

NAME

streamzip – create a zip file from stdin

SYNOPSIS

```
producer | streamzip [opts] | consumer
producer | streamzip [opts] -zipfile=output.zip
```

DESCRIPTION

This program will read data from `stdin`, compress it into a zip container and, by default, write a *streamed* zip file to `stdout`. No temporary files are created.

The zip container written to `stdout` is, by necessity, written in streaming format. Most programs that read Zip files can cope with a streamed zip file, but if interoperability is important, and your workflow allows you to write the zip file directly to disk you can create a non-streamed zip file using the `zipfile` option.

OPTIONS**-zip64**

Create a Zip64-compliant zip container. Use this option if the input is greater than 4Gig.

Default is disabled.

-zipfile=F

Write zip container to the filename F.

Use the `Stream` option to force the creation of a streamed zip file.

-member-name=M

This option is used to name the “file” in the zip container.

Default is `'-'`.

-stream

Ignored when writing to `stdout`.

If the `zipfile` option is specified, including this option will trigger the creation of a streamed zip file.

Default: Always enabled when writing to `stdout`, otherwise disabled.

-method=M

Compress using method M.

Valid method names are

```
* store      Store without compression
* deflate    Use Deflate compression [Default]
* bzip2      Use Bzip2 compression
* lzma       Use LZMA compression
* xz         Use xz compression
* zstd       Use Zstandard compression
```

Note that Lzma compress needs `IO::Compress::Lzma` to be installed.

Note that Zstd compress needs `IO::Compress::Zstd` to be installed.

Default is `deflate`.

-0, -1, -2, -3, -4, -5, -6, -7, -8, -9

Sets the compression level for `deflate`. Ignored for all other compression methods.

-0 means no compression and -9 for maximum compression.

Default is 6

```

-version
    Display version number

-help
    Display help

```

Examples

Create a zip file by reading data from stdin

```
$ echo Lorem ipsum dolor sit | perl ./bin/streamzip >abcd.zip
```

Check the contents of `abcd.zip` with the standard `unzip` utility

```

Archive:  abcd.zip
  Length      Date    Time    Name
-----
          22  2021-01-08 19:45    -
-----
          22                          1 file

```

Notice how the Name is set to `-`. That is the default for a few zip utilities where the member name is not given.

If you want to explicitly name the file, use the `-member-name` option as follows

```

$ echo Lorem ipsum dolor sit | perl ./bin/streamzip -member-name latin >abcd.zip

$ unzip -l abcd.zip
Archive:  abcd.zip
  Length      Date    Time    Name
-----
          22  2021-01-08 19:47    latin
-----
          22                          1 file

```

When to write a Streamed Zip File

A Streamed Zip File is useful in situations where you cannot seek backwards/forwards in the file.

A good example is when you are serving dynamic content from a Web Server straight into a socket without needing to create a temporary zip file in the filesystem.

Similarly if your workflow uses a Linux pipelined commands.

SUPPORT

General feedback/questions/bug reports should be sent to <https://github.com/pmq/IO-Compress/issues> (preferred) or <https://rt.cpan.org/Public/Dist/Display.html?Name=IO-Compress>.

AUTHOR

Paul Marquess pmqs@cpan.org.

COPYRIGHT

Copyright (c) 2019–2021 Paul Marquess. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.