In press.

[6] Viel, KR. 2023. “SAS® System Macros to Summarize the COMPARE Procedure Results and SAS Logs for a Directory or Single File.” Proceedings of the PharmaSUG 2023 Conference, San Diego, CA: PharmaSUG. https://pharmasug.org/proceedings/2023/SD/PharmaSUG-2023-SD-221.pdf

[7] Pusarla SK, Viel K, and Di Tommaso D. “An open-source project to map lab terminology and standardize units for analysis.” Proceedings of PHUSE US Connect 2022.
https://phuse.s3.eu-central-1.amazonaws.com/Archive/2022/Connect/US/Atlanta/PAP_DS02.pdf
https://phuse.s3.eu-central-1.amazonaws.com/Archive/2022/Connect/US/Atlanta/PRE_DS02.pdf

CONTACT INFORMATION

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

Kevin R. Viel, Ph.D.
Histonis LLC
Navitas Data Sciences
genepistat@gmail.com

Appendix 1. The MAC_U_7zip macro.

```
%macro mac_u_7zip
   ( command      =
   , archive_path  =
   , archive_name  =
   , path          = %str()
   , file          = %str()
   , pathfile      = %str()
   , path_file_ds  =
   , extract       = x
   , extract_o     = %str()
   , extract_c     = %str()
   , original_name = %str()
   , rename        = %str()
   , list_ds       = _7z_list
   , passwd        = %str()
   , switch        = %str()
   , help          = N
   ) ;

  skip ;
  %let mac_u_7zip_version = 1.00 ;
  %put BEGINNING OF EXECUTION: mac_u_7zip Version: &mac_u_7zip_version. ;
  skip ;

  %let xwait   = %sysfunc( getoption ( xwait )) ;
  %let xsync   = %sysfunc( getoption ( xsync )) ;
  %let xmin    = %sysfunc( getoption ( xmin  )) ;

  %if &help. = Y
  %then
     %do ;
        %let mprint_orig = %sysfunc(getoption(mprint)) ;
        options nomprint ;

        skip ;
        skip ;
        %put _____________________________________________________________________________________________________________ ;
        %put Purpose of program:     This utility macro creates or interacts with a zip archive using 7zip.          ;
        skip ;
        %put Macro Parameter         Description                                                                      ;
        %put _____________________   ________________________________________________________________________________ ;
        %put command              = Corresponds to the command for the command line of 7zip.                         ;
        %put %str(                        )Add - a%str(;) Extract - e%str(;)                                         ;
        %put %str(                        )List - l%str(;) Rename - rn%str(;)                                        ;
        %put %str(                        )Delete - d                                                                ;
        %put archive_path         = Path of zip archive                                                            ;
        %put archive_name         = Name of zip archive                                                            ;
        %put path                 = Path of file to archive                                                        ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put file                 = Name of file to archive (include extension or wildcard)                        ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put pathfile             = The path and filename of a single file of interest (to archive).               ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put path_file_ds         = Name of data set with paths/files                                              ;
        %put extract              = x (Extract with full paths) command (Extracts files from an archive with their ;
        %put %str(                        )full paths in the current directory, or in an output directory if       ;
        %put %str(                        )specified.                                                              ;
        %put %str(                       )e (Extract) command (Extracts files from an archive to the current       ;
        %put %str(                        )directory or to the output directory.)                                  ;
        %put %str(                         )Default: x                                                             ;
        %put extract_o            = The directory corresponding to the switch -o{Directory} : set Output directory ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put extract_c            = Extract criteria.                                                              ;
        %put %str(                        )For instance: Version_001_000/adsl.sas                                  ;
        %put %str(                                      )*.sas -r                                                  ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put original_name        = Name of file in the archive to change.                                         ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put rename               = New name of file                                                               ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put list_ds              = Output data set for the parsed output from the 7zip listing of the archive.    ;
        %put %str(                         )Default: _7z_list                                                      ;
        %put passwd               = password for the -p switch                                                     ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put switch               = Additional switches appending to the end of the command line                   ;
        %put %str(                        )Example:  -y (Assume Yes on all queries) switch                         ;
        %put %str(                         )Default: %nrstr(%%)str%str(())                                         ;
        %put _____________________________________________________________________________________________________________ ;
        %put End of help                                                                                            ;

        options &mprint_orig. ;

        %goto __END ;
     %end ;

  /*********************************************************************/
  options noxwait
          xsync
          xmin
          ;

  /* Remove the trailing \ if any */
```

```
  93    %let archive_path = %sysfunc( prxchange( s/(^*.)\\$/$1/ , 1 , %nrbquote(&archive_path.))) ;
  94    %if %sysfunc( fileexist( &archive_path. )) = 0
  95    %then
  96       %do ;
  97           %put W%str(ARNING: ) &archive_path. does not exists.  ;
  98           %goto __END ;
  99       %end ;
 100
 101   /* For all commands but add, the archive must already exists */
 102   %if %sysfunc( prxmatch( /^(?:add|extract|list|rename|delete)$/i , &command. )) = 0
 103   %then
 104      %do ;
 105          %put W%str(ARNING: ) command must be one of add, extract, list, rename, or delete. ;
 106          %put W%str(ARNING: ) command = &command. ;
 107          %goto __END ;
 108      %end ;
 109
 110   /* For all commands but add, the archive must already exists */
 111   %if %upcase(&command.) ne ADD
 112   %then
 113      %do ;
 114          %if %sysfunc( fileexist( &archive_path.\&archive_name. )) = 0
 115          %then
 116             %do ;
 117                 %put W%str(ARNING: ) &archive_path.\&archive_name. does not exists.  ;
 118                 %goto __END ;
 119             %end ;
 120      %end ;
 121
 122   /* Extract command */
 123   %if %sysfunc( prxmatch( /^(?:e|x)$/i , %nrbquote(&extract.))) = 0
 124   %then
 125      %do ;
 126          %put W%str(ARNING: ) EXTRACT = &extract. must be X or E ;
 127          %goto __END ;
 128      %end ;
 129
 130
 131   /***************/
 132   %if %nrbquote(&passwd.) ne %str()
 133   %then %let passwd = -p&passwd. ;
 134
 135   /*************************************/
 136   /* BLOCK_1_A */
 137   %if %upcase(&command.) = ADD
 138   %then
 139      %do ;
 140          /****/
 141          %if     %nrbquote(&path.) ne %str()
 142              and %nrbquote(&file.) ne %str()
 143          %then
 144             %do ;
 145                 /* Remove the trailing \ if any */
 146                 %let path         = %sysfunc( prxchange( s/(^*.)\\$/$1/ , 1 , %nrbquote(&path.))) ;
 147
 148                 %if %sysfunc( fileexist( &path.\&file.))
 149                 %then
 150                    %do ;
 151                        /* Create a ZIP archive in the destination path/folder */
 152                        x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" a %")&archive_path.\&archive_name.%str(%"
 153   %")&path.\&file.%str(%") &passwd. &switch. %str(%')) ;
 154                    %end ;
 155                 %else
 156                    %do ;
 157                        %put W%str(ARNING: ) &path.\&file. does not exists.  ;
 158                        %goto __END ;
 159                    %end ;
 160             %end ;
 161
 162           %else %if %nrbquote(&pathfile.) ne %str()
 163           %then
 164              %do ;
 165                 %if %sysfunc( fileexist( &pathfile.))
 166                 %then
 167                    %do ;
 168                        /* Create a ZIP archive in the destination path/folder */
 169                        x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" a %")&archive_path.\&archive_name.%str(%"
 170   %")&pathfile.%str(%") &passwd. &switch. %str(%')) ;
 171                    %end ;
 172                 %else
 173                    %do ;
 174                        %put W%str(ARNING: ) &pathfile. does not exists.  ;
 175                        %goto __END ;
 176                    %end ;
 177             %end ;
 178
 179           %else %if %sysfunc( exist( &path_file_ds. ))
 180           %then
 181              %do ;
 182                  %let files_exist = 1 ;
 183
 184                 data _null_ ;
 185
 186                   length pathfile $ 500
 187                          path     $ 500
```

```
                          file     $ 100
                          ;

                    retain __files_exist 1 ;

                    if _n_ = 1
                    then call missing( pathfile
                                     , path
                                     , file
                                     ) ;

                    set &path_file_ds. ;

                    if     pathfile =  " "
                       and path     ne " "
                       and file     ne " "
                     then pathfile = catx( "\"
                                         , prxchange( "s/(^*.)\\$/$1/"
                                             , 1
                                             , path
                                             )
                                         , file
                                         ) ;

                    __rc = fileexist( pathfile ) ;
                    if __rc ne 1
                    then
                      do ;
                         put "W" "ARNING: does not exist "
                             pathfile=
                             ;
                         if __files_exist = 1
                         then
                           do ;
                              __files_exist = 0 ;
                              call symput( "files_exist" , "0" ) ;
                           end ;
                      end ;
                run ;

                %if &files_exist. = 1
                %then
                   %do ;
                       proc sql noprint ;
                         select quote( catx( "\"
                                           , prxchange( "s/(^*.)\\$/$1/" , 1 , strip( path ))
                                           , file
                                           )
                                     ) into : pathfile separated by " "
                         from &path_file_ds.
                         ;
                       quit ;

                       x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" a %")&archive_path.\&archive_name.%str(%"
)&pathfile. &passwd. &switch. %str(%')) ;
                   %end ;

                %else
                   %do ;
                       %goto __END ;
                    %end ;

            %end ;/* END OF sysfunc( exist( path_file_ds )) */

    %end ;/* END OF upcase(command) = ADD */
    /* END OF BLOCK_1_A */

  /***************/
  /* BLOCK_2_A */
  %if %upcase(&command.) = EXTRACT
  %then
     %do ;
         %if %nrbquote(&extract_o.) = %str()
         %then
            %do ;
                %put W%str(ARNING: ) Please specify the extraction directory (extract_o). ;
                %goto __END ;
            %end ;


         %if     %nrbquote(&extract_o.) ne %str()
             and %nrbquote(&extract_c.) ne %str()
         %then
            %do ;
               x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" &extract. %")&archive_path.\&archive_name.%str(%" -
o%")&extract_o.%str(%") %str(%")&extract_c.%str(%" -r ) &passwd. &switch. %str(%')) ;
            %end ;

          %else %if     %nrbquote(&extract_o.) ne %str()
                    and %nrbquote(&extract_c.) =  %str()
          %then
             %do ;
                x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" &extract. %")&archive_path.\&archive_name.%str(%" -
o%")&extract_o.%str(%") &passwd. &switch. %str(%')) ;
            %end ;
```

```
283
284       %end ;/* END OF upcase(command) = EXTRACT */
285       /* END OF BLOCK_2_A */
286
287     /***************/
288     /* BLOCK_3_A */
289     %if %upcase(&command.) = LIST
290     %then
291        %do ;
292           data _&list_ds.
293                  ( keep   = date
294                             time
295                             name
296                  )
297                ;
298
299              infile %unquote(%str(%'C:\PROGRA~1\7-Zip\7z.exe l %")&archive_path.\&archive_name.%str(%") &passwd. &switch.
300   %str(%'))
301                     pipe
302                     ;
303
304              length name   $  500
305                     __line $ 1000
306                     ;
307
308              retain flag       " "
309                     name_start .
310                     ;
311              input ;
312
313              if flag = "1"
314              then
315                do ;
316                   if compress( _infile_ , "- " ) = " " then stop ;
317                   date = input( scan( _infile_ , 1 , " " ) , yymmdd10. ) ;
318                   time = input( scan( _infile_ , 2 , " " ) , time. ) ;
319
320                   /* This accounts for paths/filenames with spaces */
321                   __line = _infile_ ;
322                   name   = substr( __line , name_start ) ;
323
324                   output ;
325                end ;
326
327              if prxmatch( "/Date\s+Time\s+Attr\s+Size\s+Compressed\s+Name/" , _infile_ )
328              then
329                do ;
330                   name_start = prxmatch( "/Name/" , _infile_ ) ;
331                   input ;
332                   flag = "1" ;
333                end ;
334
335              format date yymmdd10.
336                     time time8.
337                     ;
338
339           run ;
340
341        %end ;/* END OF upcase(command) = LIST */
342        /* END OF BLOCK_3_A */
343
344     /* BLOCK_4_A */
345     %if %upcase(&command.) = RENAME
346     %then
347        %do ;
348           x %unquote(%str(%"C:\Program Files\7-Zip\7z.exe%" rn %")&archive_path.\&archive_name.%str(%"
349   %")&original_name.%str(%" %")&rename.%str(%") &passwd. &switch. %str(%')) ;
350        %end ;/* END OF upcase(command) = RENAME */
351        /* END OF BLOCK_4_A */
352
353     /* BLOCK_5_A */
354     %if %upcase(&command.) = DELETE
355     %then
356        %do ;
357           data __delete_archive
358                  ( keep   = name )
359                ;
360
361              infile %unquote(%str(%'C:\PROGRA~1\7-Zip\7z.exe l %")&archive_path.\&archive_name.%str(%") &passwd. &switch.
362   %str(%'))
363                     pipe
364                     ;
365
366              length name   $  500
367                     __line $ 1000
368                     ;
369
370              retain flag       " "
371                     name_start .
372                     ;
373              input ;
374
375              if flag = "1"
376              then
377                do ;
```

```
378                 if compress( _infile_ , "- " ) = " " then stop ;
379
380                 /* This accounts for paths/filenames with spaces */
381                 __line = _infile_ ;
382                 name   = substr( __line , name_start ) ;
383
384                 output ;
385               end ;
386
387             if prxmatch( "/Date\s+Time\s+Attr\s+Size\s+Compressed\s+Name/" , _infile_ )
388             then
389               do ;
390                  name_start = prxmatch( "/Name/" , _infile_ ) ;
391                  input ;
392                  flag = "1" ;
393               end ;
394
395           run ;
396
397           /****/
398           %if     %nrbquote(&path.) ne %str()
399               and %nrbquote(&file.) ne %str()
400           %then
401              %do ;
402                 data __delete_ds
403                        ( keep   = name )
404                      ;
405
406                   length name   $  500 ;
407
408                   name = catx( "\"
409                              , prxchange( "s/(?<=^[A-Z]):://i"
410                                         , 1
411                                         , "&path."
412                                         )
413                              , "&file."
414                              ) ;
415                 run ;
416              %end ;
417
418            %else %if %nrbquote(&pathfile.) ne %str()
419            %then
420               %do ;
421                  data __delete_ds
422                         ( keep   = name )
423                       ;
424
425                   length name   $  500 ;
426
427                   name = prxchange( "s/(?<=^[A-Z]):://i"
428                                   , 1
429                                   , "&pathfile."
430                                   ) ;
431                 run ;
432               %end ;
433
434            %else %if %sysfunc( exist( &path_file_ds. ))
435            %then
436               %do ;
437                  data __delete_ds
438                         ( keep   = name )
439                       ;
440
441                     length pathfile $ 500
442                            path     $ 500
443                            file     $ 100
444                            ;
445
446                     if _n_ = 1
447                     then call missing( pathfile
448                                      , path
449                                      , file
450                                      ) ;
451
452                     set &path_file_ds. ;
453
454                     if pathfile ne  " "
455                     then name = prxchange( "s/(?<=^[A-Z]):://i"
456                                          , 1
457                                          , pathfile
458                                          ) ;
459                      else if     path     ne " "
460                              and file     ne " "
461                     then name = catx( "\"
462                                     , prxchange( "s/(^*.)\\$/$1/"
463                                                , 1
464                                                , prxchange( "s/(?<=^[A-Z]):://i"
465                                                           , 1
466                                                           , path
467                                                           )
468                                                )
469                                     , file
470                                     ) ;
471                 run ;
472               %end ;/* END OF sysfunc( exist( path_file_ds )) */
```

```
473
474             /*********/
475             proc sql
476                undo_policy = none ;
477               create table __delete_ds as
478               select a.name
479                    , b.name ne " " as flag
480               from __delete_ds                  as a
481                    left join __delete_archive   as b
482                       on a.name = b.name
483               ;
484             quit ;
485
486             data _null_ ;
487               set __delete_ds
488                     ( where  = ( flag = 0 )) ;
489               put "E" "RROR: Not in &archive_path.\&archive_name.:  "
490                   name=
491                   ;
492             run ;
493
494             proc sql
495                noprint ;
496               select name
497                         into : __d_1 -
498               from __delete_ds
499               where flag = 1
500               ;
501             quit ;
502
503             %do __j = 1 %to &sqlobs. ;
504
505               x %unquote(%str(%'%"C:\Program Files\7-Zip\7z.exe%" d %")&archive_path.\&archive_name.%str(%"
506 %")&&__d_&__j.%str(%") &passwd. &switch. %str(%')) ;
507
508               %put A%str(LERT: ) &&__d_&__j. was deleted from &archive_path.\&archive_name. ;
509
510             %end ;
511
512      %end ;/* END OF upcase(command) = DELETE */
513      /* END OF BLOCK_5_A */
514
515   %__END :
516
517   options &xwait.
518           &xsync.
519           &xmin.
520           ;
521
522   skip ;
523   %put END OF EXECUTION: mac_u_7zip Version: &mac_u_7zip_version. ;
524   skip ;
525
526 %mend  mac_u_7zip ;
```

Appendix 2. The MAC_U_FILE_ARCHIVE macro.

```
%macro mac_u_file_archive
   ( archive_path  = %str()
   , archive_name  = %str()
   , passwd        = %str()
   , pathfile      = %str()
   , final         = N
   , library_csv   = %str()
   , help          = N
   ) ;

  skip ;
  %let mac_u_file_archive_version = 1.00 ;
  %put BEGINNING OF EXECUTION: mac_u_file_archive Version: &mac_u_file_archive_version. ;
  skip ;

  %let xwait   = %sysfunc( getoption ( xwait )) ;
  %let xsync   = %sysfunc( getoption ( xsync )) ;
  %let xmin    = %sysfunc( getoption ( xmin  )) ;

  %if &help. = Y
  %then
     %do ;
         %let mprint_orig = %sysfunc(getoption(mprint)) ;
         options nomprint ;

         skip ;
         skip ;
         %put ____________________________________________________________________________________________________________________________ ;
         %put Purpose of program:     This utility macro uses 7zip to work with a zip archive, for instance to create a new          ;
         %put %str(                        )zip archive and add a file to it, to rename the file for archival purposes, to list the  ;
         %put %str(                        )contents of the zip archive, or to delete a file from the archive.  This macro also      ;
         %put %str(                        )creates, modifies, or adds to a Table of Contents (CSV) corresponding to the zip         ;
         %put %str(                        )archive.                                                                                 ;
         skip ;
         %put %str(                        )This macro has two modes: FINAL = Y and FINAL = N.                                       ;
         skip ;
         %put %str(                        )For FINAL = Y, if the file of interest is younger than or the same age as the file in    ;
         %put %str(                        )the archive, then it will not be replaced.                                               ;
         skip ;
         %put %str(                        )For FINAL = N, if the file of interest is not same age as the file in the archive, then  ;
         %put %str(                        )the file is added to the archive, then the name is changed by appending the datetime     ;
         %put %str(                        )to the filename.                                                                         ;
         skip ;
         %put %str(                        )This macro requires 7zip: https://www.7-zip.org/. The reason this program was chosen     ;
         %put %str(                        )over the SAS ZIP engine for the FILENAME statement is that 7zip preserves the datetime   ;
         %put %str(                        )of the archived then extracted file. See:                                                ;
         %put %str(                        )https://blogs.sas.com/content/sasdummy/2015/05/11/using-filename-zip-to-unzip-and-read-
data-files-in-sas/ ;
         skip ;
         %put Macro Parameter         Description                                                                                       ;
         %put _________________       ____________________________________________________________________________________________ ;
         %put archive_path          = The path to the zip archive of interest.                                                        ;
         %put %str(                          )Default: %nrstr(%%)str%str(())                                                         ;
         %put archive_name          = The name of the zip archive of interest.                                                        ;
         %put %str(                          )Default: %nrstr(%%)str%str(())                                                         ;
         %put pathfile              = The path and filename, with extention, of the file of interest.                                 ;
         %put %str(                          )Default: %nrstr(%%)str%str(())                                                         ;
         %put final                 = Whether the file of interest is the final version.  "Final" may mean Validated and approved,   ;
         %put %str(                        )but not the latest version.  FINAL is of interest so that one may search an archive for  ;
         %put %str(                        )"production" programs.                                                                   ;
         %put %str(                          )Default: N                                                                             ;
         %put library_csv           = The path and filename of the .csv file that may act as the table of contents of the archive.   ;
         %put %str(                          )Default: %nrstr(%%)str%str(())                                                         ;
         %put ____________________________________________________________________________________________________________________________ ;
         %put End of help                                                                                                              ;

         options &mprint_orig. ;

         %goto __END ;
     %end ;

  /*********************************************************************/
  options noxwait
          xsync
          xmin
          ;

  %if %nrbquote(&archive_path.) = %str()
  %then
     %do ;
         %put W%str(ARNING: ) archive_path cannot be missing  ;
         %goto __END ;
     %end ;

  %if %nrbquote(&archive_name.) = %str()
  %then
     %do ;
         %put W%str(ARNING: ) archive_name cannot be missing  ;
         %goto __END ;
     %end ;

  %if %nrbquote(&pathfile.) = %str()
```

```
 93    %then
 94       %do ;
 95           %put W%str(ARNING: ) pathfile cannot be missing  ;
 96           %goto __END ;
 97       %end ;
 98
 99    %if %sysfunc( fileexist( &pathfile. )) = 0
100    %then
101       %do ;
102           %put W%str(ARNING: ) &pathfile. does not exists.  ;
103           %goto __END ;
104       %end ;
105
106    /* Remove the trailing \ if any */
107    %let archive_path = %sysfunc( prxchange( s/(^*.)\\$/$1/ , 1 , %nrbquote(&archive_path.))) ;
108    %if %sysfunc( fileexist( &archive_path. )) = 0
109    %then
110       %do ;
111           %put W%str(ARNING: ) &archive_path. does not exists.  ;
112           %goto __END ;
113       %end ;
114
115    %if %nrbquote(&library_csv.) = %str()
116    %then %let library_csv = %sysfunc( prxchange( s/^(.*\\[^\\]+\.)(?:[^\\]+)/$1csv/i
117                                                , 1
118                                                , %nrbquote(&archive_path.\&archive_name.)
119                                                )
120                                     ) ;
121
122    %if %sysfunc( exist( library_csv ))
123    %then
124       %do ;
125           proc datasets
126              library = work
127              nolist
128              ;
129             delete library_csv ;
130           quit ;
131       %end ;
132
133    /*************************************************************************/
134    /* Obtain the datetime stamp for the file of interest. */
135    %mac_u_finfo
136       ( in_ds                = %str()
137       , pathfile             = &pathfile.
138       , keep                 = pathfile
139       , foptname             = Last Modified
140       , print                = N
141       , delete               = files
142       ) ;
143
144    proc sql
145       noprint ;
146             /* For example: last_modified = 14JAN2023:12:45:10 */
147             /* 20230114T124510, this is for inclusion into the file name for FINAL = N (versioned). */
148      select put( last_modified , b8601dt. )
149             /* 2023-01-14T12:45:10, this is for records, like the LIBRARY_CSV. */
150           , put( last_modified , e8601dt. )
151                into : dt  separated by ""
152                   , : edt separated by ""
153      from finfo
154      ;
155    quit ;
156
157    proc datasets
158       library = work
159       nolist
160       ;
161      delete finfo
162      ;
163    quit ;
164
165    /* Original name of the file (without its path)
166
167       For example:
168          C:\Users\vielk\Documents\My SAS Files\9.4\Autocall_Macros\mac_u_file_archive.sas
169          mac_u_file_archive.sas
170    */
171    %let original = %sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
172                                       , 1
173                                       , %nrbquote(&pathfile.)
174                                       )
175                            ) ;
176
177
178    /* If the type of archive is not a final version, then the datetime stamp will be incorporated in the name.
179
180         1) Remove the colon (the drive becomes a "folder" in the zip archive.
181         2) Add the last_modified datetime value into the file name.
182
183       For example:
184          C:\Users\vielk\Documents\My SAS Files\9.4\Autocall_Macros\mac_u_file_archive.sas
185          C\Users\vielk\Documents\My SAS Files\9.4\Autocall_Macros\mac_u_file_archive_20250119T135437.sas
186    */
187    %if &final. = N
```

```
188   %then %let rename_file = %sysfunc( prxchange( s/(.*)(\.[0-9a-z]+)$/$1_&dt.$2/
189                                               , 1
190                                               , %sysfunc( prxchange( s/^\\\\//
191                                                                    , 1
192                                                                    , %sysfunc( prxchange( s/(?<=^[A-Z]):/i
193                                                                                         , 1
194                                                                                         , %nrbquote(&pathfile.)
195                                                                                         )
196                                                                              )
197                                                                    )
198                                                          )
199                                               )
200                                     ) ;
201
202
203    /* If the type of archive is a final version, the current final version, if any will be overwritten
204
205         1) Remove the colon (the drive becomes a “folder” in the zip archive.
206
207       For example:
208          C:\Users\vielk\Documents\My SAS Files\9.4\Autocall_Macros\mac_u_file_archive.sas
209          C\Users\vielk\Documents\My SAS Files\9.4\Autocall_Macros\mac_u_file_archive.sas
210    */
211    %else %let rename_file = %sysfunc( prxchange( s/^\\\\//
212                                               , 1
213                                               , %sysfunc( prxchange( s/(?<=^[A-Z]):/i
214                                                                    , 1
215                                                                    , %nrbquote(&pathfile.)
216                                                                    )
217                                                          )
218                                               )
219                                      ) ;
220
221   /*************************************************************************/
222   %if %sysfunc( fileexist( &library_csv. ))
223   %then
224      %do ;
225          data library_csv ;
226            infile “&library_csv.”
227                   dsd
228                   dlm = “2c”x
229                   ;
230            length Program   $ 100
231                   Path      $ 256
232                   DateTime  $  19
233                   final     $   1
234                   comment   $ 200
235                   ;
236            input Program
237                  Path
238                  DateTime
239                  final
240                  comment
241                  ;
242          run ;
243      %end ;
244
245   /* BLOCK 1_A */
246   %if %sysfunc( fileexist( &archive_path.\&archive_name. ))
247   %then
248      %do ;
249          %mac_u_7zip
250             ( command       = list
251             , archive_path  = &archive_path.
252             , archive_name  = &archive_name.
253             , passwd        = &passwd.
254             ) ;
255
256          /* The following two macro variables pertain to the zip archive. */
257          %let exists = 0 ;
258          %let delete = 1 ;
259
260          data _null_ ;
261
262            set __7z_list
263                  ( where = ( name = “&rename_file.” ))
264                ;
265
266            %if &final. = Y
267            %then
268               %do ;
269                   __dt = input( “&dt.” , b8601dt. ) ;
270
271                   /* If the file of interest is younger than or the same age as the file in the archive, then do not delete it
272                      from the zip archive.  Note that if one added a file with FINAL = N, did not update (save) it, then
273                      added the same file with FINAL = Y, then both entries would be in the zip archive, one with a datetime component
274                      and one without. bit
275
276                      Obs                          name                                  date        time
277
278                       1     ~_Macros\mac_u_file_archive_20250119T185238.sas    2025-01-19    18:52:38
279                       2     ~\mac_u_file_archive.sas                           2025-01-19    18:52:38
280
281                      Obs                          name                                  date        time
282
```

```
283                        1     ~_Macros\mac_u_file_archive.sas                    2025-01-19    18:52:38
284                        2     ~\mac_u_file_archive_20250119T185238.sas           2025-01-19    18:52:38
285                        3     ~\mac_u_file_archive_20250119T185708.sas           2025-01-19    18:57:08
286                    */
287                    if     datepart( __dt ) < date
288                       or  (     datepart( __dt ) =  date
289                             and timepart( __dt ) <= time
290                           )
291                    then call symput( "delete"   , "0" ) ;
292                %end ;
293
294              call symput( "exists"    , "1" ) ;
295                        /* foi = file of interest */
296              call symputx( "foi_date" , put( date ,  yymmdd10. -L )) ;
297              call symputx( "foi_time" , put( time ,  tod11.    -L )) ;
298
299            run ;
300
301            proc datasets
302               library = work
303               nolist
304               ;
305              delete __7z_list ;
306            quit ;
307
308            %if &delete. = 0
309            %then
310            %do ;
311                %put W%str(ARNING: ) &pathfile. already has been archived with the timestamp: ;
312                %put W%str(ARNING: ) &foi_date.T&foi_time. ;
313                %put W%str(ARNING: ) &edt. is the Last Modified value of the file of interest ;
314                %put W%str(ARNING: ) FINAL = &final. ;
315                %goto __END ;
316            %end ;
317
318            /* This would be the first version in the archive */
319            %if     &final.  = Y
320                and &exists. = 0
321            %then
322               %do ;
323                   %if %sysfunc( fileexist( &library_csv. )) = 0
324                   %then
325                      %do ;
326                          data _null_ ;
327
328                            file "&library_csv."
329                                 dsd
330                                 dlm = "2c"x
331                                 ;
332                            if _n_ = 1
333                            then put "Program,Path,DateTime,Final,Description/Comments" ;
334
335                            program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
336                                                          , 1
337                                                          , %nrbquote(&pathfile.)
338                                                          )
339                                                )" ;
340
341                            path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
342                                                       , 1
343                                                       , %nrbquote(&pathfile.)
344                                                       )
345                                           )" ;
346
347                            datetime = "&edt." ;
348                            final    = "&final." ;
349                            comment  = " " ;
350                            put program
351                                path
352                                datetime
353                                final
354                                comment
355                                ;
356                          run ;
357                      %end ; /* END OF sysfunc( fileexist( library_csv )) = 0 */
358
359                   /* The CSV file exists, append to it using MOD */
360                   %else
361                      %do ;
362                          data _null_ ;
363
364                            file "&library_csv."
365                                 dsd
366                                 dlm = "2c"x
367                                 mod
368                                 ;
369
370                            program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
371                                                          , 1
372                                                          , %nrbquote(&pathfile.)
373                                                          )
374                                                )" ;
375
376                            path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
377                                                       , 1
```

```
                                                , %nrbquote(&pathfile.)
                                                )
                                        )" ;

                            datetime = "&edt." ;
                            final    = "&final." ;
                            comment  = " " ;
                            put program
                                path
                                datetime
                                final
                                comment
                                ;
                         run ;
                   %end ; /* END OF else sysfunc( fileexist( library_csv )) = 0 */
               %end ; /* END OF final = Y and exists = 0 */

            /*************************************************/
            %if     &final.  = Y
                and &exists. = 1
                and &delete. = 1
            %then
               %do ;

                   /* Delete the file from the archive */
                   /* x %unquote(%str(%'"C:\Program Files\7-Zip\7z.exe" d ")&archive_path.\&archive_name.%str("
%")&rename_file.%str("%')) ; */
                   %mac_u_7zip
                      ( command      = delete
                      , archive_path  = &archive_path.
                      , archive_name  = &archive_name.
                      , pathfile      = &rename_file.
                      , passwd        = &passwd.
                      ) ;

                   /* After the deletion, the file no longer exists in the zip archive, so reset EXISTS */
                   %let exists = 0 ;

                   %if %sysfunc( exist( library_csv ))
                   %then
                      %do ;
                          /* Update the datetime of the file of interest */
                          data _null_ ;

                            file "&library_csv."
                                 dsd
                                 dlm = "2c"x
                                 ;

                            set library_csv ;

                            if cats( path
                                   , program
                                   ) = "&pathfile."
                            then
                              do ;
                                 datetime = "&edt." ;
                                 final    = "&final." ;
                              end ;

                            put Program
                                Path
                                DateTime
                                final
                                comment
                                ;
                           run ;
                      %end ;

               %end ; /* END OF final = Y and exists = 1 and delete = 1 */

            /********************/
            /****  FINAL = N ****/
            /********************/
            %if     &final.  = N
                and &exists. = 0
            %then
               %do ;
                   /* This would be the first entry, create the LIBRARY_CSV file */
                   %if %sysfunc( fileexist( &library_csv. )) = 0
                   %then
                      %do ;
                          data _null_ ;

                            file "&library_csv."
                                 dsd
                                 dlm = "2c"x
                                 ;
                            if _n_ = 1
                            then put "Program,Path,DateTime,Final,Description/Comments" ;

                            program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
                                                          , 1
                                                          , %nrbquote(&pathfile.)
                                                          )
```

```
                                          )" ;

                        path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
                                                   , 1
                                                   , %nrbquote(&pathfile.)
                                                   )
                                       )" ;

                        datetime = "&edt." ;
                        final    = "&final." ;
                        comment  = " " ;
                        put program
                            path
                            datetime
                            final
                            comment
                            ;
                    run ;
                %end ; /* END OF sysfunc( fileexist( library_csv )) = 0 */

              /* The CSV file exists, append to it using MOD */
              %else
                 %do ;
                     data _null_ ;

                       file "&library_csv."
                            dsd
                            dlm = "2c"x
                            mod
                            ;

                       program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
                                                     , 1
                                                     , %nrbquote(&pathfile.)
                                                     )
                                          )" ;

                       path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
                                                  , 1
                                                  , %nrbquote(&pathfile.)
                                                  )
                                       )" ;

                       datetime = "&edt." ;
                       final    = "&final." ;
                       comment  = " " ;
                       put program
                           path
                           datetime
                           final
                           comment
                           ;
                     run ;
                %end ; /* END OF else sysfunc( fileexist( library_csv )) = 0 */
          %end ; /* END OF final = N and exists = 0 */

    %end ; /* END OF sysfunc( fileexist( archive_path\archive_name )) */
    /* END OF BLOCK_1_A */

  /* BLOCK_1_B: the zip archive (archive_path\archive_name) does not e%str(xist) */
  %else
     %do ;
         %let exists = 0 ;

         /* If the LIBRARY_CSV file does not e%str(xist), the create it and add the first entry. */
         %if %sysfunc( fileexist( &library_csv. )) = 0
         %then
            %do ;
                data _null_ ;

                  file "&library_csv."
                       dsd
                       dlm = "2c"x
                       ;
                    if _n_ = 1
                    then put "Program,Path,DateTime,Final,Description/Comments" ;

                    program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
                                                  , 1
                                                  , %nrbquote(&pathfile.)
                                                  )
                                       )" ;

                    path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
                                               , 1
                                               , %nrbquote(&pathfile.)
                                               )
                                    )" ;

                    datetime = "&edt." ;
                    final    = "&final." ;
                    comment  = " " ;
                    put program
                        path
                        datetime
```

```
                        final
                        comment
                        ;
                run ;

            %end ; /* END OF sysfunc( fileexist( library_csv )) = 0 */

         /* If the LIBRARY_CSV file e%str(xists), then add the entry. */
         %else
            %do ;
                data _null_ ;

                  file "&library_csv."
                       dsd
                       dlm = "2c"x
                       ;

                  retain flag " " ;

                  set library_csv
                         end = end
                      ;

                  /* Update the datetime for the path/program of interest */
                  if cats( path
                         , program
                         ) = "&pathfile."
                  then
                    do ;
                       datetime = "&edt." ;
                       flag     = "1" ;
                    end ;

                  put program
                      path
                      datetime
                      final
                      comment
                      ;

                  if     end
                     and flag = " "
                  then
                    do ;
                       program = "%sysfunc( prxchange( s/^(?:.*\\)([^\\]+)$/$1/
                                                     , 1
                                                     , %nrbquote(&pathfile.)
                                                     )
                                         )" ;

                       path = "%sysfunc( prxchange( s/^(.*\\)(?:[^\\]+)$/$1/
                                                   , 1
                                                   , %nrbquote(&pathfile.)
                                                   )
                                       )" ;

                       datetime = "&edt." ;
                       final    = "&final." ;
                       comment  = " " ;
                       put program
                           path
                           datetime
                           final
                           comment
                           ;
                    end ;

                run ;
            %end ; /* END OF else sysfunc( fileexist( library_csv )) = 0 */

      %end ; /* END OF else %sysfunc( fileexist( archive_path\archive_name )) */
      /* END OF BLOCK_1_B */

   /* Update the zip archive */
   /* If the entry is a duplicate, that is a path and file with the matching datetime
      already exists, then send a note to the log and exit the macro
   */
   %if &exists. = 1
   %then
      %do ;
          %put W%str(ARNING: ) &pathfile. already has been archived with the timestamp: ;
          %put W%str(ARNING: ) &foi_date.T&foi_time. ;
          %put W%str(ARNING: ) &edt. is the Last Modified value of the file of interest ;
          %put W%str(ARNING: ) FINAL = &final. ;
          %goto __END ;
      %end ;

   /* If the entry is not a duplicate, then update the archive and rename the entry to include the datetime */
   %mac_u_7zip
      ( command       = add
      , archive_path  = &archive_path.
      , archive_name  = &archive_name.
      , pathfile      = &pathfile.
      , passwd        = &passwd.
      ) ;
```

```
663
664     %mac_u_7zip
665          ( command       = rename
666          , archive_path  = &archive_path.
667          , archive_name  = &archive_name.
668          , original_name = &original.
669          , rename        = &rename_file.
670          , passwd        = &passwd.
671          ) ;
672
673     %__END :
674
675     options &xwait.
676             &xsync.
677             &xmin.
678             ;
679
680     skip ;
681     %put END OF EXECUTION: mac_u_file_archive Version: &mac_u_file_archive_version. ;
682     skip ;
683
684   %mend  mac_u_file_archive ;
```

Appendix 3. The MAC_PGM_ARCHIVE macro.

```
%macro mac_pgm_archive
   ( archive_path  = C:\Users\&sysuserid.\Documents\file_archive
   , archive_name  = file_archive.zip
   , ds            = files_archive
   , final         =
   ) ;

  %if %sysfunc( fileexist( &archive_path. )) = 0
  %then
     %do ;
         %mac_u_dcreate
            ( path   = &archive_path. ) ;
     %end ;

  /***/
  %if %sysfunc( index( &ds. , . ))
  %then
     %do ;
         %let lr = %sysfunc( scan( &ds. , 1 , . )) ;
         %let mn = %sysfunc( scan( &ds. , 2 , . )) ;
     %end ;
   %else
      %do ;
          %let lr = WORK ;
          %let mn = &ds. ;
      %end ;

  %let ds_final = %str() ;

  proc sql
     noprint ;
    select name
              into : ds_final
    from sashelp.vcolumn
    where     upcase( libname ) = upcase( "&lr." )
          and upcase( memname ) = upcase( "&mn." )
          and upcase( name    ) = "FINAL"
    ;
  quit ;

  /***/
  proc sql
     noprint ;
    %if &ds_final. = %str()
    %then select pathfile
                    into : pf_1 -
    ;
     %else select pathfile
                , final
                    into : pf_1 -
                       , : f_1  -
           ;
    from &ds.
    ;
  quit ;

  %do __i = 1 %to &sqlobs. ;

     %mac_u_file_archive
        ( archive_path  = &archive_path.
        , archive_name  = &archive_name.
        , pathfile      = &&pf_&__i.
        , final         = %if &ds_final. = %str()
                          %then &final. ;
                           %else &&f_&__i. ;
        ) ;
  %end ;

%mend mac_pgm_archive ;
```

Appendix 4. The test macro mac_pgm_list_7z_csv.

```
%macro mac_pgm_list_7z_csv ;

  %mac_u_7zip
     ( command      = list
     , archive_path  = C:\Users\&sysuserid.\Documents\file_archive
     , archive_name  = file_archive.zip
     ) ;

  proc sort
     data = __7z_list ;
    by name ;
  run ;

  proc print
     data = __7z_list ;
  run ;

  proc datasets
     library = WORK
     nolist
     ;
    delete __7z_list ;
  quit ;

  data csv ;
    infile "C:\Users\&sysuserid.\Documents\file_archive\file_archive.csv"
           dsd
           dlm       = "2c"x
           firstobs  = 2
           ;

    length program    $  50
           path       $ 200
           datetime   $  30
           final      $   1
           version    $   5
           comment    $  50
           ;

    input program
          path
          datetime
          final
          version
         comment
         ;
  run ;

  proc sort
     data = csv ;
    by program
       version
       datetime
       ;
  run ;

  proc print
     data = csv ;
  run ;

  proc datasets
     library = WORK
     nolist
     ;
    delete csv ;
  quit ;

%mend mac_pgm_list_7z_csv ;
```