
GZip.jl Documentation

Release 0.1

Kevin Squire

Apr 19, 2017

Contents

1	Notes	3
2	Functions	5
3	Types	7
	Python Module Index	9

This module provides a wrapper for the gzip related functions of ([zlib](#)), a free, general-purpose, legally unencumbered, lossless data-compression library. These functions allow the reading and writing of gzip files.

- This interface is only for gzipped files, not the streaming zlib compression interface. Internally, it depends on/uses the streaming interface, but the gzip related functions are higher level functions pertaining to gzip files only.
- *GZipStream* is an implementation of `IO` and can be used virtually anywhere `IO` is used.
- This implementation mimics the `IOStream` implementation, and should be a drop-in replacement for `IOStream`, with some caveats:
 - `seekend()` and `truncate()` are not available
 - `readuntil()` is available, but is not very efficient. (But `readline()` works fine.)

In addition to *`open()`*, *`gzopen()`* and *`gzdopen()`*, the following `IO/IOStream` functions are supported:

- `close()`
- `flush()`
- `seek()`
- `skip()`
- `position()`
- `eof()`
- `read()`
- `readuntil()`
- `readline()`
- `write()`
- `peek()`

Due to limitations in `zlib`, `seekend()` and `truncate()` are not available.

`GZip.open (fname[, gzmode[, buf_size ]])`

Alias for `gzopen()`. This is not exported, and must be called using `GZip.open()`.

`GZip.gzopen (fname[, gzmode[, buf_size ]])`

Opens a file with mode (default "r"), setting internal buffer size to `buf_size` (default `Z_DEFAULT_BUFSIZE=8192`), and returns a the file as a `GZipStream`.

`gzmode` must contain one of

r	read
w	write, create, truncate
a	write, create, append

In addition, `gzmode` may also contain

x	create the file exclusively (fails if file exists)
0-9	compression level

and/or a compression strategy:

f	filtered data
h	Huffman-only compression
R	run-length encoding
F	fixed code compression

Note that `+` is not allowed in `gzmode`.

If an error occurs, `gzopen` throws a `GZError`

`GZip.gzdopen (fd[, gzmode[, buf_size ]])`

Create a `GZipStream` object from an integer file descriptor. See `gzopen()` for `gzmode` and `buf_size` descriptions.

`GZip.gzdopen (s[, gzmode[, buf_size ]])`

Create a `GZipStream` object from `IOStream s`.

CHAPTER 3

Types

type `GZip.GZipStream` (*name*, *gz_file*`[`, *buf_size*`[`, *fd*`[`, *s*`]``]`)

Subtype of `IO` which wraps a gzip stream. Returned by `gzopen()` and `gzdopen()`.

type `GZip.GZError` (*err*, *err_str*)

gzip error number and string. Possible error values:

<code>Z_OK</code>	No error
<code>Z_ERRNO</code>	Filesystem error (consult <code>errno()</code>)
<code>Z_STREAM_ERROR</code>	Inconsistent stream state
<code>Z_DATA_ERROR</code>	Compressed data error
<code>Z_MEM_ERROR</code>	Out of memory
<code>Z_BUF_ERROR</code>	Input buffer full/output buffer empty
<code>Z_VERSION_ERROR</code>	zlib library version is incompatible with caller version

g

GZip, 3

G

`gzdopen()` (in module `GZip`), [5](#)

`GZip` (module), [1](#)

`gzopen()` (in module `GZip`), [5](#)

O

`open()` (in module `GZip`), [5](#)