

# m\_cst\_unpack\_zip.sas File Reference

## Custom

Custom macro to extract files and directories from a ZIP file

---

### Description

This program extracts all files from a given ZIP archive to a given target directory preserving the directory structure that is contained in the ZIP archive. The macro uses byte-by-byte including dynamic chunksize copying to extract files from the given ZIP archive, delivering high performance ZIP extraction. The finfo option for Zip archive members is based on the blog article "Using FILENAME ZIP and FINFO to list the details in your ZIP files" by Chris Hemedinger (chris.hemedinger@sas.com). The chunksize option that is used in this macro is based on the binaryfilecopy by Bruno Mueller (bruno.mueller@sas.com).

### Note

*This program is able to work in system environments where x-command or unix pipes are not allowed or cannot be used.*

### Autors

Paul Alexander Canals y Trocha (paul.canals@gmail.com)

### Date

2023-09-11 00:00:00

### Version

23.1.09

### Link

<https://github.com/paul-canals/toolbox>

## Parameters

Input	help	Parameter, if set (Help or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	infile	Full qualified path and file name of the archive to be unpacked. The default value is: <code>_NONE_</code> .
Input	outdir	Full qualified path to the target directory where the files and directory structure from the archive are to be unpacked. The default value is: <code>_NONE_</code> .
Input	runmode	Indicator [E V] to specify whether the macro uses the (E)xtract or the (V)iew function mode. The view mode does only list all files and directory structure contained by the ZIP archive, where as the extract mode unpacks all the ZIP contents. The default value for RUNMODE is: E (extract).
Input	print	Boolean [Y N] parameter to generate the output by a proc print step. The default value is N.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

## Returns

- The program returns unpacked files from a ZIP archive

## Calls

- [None](#)

## Usage

Example 1: Show help information:

```
%m_cst_unpack_zip(?)
```

Example 2: View all files listed in a ZIP archive (mode: View):

```
/* Create ZIP archive: ;
filename tmpfile "%sysget(SASROOT)/core/sashelp/cars.sas7bdat";
filename zipfile zip "%sysfunc(getoption(WORK))/sashelp.zip" member="cars.sas7bdat";

/* byte-by-byte copy ;
data _null_;
  infile tmpfile recfm=n;
  file zipfile recfm=n;
  input byte $char1. @;
  put byte $char1. @;
run;

filename tmpfile "%sysget(SASROOT)/core/sashelp/class.sas7bdat";
filename zipfile zip "%sysfunc(getoption(WORK))/sashelp.zip" member="class.sas7bdat";

/* byte-by-byte copy ;
data _null_;
  infile tmpfile recfm=n;
  file zipfile recfm=n;
  input byte $char1. @;
  put byte $char1. @;
run;

filename tmpfile clear;
filename zipfile clear;

/* view ZIP contents ;
%m_cst_unpack_zip(
  infile = %sysfunc(getoption(WORK))/sashelp.zip
  , runmode = VIEW
  , print = Y
  , debug = Y
  );
```

Example 3: Extract all files listed in a ZIP archive (mode: Extract):

```

%* Create ZIP archive: ;
filename tmpfile "%sysget(SASROOT)/core/sashelp/cars.sas7bdat";
filename zipfile zip "%sysfunc(getoption(WORK))/sashelp.zip" member="cars.sas7bdat";

%* byte-by-byte copy ;
data _null_;
    infile tmpfile recfm=n;
    file zipfile recfm=n;
    input byte $char1. @;
    put byte $char1. @;
run;

filename tmpfile "%sysget(SASROOT)/core/sashelp/class.sas7bdat";
filename zipfile zip "%sysfunc(getoption(WORK))/sashelp.zip" member="class.sas7bdat";

%* byte-by-byte copy ;
data _null_;
    infile tmpfile recfm=n;
    file zipfile recfm=n;
    input byte $char1. @;
    put byte $char1. @;
run;

filename tmpfile clear;
filename zipfile clear;

%* extract ZIP contents ;
%m_cst_unpack_zip(
    infile = %sysfunc(getoption(WORK))/sashelp.zip
    , outdir = %sysfunc(getoption(WORK))/sashelp
    , runmode = EXTRACT
    , print = Y
    , debug = Y
);

```

## **Copyright**

Copyright 2008-2023 Paul Alexander Canals y Trocha.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.