

The SAS Dummy

A SAS® blog for the rest of us

Add files to a ZIP archive with FILENAME ZIP



Chris Hemedinger | MARCH 4, 2016

368 1



Share

1

In previous articles, I've shared tips about how you can work with SAS and ZIP files without requiring an external tool like WinZip, gzip, or 7-Zip. I've covered:

- How to create ZIP files with ODS PACKAGE ZIP (available since SAS 9.2)
- [How to "unzip" and read ZIP files using FILENAME ZIP](#) (SAS 9.4 and later)
- How to create and update ZIP files with FILENAME ZIP (SAS 9.4 and later)

But a customer approached me the other day with one scenario I missed: how to add SAS data sets to an existing ZIP file. It's a variation of a tip that I've already shared, but with two differences. First, in order to add a data set to a ZIP file, you have to know its physical filename -- not just the LIBNAME.MEMBER reference that you use in SAS procedure steps. And second, I had not shown how to **add a new file** to an existing ZIP archive -- though it turns out that's pretty simple.

Find the file name for a SAS data set

There are several ways to do this. For my approach, I used the output from PROC CONTENTS. Notice that I had to capture the ODS output (not the OUT= data set) to grab the file name. I wrapped it in a macro for easy reuse. And since I ultimately need a SAS fileref to map to the path, I've assigned one (**data_fn**) in my macro.

```
/* macro to assign a fileref to a SAS data set in a Base Library */  
%macro assignFilerefToDataset(_dataset_name);  
  %local outDsName;
```

```
ods output EngineHost=File;
proc contents data=&_dataset_name.;
run;
proc sql noprint;
    select cValue1 into: outDsName
        from work.file where Label1="Filename";
quit;
filename data_fn "&outDsName.";
%mend;
```

How to add a new member to a ZIP file

Now that I have the source file, I need to designate a destination file in a ZIP archive. The FILENAME ZIP method will create a new ZIP file if one does not yet exist, or it can add to an existing ZIP. To ensure I'm starting from scratch, I assign a simple fileref to my target destination and then delete the file.

```
/* Assign the fileref - basic file method */
filename projzip "&projectDir./project.zip";
/* Start with a clean slate - delete ZIP if it exists */
data _null_;
    rc=fdelete('projzip');
run;
```

To create a new ZIP file and designate a path and file name within it, I used the FILENAME ZIP method with the MEMBER= option. Note that I specified the "data/" subfolder in the MEMBER= value; this will place the file into a named subfolder within the archive.

```
/* Use FILENAME ZIP to add a new member -- CLASS */
/* Put it in the data subfolder */
filename addfile zip "&projectDir./project.zip"
    member='data/class.sas7bdat';
```

Then finally, I need to actually "copy" the file into the archive. I do this by streaming the source file into the target fileref byte-by-byte:

```
/* byte-by-byte copy */
/* "copies" the new file into the ZIP archive */
data _null_;
    infile data_fn recfm=n;
```

```
file addfile recfm=n;  
input byte $char1. @;  
put byte $char1. @;  
run;  
  
filename addfile clear;
```

That's it! I now have a ZIP file with one member entry. Now I can "press repeat" to add a second entry:

```
%assignFilerefToDataset(sashelp.cars);  
/* Use FILENAME ZIP to add a new member -- CARS */  
/* Put it in the data subfolder */  
filename addfile zip "&projectDir./project.zip"  
    member='data/cars.sas7bdat';  
/* byte-by-byte copy */  
/* "copies" the new file into the ZIP archive */  
data _null_;  
    infile data_fn recfm=n;  
    file addfile recfm=n;  
    input byte $char1. @;  
    put byte $char1. @;  
run;  
  
filename addfile clear;
```

Optional: Report on the ZIP file contents

If I want to report on the total contents of the ZIP file now, here's a DATA step and PROC CONTENTS step that does the job:

```
/* OPTIONAL for reporting */  
/* Report on the contents of the ZIP file */  
/* Assign a fileref with the ZIP method */  
filename inzip zip "&projectDir./project.zip";  
/* Read the "members" (files) from the ZIP file */  
data contents(keep=memname);  
    length memname $200;  
    fid=dopen("inzip");  
    if fid=0 then  
        stop;  
    memcount=dnum(fid);  
    do i=1 to memcount;  
        memname=dread(fid,i);  
        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;
```

Result:

```
Files in the ZIP file  
  
memname  
-----  
data/class.sas7bdat  
data/cars.sas7bdat  
N = 2
```

I hope that this helps to make the FILENAME ZIP method more useful to those who want to try it out. I'm sure that there will be more scenarios that people will ask about; someday, if I write enough blog posts, I'll have it all covered!

Sample program: You can view/download the entire SAS program (containing the snippets I've featured and more) [from my GitHub profile](#).

tags: [FILENAME ZIP](#), [SAS 9.4](#), [SAS programming](#), [ZIP files](#)

One Comment

Gopal

Posted March 4, 2016 at 5:34 pm | [Permalink](#)

Reply

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

Name *

Email *

Website

Comment*

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

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

The blog content appearing on this site does not necessarily represent the opinions of SAS. Your use of this blog is governed by the [Terms of Use](#) and the [SAS Privacy Statement](#).

Copyright © SAS Institute Inc. All Rights Reserved