**zlib 1.2.11**

January 15, 2017

Version 1.2.11 has these key improvements over 1.2.10:

* Fix deflate stored bug when pulling last block from window
* Permit immediate deflateParams changes before any deflate input

*Due to the bug fixes, any installations of 1.2.9 or 1.2.10 should be immediately replaced with 1.2.11.*

Version 1.2.10 has these key improvements over 1.2.9:

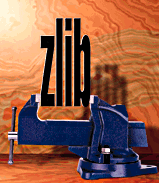
* Fix bug in deflate\_stored() for zero-length input
* Fix bug in gzwrite.c that produced corrupt gzip files

Version 1.2.9 has these key improvements over 1.2.8:

* Improve compress() and uncompress() to support large lengths
* Allow building zlib outside of the source directory
* Fix bug when level 0 used with Z\_HUFFMAN or Z\_RLE
* Fix bugs in creating a very large gzip header
* Add uncompress2() function, which returns the input size used
* Dramatically speed up deflation for level 0 (storing)
* Add gzfread() and gzfwrite(), duplicating the interfaces of fread() and fwrite()
* Add crc32\_z() and adler32\_z() functions with size\_t lengths
* Many portability improvements

You can also look at the complete [Change Log](https://www.zlib.net/ChangeLog.txt).

Canonical URL: [**http://zlib.net/**](http://zlib.net/) (US)

**zlib** is designed to be a [free](https://www.zlib.net/zlib_license.html), general-purpose, legally unencumbered -- that is, not covered by any patents -- lossless data-compression library for use on virtually any computer hardware and operating system. The zlib data format is itself portable across platforms. Unlike the LZW compression method used in Unix *compress*(1) and in the GIF image format, the compression method currently used in zlib essentially never expands the data. (LZW can double or triple the file size in extreme cases.) zlib's memory footprint is also independent of the input data and can be reduced, if necessary, at some cost in compression. A more precise, technical discussion of both points is available on [another page](https://www.zlib.net/zlib_tech.html).

**zlib** was written by [Jean-loup Gailly](http://gailly.net/) (compression) and [Mark Adler](http://en.wikipedia.org/wiki/Mark_Adler) (decompression). Jean-loup is also the primary author of [*gzip*](http://www.gzip.org/)(1), the author of the [comp.compression FAQ list](http://www.faqs.org/faqs/compression-faq/) and the former maintainer of [Info-ZIP](http://www.info-zip.org/pub/infozip/)'s [Zip](http://www.info-zip.org/pub/infozip/Zip.html); Mark is also the author of gzip's and [UnZip](http://www.info-zip.org/pub/infozip/UnZip.html)'s main decompression routines and was the original author of Zip. Not surprisingly, the compression algorithm used in zlib is essentially the same as that in gzip and Zip, namely, the `deflate' method that originated in [PKWARE](http://www.pkware.com/)'s PKZIP 2.x.

Mark and Jean-loup can be reached by e-mail at zlib email address. Please read the [**FAQ**](https://www.zlib.net/zlib_faq.html) and the [**manual**](https://www.zlib.net/manual.html) before asking us for help. We are getting too many questions which already have an answer in the *zlib* documentation.

The deflate and zlib specifications both achieved official Internet RFC status in May 1996, and zlib itself was adopted in version 1.1 of the Java Development Kit (JDK), both as a [raw class](https://docs.oracle.com/javase/8/docs/api/java/util/zip/package-summary.html) and as a component of the [JAR archive format](https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html).

The lovely zlib-vise image above was provided courtesy of Bruce Gardner, art director of [Dr. Dobb's Journal](http://www.ddj.com/). It appears in Mark Nelson's article in the January 1997 issue (see below).

The current release is publicly available here:

 *  **zlib** source code, version 1.2.11, tar.gz format (593K, SHA-256 hash c3e5e9fdd5004dcb542feda5ee4f0ff0744628baf8ed2dd5d66f8ca1197cb1a1):

 *  [US (zlib.net)](https://www.zlib.net/zlib-1.2.11.tar.gz) ([GPG signature](https://www.zlib.net/zlib-1.2.11.tar.gz.asc))

 *  [Pick a mirror (prdownloads.sourceforge.net)](http://prdownloads.sourceforge.net/libpng/zlib-1.2.11.tar.gz?download)

 *  **zlib** source code, version 1.2.11, tar.xz format (457K, SHA-256 hash 4ff941449631ace0d4d203e3483be9dbc9da454084111f97ea0a2114e19bf066):

 *  [US (zlib.net)](https://www.zlib.net/zlib-1.2.11.tar.xz) ([GPG signature](https://www.zlib.net/zlib-1.2.11.tar.xz.asc))

 *  [Pick a mirror (prdownloads.sourceforge.net)](http://prdownloads.sourceforge.net/libpng/zlib-1.2.11.tar.xz?download)

 *  **zlib** source code, version 1.2.11, zipfile format (730K, SHA-256 hash d7510a8ee1918b7d0cad197a089c0a2cd4d6df05fee22389f67f115e738b178d):

 *  [US (zlib.net)](https://www.zlib.net/zlib1211.zip) ([GPG signature](https://www.zlib.net/zlib1211.zip.asc))

 *  [Pick a mirror (prdownloads.sourceforge.net)](http://prdownloads.sourceforge.net/libpng/zlib1211.zip?download)

Note that zlib is an integral part of [libpng](http://www.libpng.org/pub/png/pngcode.html) and has been tested extensively as part of many [PNG-supporting applications](http://www.libpng.org/pub/png/pngapps.html).

**zlib Information**

 *  [**zlib Frequently Asked Questions**](https://www.zlib.net/zlib_faq.html)

 *  [Zlib-announce mailing list](http://zlib.net/mailman/listinfo/zlib-announce_madler.net)

New versions of zlib are announced on this list.

 *  [Zlib-devel mailing list](http://zlib.net/mailman/listinfo/zlib-devel_madler.net)

Please do not send questions or comments about zlib to this mailing list. Send those directly to the authors at zlib email address after checking the [FAQ](https://www.zlib.net/zlib_faq.html) and the [manual](https://www.zlib.net/manual.html), of course. The zlib-devel list is for the development of zlib—members are contributors to and testers of new versions of zlib.

 *  [zlib Manual](https://www.zlib.net/manual.html)

 *  [zlib Usage Example](https://www.zlib.net/zlib_how.html)

 *  [zlib Technical Details](https://www.zlib.net/zlib_tech.html)

 *  zlib-related specifications:

* [RFC 1950 ZLIB Compressed Data Format Specification version 3.3](http://tools.ietf.org/html/rfc1950)
* [RFC 1951 DEFLATE Compressed Data Format Specification version 1.3](http://tools.ietf.org/html/rfc1951)
* [RFC 1952 GZIP file format specification version 4.3](http://tools.ietf.org/html/rfc1952)

 *  [Deflate stream disassembler.](https://github.com/madler/infgen/) infgen.c produces a readable description of a gzip, zlib, or raw deflate stream.

 *  [zlib's Deflate Algorithm](https://www.zlib.net/feldspar.html)

 *  [zlib's deflate flush modes](http://www.bolet.org/~pornin/deflate-flush.html)

 *  [zlib License](https://www.zlib.net/zlib_license.html)

 *  [All released versions of zlib](https://www.zlib.net/fossils)

 *  [zlib on github](https://github.com/madler/zlib)

**CRC (Cyclic Redundancy Check) Bonus Information**

 *  [Ross Williams' classic "A Painless Guide to CRC Error Detection Algorithms"](https://www.zlib.net/crc_v3.txt)

 *  [Code to generate any CRC, with a list of CRC descriptions.](https://github.com/madler/crcany) crcany.c can take a description of a CRC and compute that CRC efficiently. It includes bit-wise, table-driven byte-wise, and table-driven word-wise CRC algorithms.

 *  [Code to modify a message so that it generates the desired CRC.](https://github.com/madler/spoof) spoof.c takes an abbreviated description of the CRC, the exclusive-or of the current CRC of the message and the desired CRC, the length of the message, and a list of bit locations in a message, and tells you which of those bits should be inverted in the message to get the desired CRC. Note that it does not need the message itself, due to the linearity property of CRCs.

**ZIP File Processing Bonus Software**

 *  [Code to read a zip file as a stream and extract its contents.](https://github.com/madler/sunzip) sunzip.c will read a zip file from stdin and extract the files therein that use compression methods 0, 8, 9, or 12 (stored, deflate, deflate64, or bzip2). It accepts Zip64 input.

 *  [Code to merge multiple zip files into a single zip file.](https://www.zlib.net/zipknit15.tar.gz) zipknit.c accepts Zip64 input files, and will create Zip64 output if the combined size of the merged zip file warrants it. All compression formats are permitted, since no decompression or recompression is performed. Encrypted entries are permitted, and pass through unscathed.

**Related External Links**

 *  zlib for Linux, both [shared](http://rpmfind.net/linux/rpm2html/search.php?query=zlib) and [static plus headers](http://rpmfind.net/linux/rpm2html/search.php?query=zlib-devel) (RPM format, many architectures)

 *  [zlib for Solaris](http://www.ibiblio.org/pub/packages/solaris/sparc/) (alternate)

 *  zlib for macOS (Mac OS X): zlib is already included as part of macOS

 *  [zlib for Palm Pilot](http://palmzlib.sourceforge.net/)

 *  [zlib for Newton OS](http://www.kallisys.com/newton/zlib/)

 *  [zlib for Windows CE](http://www.tenik.co.jp/~adachi/wince/)

 *  [zlib for Windows 9x/NT/2000/XP/2003](http://www.winimage.com/zLibDll/) (DLL version, plus related utilities)

 *  [zlib for Windows 9x/NT](http://gnuwin32.sourceforge.net/packages/zlib.htm) (DLL and static version)

 *  [DotNetZip](http://www.codeplex.com/DotNetZip) zip file manipulation for .NET, and more (including replacements for the buggy Microsoft GZipStream and DeflateStream classes)

 *  [zlib for .NET in C#](http://www.componentace.com/zlib_.NET.htm)

 *  [zlib DLL wrapper for .NET in C#](http://zlibnet.codeplex.com/)

 *  [Zip for .NET](http://xceed.com/Zip_Net_Intro.html?SrcUrl=zlib.net)

 *  [Mark Nelson](http://www.dogma.net/markn/)'s [ZlibTool article](http://www.dogma.net/markn/articles/zlibtool/zlibtool.htm) (January 1997)

 *  [zlib C++ wrapper](http://www.cs.unc.edu/Research/compgeom/gzstream/) for the gz\* functions.

 *  [C++ zlib and gzip filters](http://www.boost.org/libs/iostreams/doc/home.html) in an iostream framework.

 *  [zlib 32-bit OCX](http://www.dogma.net/markn/articles/zlibtool/zlibtool.htm) (C++ source and binaries for use with Visual Basic 4.x or Delphi 2.0)

(unsupported [VB5 binary](http://www.dogma.net/markn/ZlibOCX2.dll) also available)

 *  [zlib Delphi 5 interface](http://www.base2ti.com/zlib.htm)

(includes compiled object files and corresponding C++ Builder 5 project files)

 *  [zlib Perl interface](http://www.cpan.org/modules/by-module/Compress/) (source code; look for Compress-Zlib\*.tar.gz)

 *  [zlib Python interface](http://docs.python.org/library/zlib.html) (online manual; part of the standard library as of Python 1.5)

 *  [zlib Tcl interface](http://mkextensions.sourceforge.net/) mkZiplib

 *  [zlib Haskell interface](http://hackage.haskell.org/package/zlib/)

 *  [zlib Java interface](https://docs.oracle.com/javase/8/docs/api/java/util/zip/package-summary.html) (see also [JAR format](https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html))

 *  [zlib reimplementation in pure Java](http://www.jcraft.com/jzlib/)

(not tested by us, but looks like a good alternative to java.util.zip)

 *  [Mark Nelson](http://www.dogma.net/markn/)'s [JavaZip article](http://www.dogma.net/markn/articles/JavaZip/JavaZip.html) (with source code) (December 1997)

 *  [Random access for gzip archives, for Java](http://code.google.com/p/jzran)

 *  [Gilles Vollant](http://www.winimage.com/)'s zlib-based [mini-zip and mini-unzip](http://www.winimage.com/zLibDll/unzip.html)

(see also Info-ZIP's [UnZip](http://www.info-zip.org/pub/infozip/UnZip.html), which optionally can be compiled with zlib)

 *  [Scott Ludwig](http://www.eskimo.com/~scottlu/)'s zlib-based [CExe executable compressor](http://www.eskimo.com/~scottlu/win/) for Win32

 *  [zlib technical issues, including spec errors](http://optipng.sourceforge.net/pngtech/)

 *  [zlib information in Japanese](http://oku.edu.mie-u.ac.jp/~okumura/compression/zlib.html)

 *  [zlib information in Russian](http://zlib.net.ru/)

 *  [*Real World Scanning and Halftones*](https://books.google.com/books/about/Real_World_Scanning_and_Halftones.html?id=Yr01lLfnQ54C) (second edition includes a section on zlib)

 *  Markus Oberhumer's [LZO `real-time' data compression library](http://www.oberhumer.com/opensource/lzo/)

(not tested by us, but looks like a good alternative if you need more speed and less compression)

 *  [lz4, a very fast compression algorithm](http://code.google.com/p/lz4/)

(not tested by us, but looks like an even better alternative if you need more speed and less compression)

 *  [Zstandard, a better compression algorithm](https://github.com/facebook/zstd)

(not tested by us, but appears to be a better alternative to zlib in both dimensions of compression and speed, as well as decompression speed)

 *  [libbzip2](http://sources.redhat.com/bzip2/)

(not tested by us, but looks like a good alternative if you need more compression and less speed)

 *  [PPP Deflate Protocol](http://www.ietf.org/rfc/rfc1979.txt) (RFC 1979)

 *  [Info-ZIP Home Page](http://www.info-zip.org/pub/infozip/)

 *  [Portable Network Graphics (PNG) Home Page](http://www.libpng.org/pub/png/)

 *  [*gzip* Home Page](http://www.gzip.org/)

 *  [*pigz* (parallel gzip) Home Page](http://zlib.net/pigz/)

 *  [DataCompression.info](http://datacompression.info/)

 *  [comp.compression Frequently Asked Questions list](http://www.faqs.org/faqs/compression-faq/)