

:mod:`zlib` --- Compression compatible with

:program:`gzip`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 1); [backlink](#)

Unknown interpreted text role "program".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 4)

Unknown directive type "module".

```
.. module:: zlib
   :synopsis: Low-level interface to compression and decompression routines
             compatible with gzip.
```

For applications that require data compression, the functions in this module allow compression and decompression, using the zlib library. The zlib library has its own home page at <https://www.zlib.net>. There are known incompatibilities between the Python module and versions of the zlib library earlier than 1.1.3; 1.1.3 has a [security vulnerability](#), so we recommend using 1.1.4 or later.

zlib's functions have many options and often need to be used in a particular order. This documentation doesn't attempt to cover all of the permutations; consult the zlib manual at <http://www.zlib.net/manual.html> for authoritative information.

For reading and writing .gz files see the `mod:`gzip`` module.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 22); [backlink](#)

Unknown interpreted text role "mod".

The available exception and functions in this module are:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 27)

Unknown directive type "exception".

```
.. exception:: error

   Exception raised on compression and decompression errors.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 32)

Unknown directive type "function".

```
.. function:: Adler32(data[, value])

   Computes an Adler-32 checksum of *data*. (An Adler-32 checksum is almost as
   reliable as a CRC32 but can be computed much more quickly.) The result
   is an unsigned 32-bit integer. If *value* is present, it is used as
   the starting value of the checksum; otherwise, a default value of 1
   is used. Passing in *value* allows computing a running checksum over the
   concatenation of several inputs. The algorithm is not cryptographically
   strong, and should not be used for authentication or digital signatures. Since
   the algorithm is designed for use as a checksum algorithm, it is not suitable
   for use as a general hash algorithm.

.. versionchanged:: 3.0
   The result is always unsigned.
   To generate the same numeric value when using Python 2 or earlier,
   use ``adler32(data) & 0xffffffff``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 49)

Unknown directive type "function".

```
.. function:: compress(data, /, level=-1, wbits=MAX_WBITS)

   Compresses the bytes in *data*, returning a bytes object containing compressed data.
   *level* is an integer from ``0`` to ``9`` or ``-1`` controlling the level of compression;
   ``1`` (Z_BEST_SPEED) is fastest and produces the least compression, ``9`` (Z_BEST_COMPRESSION)
   is slowest and produces the most. ``0`` (Z_NO_COMPRESSION) is no compression.
   The default value is ``-1`` (Z_DEFAULT_COMPRESSION). Z_DEFAULT_COMPRESSION represents a default
   compromise between speed and compression (currently equivalent to level 6).

.. _compress-wbits:

   The *wbits* argument controls the size of the history buffer (or the
   "window size") used when compressing data, and whether a header and
   trailer is included in the output. It can take several ranges of values,
```

```

defaulting to ``15`` (MAX_WBITS):

* +9 to +15: The base-two logarithm of the window size, which
  therefore ranges between 512 and 32768. Larger values produce
  better compression at the expense of greater memory usage. The
  resulting output will include a zlib-specific header and trailer.

* â`9 to â`15: Uses the absolute value of *wbits* as the
  window size logarithm, while producing a raw output stream with no
  header or trailing checksum.

* +25 to +31 = 16 + (9 to 15): Uses the low 4 bits of the value as the
  window size logarithm, while including a basic :program:`gzip` header
  and trailing checksum in the output.

Raises the :exc:`error` exception if any error occurs.

.. versionchanged:: 3.6
   *level* can now be used as a keyword parameter.

.. versionchanged:: 3.11
   The *wbits* parameter is now available to set window bits and
   compression type.

```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 87)

Unknown directive type "function".

```

.. function:: compressobj(level=-1, method=DEFLATED, wbits=MAX_WBITS, memLevel=DEF_MEM_LEVEL, strategy=Z_DEFAULT_S

```

Returns a compression object, to be used for compressing data streams that won't fit into memory at once.

level is the compression level -- an integer from ``0`` to ``9`` or ``-1``. A value of ``1`` (Z BEST SPEED) is fastest and produces the least compression, while a value of ``9`` (Z BEST_COMPRESSION) is slowest and produces the most. ``0`` (Z_NO_COMPRESSION) is no compression. The default value is ``-1`` (Z_DEFAULT_COMPRESSION). Z_DEFAULT_COMPRESSION represents a default compromise between speed and compression (currently equivalent to level 6).

method is the compression algorithm. Currently, the only supported value is :const:`DEFLATED`.

The *wbits* parameter controls the size of the history buffer (or the "window size"), and what header and trailer format will be used. It has the same meaning as `described for compress() <#compress-wbits>`_.

The *memLevel* argument controls the amount of memory used for the internal compression state. Valid values range from ``1`` to ``9``. Higher values use more memory, but are faster and produce smaller output.

strategy is used to tune the compression algorithm. Possible values are :const:`Z_DEFAULT_STRATEGY`, :const:`Z_FILTERED`, :const:`Z_HUFFMAN_ONLY`, :const:`Z_RLE` (zlib 1.2.0.1) and :const:`Z_FIXED` (zlib 1.2.2.2).

dict is a predefined compression dictionary. This is a sequence of bytes (such as a :class:`bytes` object) containing subsequences that are expected to occur frequently in the data that is to be compressed. Those subsequences that are expected to be most common should come at the end of the dictionary.

```

.. versionchanged:: 3.3
   Added the *dict* parameter and keyword argument support.

```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 123)

Unknown directive type "function".

```

.. function:: crc32(data[, value])

```

```

.. index::
   single: Cyclic Redundancy Check
   single: checksum; Cyclic Redundancy Check

```

Computes a CRC (Cyclic Redundancy Check) checksum of *data*. The result is an unsigned 32-bit integer. If *value* is present, it is used as the starting value of the checksum; otherwise, a default value of 0 is used. Passing in *value* allows computing a running checksum over the concatenation of several inputs. The algorithm is not cryptographically strong, and should not be used for authentication or digital signatures. Since the algorithm is designed for use as a checksum algorithm, it is not suitable for use as a general hash algorithm.

```

.. versionchanged:: 3.0
   The result is always unsigned.
   To generate the same numeric value when using Python 2 or earlier,
   use ``crc32(data) & 0xffffffff``.

```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 143)

Unknown directive type "function".

```

.. function:: decompress(data, /, wbits=MAX_WBITS, bufsize=DEF_BUF_SIZE)

```

Decompresses the bytes in *data*, returning a bytes object containing the uncompressed data. The *wbits* parameter depends on the format of *data*, and is discussed further below.

If `*bufsize*` is given, it is used as the initial size of the output buffer. Raises the `:exc:`error`` exception if any error occurs.

```
.. _decompress-wbits:
```

The `*wbits*` parameter controls the size of the history buffer (or "window size"), and what header and trailer format is expected. It is similar to the parameter for `:func:`compressobj``, but accepts more ranges of values:

- * +8 to +15: The base-two logarithm of the window size. The input must include a zlib header and trailer.
- * 0: Automatically determine the window size from the zlib header. Only supported since zlib 1.2.3.5.
- * \hat{a}^8 to \hat{a}^{15} : Uses the absolute value of `*wbits*` as the window size logarithm. The input must be a raw stream with no header or trailer.
- * +24 to +31 = 16 + (8 to 15): Uses the low 4 bits of the value as the window size logarithm. The input must include a gzip header and trailer.
- * +40 to +47 = 32 + (8 to 15): Uses the low 4 bits of the value as the window size logarithm, and automatically accepts either the zlib or gzip format.

When decompressing a stream, the window size must not be smaller than the size originally used to compress the stream; using a too-small value may result in an `:exc:`error`` exception. The default `*wbits*` value corresponds to the largest window size and requires a zlib header and trailer to be included.

`*bufsize*` is the initial size of the buffer used to hold decompressed data. If more space is required, the buffer size will be increased as needed, so you don't have to get this value exactly right; tuning it will only save a few calls to `:cfunc:`malloc``.

```
.. versionchanged:: 3.6
    *wbits* and *bufsize* can be used as keyword arguments.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 189)

Unknown directive type "function".

```
.. function:: decompressobj(wbits=MAX_WBITS[, zdict])
```

Returns a decompression object, to be used for decompressing data streams that won't fit into memory at once.

The `*wbits*` parameter controls the size of the history buffer (or the "window size"), and what header and trailer format is expected. It has the same meaning as described for `decompress()` `<#decompress-wbits>`__``.

The `*zdict*` parameter specifies a predefined compression dictionary. If provided, this must be the same dictionary as was used by the compressor that produced the data that is to be decompressed.

```
.. note::
```

If `*zdict*` is a mutable object (such as a `:class:`bytearray``), you must not modify its contents between the call to `:func:`decompressobj`` and the first call to the decompressor's `decompress()` method.

```
.. versionchanged:: 3.3
    Added the *zdict* parameter.
```

Compression objects support the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 215)

Unknown directive type "method".

```
.. method:: Compress.compress(data)
```

Compress `*data*`, returning a bytes object containing compressed data for at least part of the data in `*data*`. This data should be concatenated to the output produced by any preceding calls to the `:meth:`compress`` method. Some input may be kept in internal buffers for later processing.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 223)

Unknown directive type "method".

```
.. method:: Compress.flush([mode])
```

All pending input is processed, and a bytes object containing the remaining compressed output is returned. `*mode*` can be selected from the constants `:const:`Z_NO_FLUSH``, `:const:`Z_PARTIAL_FLUSH``, `:const:`Z_SYNC_FLUSH``, `:const:`Z_FULL_FLUSH``, `:const:`Z_BLOCK`` (zlib 1.2.3.4), or `:const:`Z_FINISH``, defaulting to `:const:`Z_FINISH``. Except `:const:`Z_FINISH``, all constants allow compressing further bytestrings of data, while `:const:`Z_FINISH`` finishes the compressed stream and prevents compressing any more data. After calling `:meth:`flush`` with `*mode*` set to `:const:`Z_FINISH``, the `:meth:`compress`` method cannot be called again; the only realistic action is to delete the object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 236)

Unknown directive type "method".

```
.. method:: Compress.copy()
```

Returns a copy of the compression object. This can be used to efficiently compress a set of data that share a common initial prefix.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 242)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8
```

Added :func:`copy.copy` and :func:`copy.deepcopy` support to compression objects.

Decompression objects support the following methods and attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 250)

Unknown directive type "attribute".

```
.. attribute:: Decompress.unused_data
```

A bytes object which contains any bytes past the end of the compressed data. That is, this remains ``b""`` until the last byte that contains compression data is available. If the whole bytestring turned out to contain compressed data, this is ``b""``, an empty bytes object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 258)

Unknown directive type "attribute".

```
.. attribute:: Decompress.unconsumed_tail
```

A bytes object that contains any data that was not consumed by the last :meth:`decompress` call because it exceeded the limit for the uncompressed data buffer. This data has not yet been seen by the zlib machinery, so you must feed it (possibly with further data concatenated to it) back to a subsequent :meth:`decompress` method call in order to get correct output.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 267)

Unknown directive type "attribute".

```
.. attribute:: Decompress.eof
```

A boolean indicating whether the end of the compressed data stream has been reached.

This makes it possible to distinguish between a properly-formed compressed stream, and an incomplete or truncated one.

```
.. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 278)

Unknown directive type "method".

```
.. method:: Decompress.decompress(data, max_length=0)
```

Decompress *data*, returning a bytes object containing the uncompressed data corresponding to at least part of the data in *string*. This data should be concatenated to the output produced by any preceding calls to the :meth:`decompress` method. Some of the input data may be preserved in internal buffers for later processing.

If the optional parameter *max_length* is non-zero then the return value will be no longer than *max_length*. This may mean that not all of the compressed input can be processed; and unconsumed data will be stored in the attribute :attr:`unconsumed_tail`. This bytestring must be passed to a subsequent call to :meth:`decompress` if decompression is to continue. If *max_length* is zero then the whole input is decompressed, and :attr:`unconsumed_tail` is empty.

```
.. versionchanged:: 3.6
```

max_length can be used as a keyword argument.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 297)

Unknown directive type "method".

```
.. method:: Decompress.flush([length])
```

All pending input is processed, and a bytes object containing the remaining uncompressed output is returned. After calling :meth:`flush`, the :meth:`decompress` method cannot be called again; the only realistic action is to delete the object.

The optional parameter **length** sets the initial size of the output buffer.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 307)

Unknown directive type "method".

```
.. method:: Decompress.copy()
```

Returns a copy of the decompression object. This can be used to save the state of the decompressor midway through the data stream in order to speed up random seeks into the stream at a future point.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 314)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8
   Added :func:`copy.copy` and :func:`copy.deepcopy` support to decompression
   objects.
```

Information about the version of the zlib library in use is available through the following constants:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 323)

Unknown directive type "data".

```
.. data:: ZLIB_VERSION
```

The version string of the zlib library that was used for building the module. This may be different from the zlib library actually used at runtime, which is available as :const:`ZLIB_RUNTIME_VERSION`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 330)

Unknown directive type "data".

```
.. data:: ZLIB_RUNTIME_VERSION
```

The version string of the zlib library actually loaded by the interpreter.

```
.. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) zlib.rst, line 337)

Unknown directive type "seealso".

```
.. seealso::
```

Module :mod:`gzip`
Reading and writing :program:`gzip` \ -format files.

<http://www.zlib.net>
The zlib library home page.

<http://www.zlib.net/manual.html>
The zlib manual explains the semantics and usage of the library's many functions.