

Purpose

This specification is intended to define the implementation of zip as used by the TorrentZip standard.

Reference

For a full description of the zip files specification:

<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

For the trrntzip source code:

<http://sourceforge.net/projects/trrntzip/>

ZLib Compression:

<http://zlib.net/>

General format of a torrentzipped .zip file with n files:

[local file header 1]

[file data 1]

[local file header 2]

[file data 2]

.

[local file header n]

[file data n]

← start of central directory (SOCD File offset)

[central directory file 1]

[central directory file 2]

.

[central directory file n]

← end of central directory (EOCD File offset)

[end of central directory record]

Local file header x: (Showing torrentzipped default values)

UInt32	Local file header signature	(0x04034b50)	
UInt16	Version needed to extract	20	= File is compressed using Deflate compression
UInt16	General purpose bit flag	2	= Maximum compression option was used
UInt16	Compression method	8	= The file is Deflated
UInt16	Last mod file time	48128	= 11:32 PM
UInt16	Last mod file date	8600	= 12/24/1996
UInt32	CRC-32		= File CRC
UInt32	Compressed size		= File Compressed Size
UInt32	Uncompressed size		= File Uncompressed Size
UInt16	Filename length		= Filename length
UInt16	Extra field length	0	= No extra field information
Byte[]	Filename(variable size)		= Byte array of filename

Notes:

The default values show are required to have consistent torrentzipped files.

Default time/date of 11:32pm 12/24/1996 is the date of the first ever MAME release.

File data x:

The data compression must be exactly as ZLib version 1.1.3 using maximum compression level 9.

Central Directory file x: (Showing torrentzipped default values)

UInt32	Central file header signature	(0x02014b50)	
UInt16	Version made by	0	= MS_DOS and OS/2 (FAT/FAT32 file systems)
UInt16	Version needed to extract	20	= File is compressed using Deflate compression
UInt16	General purpose bit flag	2	= Maximum compression option was used
UInt16	Compression method	8	= The file is Deflated
UInt16	Last mod file time	48128	= 11:32 PM
UInt16	Last mod file date	8600	= 12/24/1996
UInt32	CRC-32		= File CRC
UInt32	Compressed size		= File Compressed Size
UInt32	Uncompressed size		= File Uncompressed Size
UInt16	File name length		= Filename length
UInt16	Extra field length	0	= No extra field information
UInt16	File comment length	0	= No file comment
UInt16	Disk number start	0	= Multi disk storage not used so set to disk 0
UInt16	Internal file attributes	0	= No internal attributes
UInt32	External file attributes	0	= No external attributes
UInt32	Relative offset of local header		= File offset of this files Local Header
Byte[]	File name (variable size)		= Byte array of filename

End Of Central Directory:

UInt32	End of central dir signature	(0x06054b50)	
UInt16	Number of this disk	0	= Multi disk storage not used so set to disk 0
UInt16	Number of the disk with the start of the central directory	0	= Multi disk storage not used so set to disk 0
UInt16	Total number of entries in the central directory on this disk	n	= Total number of files
UInt16	Total number of entries in the central directory	n	= Total number of files
UInt32	Size of the central directory	EOCD-SOCD	= length of the central directories
UInt32	Offset of start of central directory with respect to the starting disk number	SOCD	= Start of central directory
UInt16	.ZIP file comment length	22	= torrentzipped comment
Byte[22]	.ZIP file comment	TORRENTZIPPED-XXXXXXXX	

Notes:

See above 'General format of a torrentzipped .zip file with n files' for SOCD & EOCD

The TorrentZipped Files Comments

The .ZIP file comments in the End of Central directory is used to check the validity of the torrentzipped file.

The comment must be formatted as the 22 bytes of TORRENTZIPPED-XXXXXXXX
The XXXXXXXX is the CRC32 of the central directory records stored as hexadecimal upper case text.
(the CRC32 of the bytes in the file between SOCD & EOCD)

This comment ensures that if any change is made to the files within the zip this checksum will no longer match the byte data in the central directory, and in this way we can check the validity of a torrentzip file.

File Order with a TorrentZip

For the creation of consistent torrentzipped files, the file order is also very import.
Files must be sorted by filename using a lower case sort.

Directory separator character

As zips only store files (not directories), files in directories are represented by storing a relative path to the filename. For example file 'test1.rom' in directory 'set1' would be stored with a filename of 'set1/test1.rom'. Some zipping programs will store this as 'set1\test1.rom'.

This leads to a possible naming inconsistency. The zip file format state "All slashes should be forward slashes '/' as opposed to backwards slashes '\ ". So Torrentzip will change all '\ character to '/'.
(This must be done before sorting, to ensure that the sort is performed correctly.)

Directory Entries and Empty Directories

A directory entry is stored in a zip by adding a file entry ending in a directory separator character with a zero size and CRC. So directory 'set1' would be stored as a zero length, zero CRC file called 'set1/'.
Some zip programs when adding the previously mentioned file 'set1/test1.rom' will also add the directory 'set1/', this creates an inconsistency problem. In this example the 'set1/' directory entry is unnecessary, as the filename 'set1/test1.rom' implies the existents of the 'set1/' directory. To resolve this inconsistency un-needed directories should be removed from the zip, the only needed directory entries are empty directories that are not implied by any file entries.

Example:

Filename	Size	CRC
set1/	0	00000000
set1/test1.rom	1024	53AC4D0
set2/	0	00000000

The set1/ entry should be removed, as it is implied by the set1/test1.rom file. The set2/ entry should be kept to create the empty directory, as removing it would completely remove the set2 directory.

Repeat Files

Another test that could be performed is checking for repeat file entries inside the zip, most zip programs have a hard time handling this and will just ignore this repeat giving the user no way of knowing there is a repeat filename problem. So it would fix another possible inconsistency if torrentzip scanning at least warned about repeat filename being found inside a zip.