The SAS Dummy
A SAS® blog for the rest of us

Using FILENAME ZIP to unzip and read data files in SAS

Chris Hemedinger | MAY 11, 2015

I've written about how to use the FILENAME ZIP method to read and update ZIP files in your SAS programs. The ZIP method was added in SAS 9.4, and its advantage is that you can accomplish more in SAS *without* having to launch external utilities such as WinZip, gunzip, or 7-Zip.

Several readers replied with questions about how you can use the content of these ZIP files within your SAS program. The basic scenario is: "I've got some data files in my ZIP archive. I want to use SAS to unzip these and then use them as data within my SAS process. Can I do this?"

Yes, you can -- but it does require an extra step. Even though FILENAME ZIP can show you the contents and structure of your ZIP file, most SAS procedures cannot access the content directly while it's in the archive. So, the additional step is to copy the file to another location, effectively extracting it from the ZIP file.

As an example, I created a ZIP file with two files and a subfolder:

```
data.zip
  |__ sas_tech_talks_15.xlsx
  |__ sas/
      |__ instanttitles.sas7bdat
```

This SAS program helps me to discover how FILENAME ZIP sees the file:

```
filename inzip ZIP "c:\projects\data.zip";

/* Read the "members" (files) from the ZIP file */
data contents(keep=memname isFolder);
 length memname $200 isFolder 8;
 fid=dopen("inzip");
 if fid=0 then
  stop;
 memcount=dnum(fid);
 do i=1 to memcount;
  memname=dread(fid,i);
  /* check for trailing / in folder name */
  isFolder = (first(reverse(trim(memname)))='/');
  output;
 end;
 rc=dclose(fid);
run;

/* create a report of the ZIP contents */
title "Files in the ZIP file";
proc print data=contents noobs N;
run;
```

Output:

```
        Files in the ZIP file

memname                       isFolder

sas/                             1

sas/instanttitles.sas7bdat       0

sas_tech_talks_15.xlsx           0

               N = 3
```

With this information, I can now "copy" the XLSX file out of the ZIP file and then import it into a SAS data set. Notice how I can use the "member" syntax (fileref with the file I want in parentheses) to address a specific file in the ZIP archive. I want to copy just from the actual files, and not the folder-level entries.

```
/* identify a temp folder in the WORK directory */
```

```
filename xl "%sysfunc(getoption(work))/sas_tech_talks_15.xlsx" ;

/* hat tip: "data _null_" on SAS-L */
data _null_;
   /* using member syntax here */
   infile inzip(sas_tech_talks_15.xlsx)
       lrecl=256 recfm=F length=length eof=eof unbuf;
   file   xl lrecl=256 recfm=N;
   input;
   put _infile_ $varying256. length;
   return;
 eof:
   stop;
run;

proc import datafile=xl dbms=xlsx out=confirmed replace;
  sheet=confirmed;
run;
```

Sample output from my SAS log:

```
NOTE: The infile INZIP(sas_tech_talks_15.xlsx) is:
      Filename=c:\projects\data.zip,
      Member Name=sas_tech_talks_15.xlsx

NOTE: UNBUFFERED is the default with RECFM=N.
NOTE: The file XL is:
      Filename=C:\SAS Temporary Files\_TD396_\Prc2\sas_tech_talks_15.xlsx,
      RECFM=N,LRECL=256,File Size (bytes)=0,
      Last Modified=11May2015:11:38:59,
      Create Time=11May2015:11:20:23

NOTE: A total of 55 records were read from the infile library INZIP.
NOTE: 55 records were read from the infile INZIP(sas_tech_talks_15.xlsx).
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

To use the SAS data set in the file, I need to copy it into a location shared by a SAS

library. In this example, I will again use the WORK location. Because my SAS data set is in a logical subfolder (named "sas") within the archive, I need to include that path as part of the member syntax on the INFILE statement.

```
/* Copy a zipped data set into the WORK library */
filename ds "%sysfunc(getoption(work))/instanttitles.sas7bdat" ;

data _null_;
   /* reference the member name WITH folder path */
   infile inzip(sas/instanttitles.sas7bdat)
          lrecl=256 recfm=F length=length eof=eof unbuf;
   file   ds lrecl=256 recfm=N;
   input;
   put _infile_ $varying256. length;
   return;
 eof:
   stop;
run;

proc contents data=work.instanttitles;
run;
```

Partial output in my example:

```
                          Files in the ZIP file
                          The CONTENTS Procedure


Data Set Name        WORK.INSTANTTITLES            Observations          1475
Member Type          DATA                          Variables             6
Engine               V9                            Indexes               0
Created              01/29/2015 15:09:54           Observation Length    248
Last Modified        01/29/2015 15:09:54           Deleted Observations  0
Protection                                         Compressed            NO
Data Set Type                                      Sorted                NO
Label
Data Representation  WINDOWS_64
Encoding             wlatin1  Western (Windows)
```

Of course, all of this can be automated even further by writing SAS code that automatically iterates through the ZIP file member names and copies/imports each of the members as needed.

tags: *Copy Files*, *FILENAME ZIP*, *SAS 9.4*, *ZIP files*

20 Comments

Andrew Ziem

Posted May 11, 2015 at 4:04 pm | *Permalink*

Hoping for **better compression rates like ZIP** with more **seamless** access in SAS, I submitted a SAS ballot for much better compression for SAS data sets. If anyone else is interested in both, please consider voting on this ballot idea.

Reply

Andreas Menrath

Posted May 26, 2015 at 12:01 pm | *Permalink*

WARNING: binary file copy may cause trouble!
I just used your binary file copy snippet here:

```
data _null_;
/* using member syntax here */
infile inzip(sas_tech_talks_15.xlsx)
lrecl=256 recfm=F length=length eof=eof unbuf;
file xl lrecl=256 recfm=N;
input;
put _infile_ $varying256. length;
return;
eof:
stop;
run;
```

For 99% of my files it worked fine. But unfortunately it does not make a 1:1 copy because it drops UTF byte order marks!

I played around with the code, but was not able to fix it. It looks like the UTF BOM is dropped before it is copied into the _infile_ variable :-(

Reply

Chris Hemedinger

Posted May 26, 2015 at 12:07 pm | *Permalink*

Andreas, thanks for pointing that out. There are other methods to copy files within SAS, including the method that I shared here from SAS expert Bruno Mueller. I had used the method in this post as a bit of a shortcut.

Reply

Paige Miller

Posted June 9, 2015 at 3:49 pm | *Permalink*

The following is a SASLOG with an error when I try to duplicate your results in SAS 9.4 TS1M2. What is this error?

```
131 filename inzip ZIP "c:\users\pmiller\documents\be_output\be_output.zip";
132 filename xl "c:\users\pmiller\documents\be_output\may 2013 be monthly update.xml"
;
133 /* hat tip: "data _null_" on SAS-L */
134 data _null_;
135 /* using member syntax here */
136 infile inzip(May_2013_BE_Monthly_Update.xml) lrecl=256 recfm=F length=length
eof=eof unbuf;
137 file xl lrecl=256 recfm=N;
138 input;
139 put _infile_ $varying256. length;
140 return;
141 eof:
142 stop;
143 run;
ERROR: Open failure for c:\users\pmiller\documents\be_output\be_output.zip during
attempt to create a local file handle.
NOTE: UNBUFFERED is the default with RECFM=N.
NOTE: The file XL is:
Filename=c:\users\pmiller\documents\be_output\may 2013 be monthly update.xml,
RECFM=N,LRECL=256,File Size (bytes)=0,
Last Modified=09Jun2015:15:47:58,
```

Create Time=09Jun2015:15:47:58
NOTE: The SAS System stopped processing this step because of errors.

Reply

Chris Hemedinger
Posted June 9, 2015 at 4:32 pm | *Permalink*

I get that message if the file doesn't exist (`c:\users\pmiller\documents\be_output\be_output.zip`). Check to make sure that's the correct file name?

Reply

Andrei Mitiaev
Posted June 9, 2015 at 4:29 pm | *Permalink*

Will it be working on Unix as well?
Thanks,
Andrei Mitiaev

Reply

Chris Hemedinger
Posted June 9, 2015 at 4:30 pm | *Permalink*

Yes, it will work on all of the Unix variations.

Reply

Janice
Posted November 12, 2015 at 2:28 am | *Permalink*

Hi Chris,
Thanks for the blog, I tried your method to read and unzip a gz extension file (myfile.gz), but with no success. This is what I tried :

```
/*******************************************************************************/
filename test zip "c:\mydirectory\myfile.gz" ;
filename outtest "C:\Janicedirectory";
data _null_ ;
```

```
infile test recfm=N ;
file outtest recfm=N;
input byte $char1. ;
put byte $char1. ;
run;
```

Reply

Chris Hemedinger

Posted November 12, 2015 at 9:24 am | *Permalink*

Did you post a complete example? Try this for a start to see what's in the GZ file:

```
filename inzip ZIP "c:\mydirectory\myfile.gz";
/* Read the "members" (files) from the ZIP file */
data contents(keep=memname isFolder);
length memname $200 isFolder 8;
fid=dopen("inzip");
if fid=0 then
stop;
memcount=dnum(fid);
do i=1 to memcount;
memname=dread(fid,i);
/* check for trailing / in folder name */
isFolder = (first(reverse(trim(memname)))='/');
output;
end;
rc=dclose(fid);
run;
/* create a report of the ZIP contents */
title "Files in the GZ file";
proc print data=contents noobs N;
run;
```

Reply

janice

Posted November 12, 2015 at 3:13 pm | *Permalink*

Actually , I have just a single compressed file «myfile.dat.gz». so I need to read and process the data in sas 9.4 from this compressed file. I try to read

data from the zipped file by byte and output it to an external file , here is my code :
filename test zip "C:\mydirectory\myfile.dat.gz" member='myfile.dat.gz';
filename outtest "C:\Janicedirectory\myfile.dat";
data _null_ ;
Infile test lrecl= 256 recfm=N ;
File outtest lrecl=265 recfm=N ; /* output file*/
input ;
put _infile_ ;
run;
Still doesn't work ?

Reply

Chris Hemedinger

Posted November 12, 2015 at 5:19 pm | *Permalink*

Almost there, I think. Try something like this (removing the .gz from the member= option):

```
/* assuming file in archive is named "myfile.dat" */
filename test zip "C:\mydirectory\myfile.dat.gz"
member='myfile.dat';
```

Reply

Amin

Posted December 17, 2015 at 11:23 am | *Permalink*

Are we assuming that the contents of the zip file are as XLSX? What if the data is a CSV?

Reply

Chris Hemedinger

Posted December 17, 2015 at 11:27 am | *Permalink*

I used XLSX and a SAS7BDAT file as examples, but CSV would work the same way. Use FILENAME ZIP to "address" the file, DATA step to copy it out as file block, then another DATA step to read it. You might be able to combine those two DATA steps to read the file contents just once, but I don't

think you'll be able to INFILE the item as series of text characters while it's in the ZIP archive.

Reply

keith johnson

Posted December 23, 2015 at 5:58 pm | *Permalink*

HI I am not getting the second part to work where you read in a file. You wrote: "With this information, I can now 'copy" the XLSX file out of the ZIP file and then import it into a SAS data set. Notice how I can use the "member" syntax (fileref with the file I want in parentheses) to address a specific file in the ZIP archive. I want to copy just from the actual files, and not the folder-level entries.' I am not sure but are you manually opening the zip file to put it in the work folder? I have an assignment to work with and find metadata on my entire 2TB network share and have a count of multiple K zip files to open and wanted to make sure this process could auto read, and extract the data without manually opening the files. This code: filename inzip ZIP
"D:\MyFileSystem\This_zipped_file.zip";
/* Read the "members" (files) from the ZIP file */
data contents(keep=memname isFolder);
length memname $200 isFolder 8;
fid=dopen("inzip");
if fid=0 then
stop;
memcount=dnum(fid);
do i=1 to memcount;
memname=dread(fid,i);
/* check for trailing / in folder name */
isFolder = (first(reverse(trim(memname)))='/');
output;
end;
rc=dclose(fid);
run;
/* create a report of the ZIP contents */
title "Files in the ZIP file";
proc print data=contents noobs N;
where isFolder =0;
VAR MEMNAME;
run;
/*
makes this output:

Files in the ZIP file 83
15:58 Tuesday, December 22, 2015
memname
CSV/CollectorList.csv
CSV/Sheet_1.csv
Excel/CollectorList.xls
Excel/Sheet_1.xls
READ ME.txt
N = 5
and for example I have tried these lines of code and a few variations but it does not work...
*/
filename xl "D:\MyFileSystem\Excel\CollectorList.xls" ;
/* hat tip: "data _null_" on SAS-L */
data _null_;
/* using member syntax here */
infile inzip(Excel\CollectorList.xls)
/*... the rest of your code here..*/
any clues?

Reply

keith johnson

Posted December 23, 2015 at 6:13 pm | *Permalink*

...darn, I just tired my code again after manually extracting the folder and placing it in my path. I now works. Not what I was hoping it to do... :(Thank you for sharing. -KJ

Reply

Chris Hemedinger

Posted December 30, 2015 at 2:22 pm | *Permalink*

Keith, my example is supposed to work without you having to manually unzip/extract the ZIP file -- everything should be handled by the SAS program using the FILENAME ZIP method. Is that what you have working? If not, let me know.

Reply

Matt

Posted January 27, 2016 at 5:03 pm | *Permalink*

I'm having it hang up on the eof statement. I'm working with linux SAS. Suggestions would be appreciated.

```
36 filename ds "%sysfunc(getoption(work))/mic_file.sas7bdat";
37 data _null_;
38 /*using member syntax here*/
39 infile inzip(mic_file.sas7bdat) lrecl=256 recfm=F
39 ! length=length eof=eof unbuf;
40 file ds lrecl=256 recfm=N;
41 input;
42 put _infile_ $varying256. length;
43 return;
44 eof;
___
180
ERROR 180-322: Statement is not valid or it is used out of proper
order.
45 stop;
46 run;
```

Reply

Chris Hemedinger

Posted January 27, 2016 at 5:09 pm | *Permalink*

In this example, "eof" on line 44 is a label, a target to "goto". Use a **colon** instead of a semicolon after it.

Reply

Keith Johnson

Posted March 3, 2016 at 6:29 pm | *Permalink*

I am back on this project after a few months on other things and now, I am lost again with your sample code. I thought I had it figured out in Dec. In my case I have cloned our file systems with a robocopy command to a sub directory placed at the root level, I used a few excludes to not get any race conditions etc. I then added *.zip files to my robocopy

command. I then found a purge empty directory script cut off the excess limbs. Now I have a trimmed down directory with only a copy of the original zip files that I can use as I see fit, my next idea was to extract them all to a relative folder of the location but instead of myfile.zip I would place all the data in myfile.X folder again relative to each *.zip file found. So I have written the SAS code to make the subdir's and move the files to their new home. Now I need to extract all my files. My problem, and reason I found your post. Once I am done with my extract I have 7-or-8 SAS programs I can clone and point at this sub directly and scan for metadata (xls, xlsx, mdb, accdb, dbf, sas7bdat, and sav) and produce a report as I am required. But as stated am stuck back on the extract phase.
TIA for any pointers you can provide.
…I am just looking for a stripped down version of what you seem to be showing that does not bother with reading the meta data directly of the files found, but just extract them and if possible read from a dataset with one field like MyPathFile where the strings might be: E:\mypath1\mypath2\...\mypathN\MyFile.X\MyFile.zip
E:\mypath1\mypath2\...\mypathN\MyFile1.X\MyFile1.zip
E:\mypath1\mypath2\...\mypathN\MyFile3.X\MyFile3.zip
E:\mypath1\...\mypathN\MyFile.X\MyFile.zip
E:\MyFile.X\MyFile.zip
But I can easily split it in to path and file if needed.
-TIA -KJ

Reply

Chris Hemedinger

Posted March 4, 2016 at 9:56 am | *Permalink*

Keith,
For now, I'll have to leave this as an exercise for you -- or you can post your question and code-so-far to SAS Support Communities and perhaps someone else can help.
I can offer these tips though:
- use the dopen and dread functions to find the names/paths of the files inside the ZIP archive (as in my example in this post).
- using that information, create a fileref for each member you want to extract, then use a binary copy method (in DATA step) to copy the byte stream of an archive member to a destination fileref in your target folder. This step may be suited to a macro or to a DATA step loop or even a DOSUBL construct.
I have another FILENAME ZIP blog post teed up to publish soon, so stay tuned for more related information.

Reply

One Trackback

1. By Add files to a ZIP archive with FILENAME ZIP - The SAS Dummy on March 4, 2016 at 5:02 pm

Post a Comment

Your email is *never* published nor shared. Required fields are marked *

Name *

Email *

Website

Comment *

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <s> <strike> <strong>`

POST COMMENT

blogs.sas.com

BUSINESS LEADERSHIP

SAS Voices
News and views from the people who make SAS a great place to work

Customer Intelligence
Evolving relationships for business growth

Value Alley
Your pathway from strategy to process to repeatable value creation

GETTING TECHNICAL

The SAS Dummy
A SAS® blog for the rest of us

The DO Loop
Statistical programming in SAS with an emphasis on SAS/IML programs

Operations Research with SAS
Optimize, Simulate, Understand

JMP Blog
Data visualization, statistical discovery, design of experiments, predictive modeling and more

SAS Learning Post
Technical tips and tricks from SAS instructors, authors and other SAS experts.

Graphically Speaking
Data Visualization with a focus on SAS ODS Graphics

SAS Users
Providing technical tips and support information, written for and by SAS users.

ANALYTICS IN ACTION

Subconscious Musings
Advanced analytics - from Research Drive to the world

The Text Frontier
Text mining, voice mining and unstructured data analysis

The Business Forecasting Deal
Exposing bad practices and offering practical solutions in business forecasting

The Data Roundtable
A community of data management experts

COUNTRIES AND REGIONS

Hidden Insights
Experience the possibilities with Business Analytics in the Nordics

Klog på SAS
Tips og tricks til effektiv SAS-programmering

Mehr Wissen
Big Data Analytics in Deutschland, Österreich und der Schweiz

SAS Latin America
Business Analytics: Transforme su Negocio: Convierta sus datos en Conocimiento y Decisiones Inteligentes.

INDUSTRY INSIGHTS

The Analytic Insurer
Solving your customer, risk, fraud and operational challenges in insurance

A Shot in the Arm
Transforming quality, cost, and outcomes in the healthcare ecosystem

Generation SAS
Resources and tips for students and educators

State and Local Connection
State and local governments using data to serve citizens and save money

The Analytic Hospitality Executive
Finding analytic solutions to your forecasting, pricing and operational challenges.

Pathfinders
Exploring SAS Curriculum Pathways & Instructional Technology

Made in… Analytics
Your guide for improving manufacturing outcomes with advanced analytics