

NAME

ziptool - modify zip archives

SYNOPSIS

ziptool [-ceghnrst] [-l *length*] [-o *offset*] *zip-archive* **command** [*command-args* ...]
[**command** [*command-args* ...] ...]

DESCRIPTION

ziptool modifies the zip archive *zip-archive* according to the *commands* given.

Supported options:

- c** Check zip archive consistency when opening it.
- e** Error if archive already exists (only useful with **-n**).
- g** Guess file name encoding (for **stat** command).
- h** Display help.
- l *length*** Only read *length* bytes of archive. See also **-o**.
- n** Create archive if it doesn't exist. See also **-e**.
- o *offset*** Start reading input archive from *offset*. See also **-l**.
- r** Print raw file name encoding without translation (for **stat** command).
- s** Follow file name convention strictly (for **stat** command).
- t** Disregard current file contents, if any. *Note*: use this with care, it deletes all existing file contents when you modify the archive.

Commands

For all commands below, the index is zero-based. In other words, the first entry in the zip archive has index 0.

Supported commands and arguments are:

add *name content*

Add file called *name* using the string *content* from the command line as data.

add_dir *name*

Add directory *name*.

add_file *name file_to_add offset len*

Add file *name* to archive, using *len* bytes from the file *file_to_add* as input data, starting at *offset*.

add_from_zip *name archivename index offset len*

Add file called *name* to archive using data from another zip archive *archivename* using the entry with index *index* and reading *len* bytes from *offset*.

cat *index* Output file contents for entry *index* to stdout.

count_extra *index flags*

Print the number of extra fields for archive entry *index* using *flags*.

count_extra_by_id *index extra_id flags*

Print number of extra fields of type *extra_id* for archive entry *index* using *flags*.

delete *index* Remove entry at *index* from zip archive.

delete_extra *index extra_idx flags*

Remove extra field number *extra_idx* from archive entry *index* using *flags*.

delete_extra_by_id *index extra_id extra_index flags*

Remove extra field number *extra_index* of type *extra_id* from archive entry *index* using *flags*.

get_archive_comment

Print archive comment.

get_archive_flag *flag*

Print state of archive flag *flag*.

get_extra *index extra_index flags*

Print extra field *extra_index* for archive entry *index* using *flags*.

get_extra_by_id *index extra_id extra_index flags*

Print extra field *extra_index* of type *extra_id* for archive entry *index* using *flags*.

get_file_comment *index*

Get file comment for archive entry *index*.

get_num_entries *flags*

Print number of entries in archive using *flags*.

name_locate *name flags*

Find entry in archive with the filename *name* using *flags* and print its index.

rename *index name*

Rename archive entry *index* to *name*.

replace_file_contents *index data*

Replace file contents for archive entry *index* with the string *data*.

set_archive_comment *comment*

Set archive comment to *comment*.

get_archive_flag *flag value*

Set archive flag *flag* to *value*.

set_extra *index extra_id extra_index flags value*

Set extra field number *extra_index* of type *extra_id* for archive entry *index* using *flags* to *value*.

set_file_comment *index comment*

Set file comment for archive entry *index* to string *comment*.

set_file_compression *index method compression_flags*

Set file compression method for archive entry *index* to *method* using *compression_flags*.

Note: Currently, *compression_flags* are ignored.

set_file_encryption *index method password*

Set file encryption method for archive entry *index* to *method* with password *password*.

set_file_mtime *index timestamp*

Set file modification time for archive entry *index* to UNIX mtime *timestamp*.

set_file_mtime_all *timestamp*

Set file modification time for all archive entries to UNIX mtime *timestamp*.

set_password *password*

Set default password for encryption/decryption to *password*.

stat *index* Print information about archive entry *index*.

Flags

Some commands take flag arguments. Each character in the argument sets the corresponding flag. Use 0 or the empty string for no flags.

Supported flags are:

<i>4</i>	ZIP_FL_ENC_CP437
<i>8</i>	ZIP_FL_ENC_UTF_8
<i>C</i>	ZIP_FL_NOCASE
<i>c</i>	ZIP_FL_CENTRAL
<i>d</i>	ZIP_FL_NODIR
<i>l</i>	ZIP_FL_LOCAL
<i>r</i>	ZIP_FL_ENC_RAW
<i>s</i>	ZIP_FL_ENC_STRICT
<i>u</i>	ZIP_FL_UNCHANGED

Archive flags

get_archive_flag and **set_archive_flag** work on the following flags:

- ⊕ create-or-keep-empty-file-for-archive
- ⊕ is-torrentzip
- ⊕ ronly
- ⊕ want-torrentzip

Compression Methods

Some commands take compression method arguments. Supported methods are:

- ⊕ default
- ⊕ deflate
- ⊕ store

Encryption Methods

Some commands take encryption method arguments. Supported methods are:

- ⊕ none
- ⊕ AES-128

- AES-192
- AES-256

EXIT STATUS

The **ziptool** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Add a file called *teststring.txt* to the zip archive *testbuffer.zip* with data "This is a test.\n" where "\n" is replaced with a newline character:

```
ziptool testbuffer.zip add teststring.txt \"This is a test.\n\"
```

Delete the first file from the zip archive *testfile.zip*:

```
ziptool testfile.zip delete 0
```

SEE ALSO

zipcmp(1), zipmerge(1), libzip(3)

HISTORY

ziptool was added in libzip 1.1.

AUTHORS

Dieter Baron <*dillo@nih.at*> and Thomas Klausner <*tk@giga.or.at*>