The structure of a PKZip file

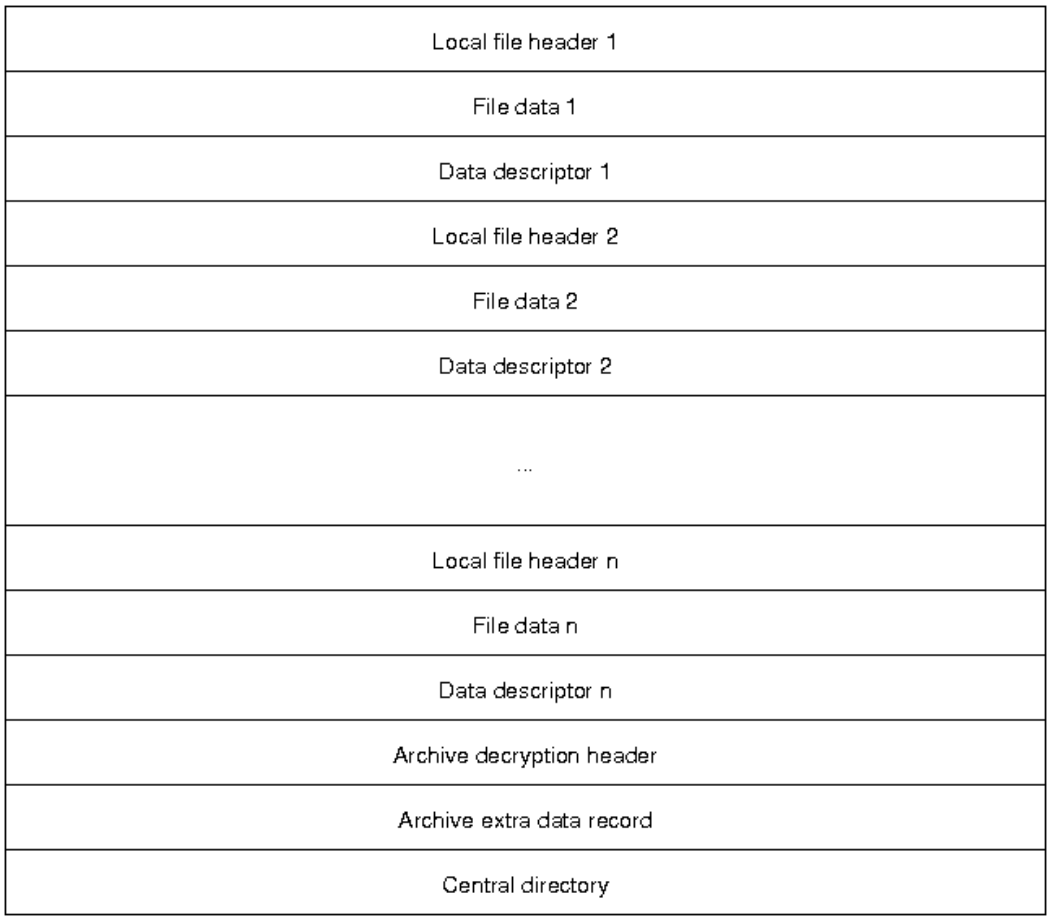
by Florian Buchholz

Overview

This document describes the on-disk structure of a PKZip (Zip) file. The documentation currently only describes the file layout format and meta information but does not address the actual compression or encryption of the file data itself. This documentation also does not discuss Zip archives that span multiple files in great detail. This documentation was created using the official documentation provided by PKWare Inc.

General structure

Each Zip file is structured in the following manner:



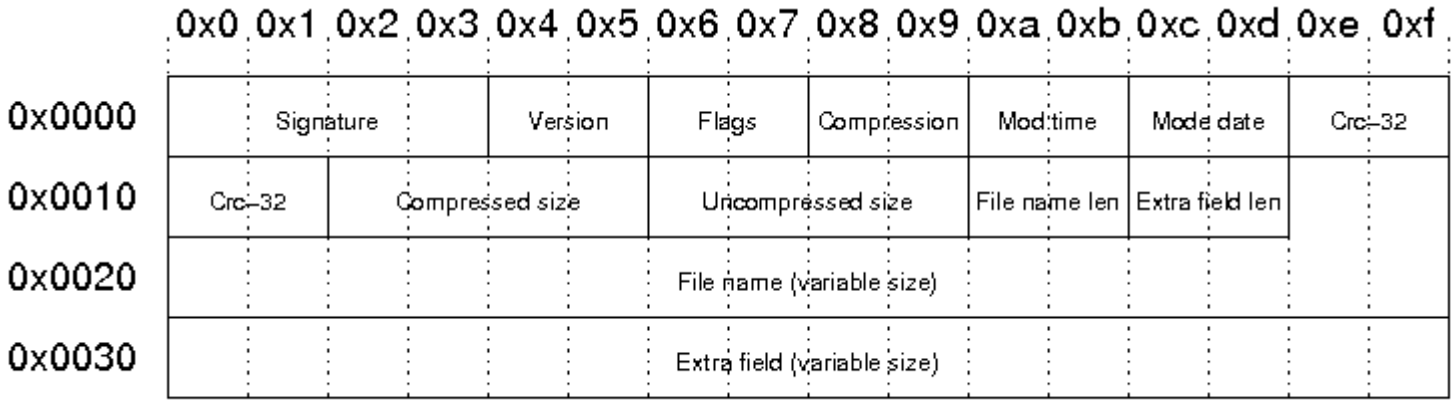
The archive consists of a series of local file descriptors, each containing a local file header, the actual compressed and/or encrypted data, as well as an optional data descriptor. Whether a data descriptor exists or not depends on a flag in the local file header.

Following the file descriptors is the archive decryption header, which only exists in PKZip file version 6.2 or greater. This header is only present if the central directory is encrypted and contains information about the encryption specification. The archive extra data record is also only for file of version 6.2 or greater and is not present in all Zip files. It is used in to support the encryption or compression of the central directory.

The central directory summarizes the local file descriptors and carries additional information regarding file attributes, file comments, location of the local headers, and multi-file archive information.

Local file headers

Each local file header has the following structure:



Signature	The signature of the local file header. This is always '\x50\x4b\x03\x04'.
Version	PKZip version needed to extract
Flags	General purpose bit flag: Bit 00: encrypted file Bit 01: compression option Bit 02: compression option Bit 03: data descriptor Bit 04: enhanced deflation Bit 05: compressed patched data Bit 06: strong encryption Bit 07-10: unused Bit 11: language encoding Bit 12: reserved Bit 13: mask header values Bit 14-15: reserved
Compression method	00: no compression 01: shrunk 02: reduced with compression factor 1 03: reduced with compression factor 2 04: reduced with compression factor 3 05: reduced with compression factor 4 06: imploded 07: reserved 08: deflated 09: enhanced deflated 10: PKWare DCL imploded 11: reserved 12: compressed using BZIP2 13: reserved 14: LZMA 15-17: reserved 18: compressed using IBM TERSE 19: IBM LZ77 z 98: PPMd version I, Rev 1
File modification time	stored in standard MS-DOS format: Bits 00-04: seconds divided by 2 Bits 05-10: minute Bits 11-15: hour
File modification date	stored in standard MS-DOS format: Bits 00-04: day Bits 05-08: month Bits 09-15: years from 1980
Crc-32 checksum	value computed over file data by CRC-32 algorithm with 'magic number' 0xdebb20e3 (little endian)
Compressed size	if archive is in ZIP64 format, this filed is 0xffffffff and the length is stored in the extra field

Uncompressed size	if archive is in ZIP64 format, this filed is 0xffffffff and the length is stored in the extra field
File name length	the length of the file name field below
Extra field length	the length of the extra field below
File name	the name of the file including an optional relative path. All slashes in the path should be forward slashes '/'.
Extra field	Used to store additional information. The field consistes of a sequence of header and data pairs, where the header has a 2 byte identifier and a 2 byte data size field.

Example

Our sample zip file starts with a local file header:

```
00000000  50 4b 03 04 14 00 00 00  08 00 1c 7d 4b 35 a6 e1  |PK.........}K5..|
00000010  90 7d 45 00 00 00 4a 00  00 00 05 00 15 00 66 69  |.}E...J.......fi|
00000020  6c 65 31 55 54 09 00 03  c7 48 2d 45 c7 48 2d 45  |le1UT....H-E.H-E|
00000030  55 78 04 00 f5 01 f5 01  0b c9 c8 2c 56 00 a2 92  |Ux.........,V...|
```

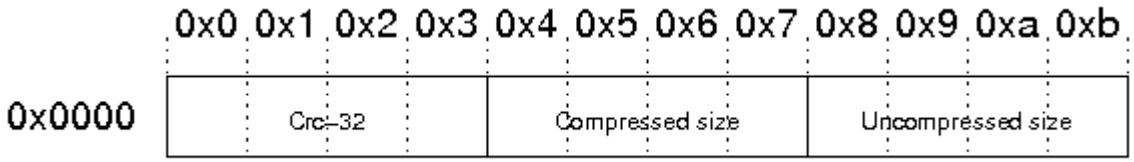
This results in the following fields and field values:

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	03	04	14	00	00	00	08	00	1c	7d	4b	35	a6	e1
0x0010	90	7d	45	00	00	00	4a	00	00	00	05	00	15	00	66	69
0x0020	6c	65	31	55	54	09	00	03	c7	48	2d	45	c7	48	2d	45
0x0030	55	78	04	00	f5	01	f5	01								

Signature	'\x50\x4b\x03\x04'.
Version	0x14 = 20 -> 2.0
Flags	no flags
Compression method	08: deflated
File modification time	0x7d1c = 0111110100011100 hour = (01111)10100011100 = 15 minute = 01111(101000)11100 = 40 second = 01111101000(11100) = 28 = 56 seconds 15:40:56
File modification date	0x354b = 0011010101001011 year = (0011010)101001011 = 26 month = 0011010(1010)01011 = 10 day = 00110101010(01011) = 11 10/11/2006
Crc-32 checksum	0x7d90e1a6
Compressed size	0x45 = 69 bytes
Uncompressed size	0x4a = 74 bytes
File name length	5 bytes
Extra field length	21 bytes
File name	"file1"
Extra field	id 0x5455: extended timestamp, size: 9 bytes Id 0x7855: Info-ZIP UNIX, size: 4 bytes

Data descriptor

The data descriptor is only present if bit 3 of the bit flag field is set. In this case, the CRC-32, compressed size, and uncompressed size fields in the local header are set to zero. The data descriptor field is byte aligned and immediately follows the file data. The structure is as follows:



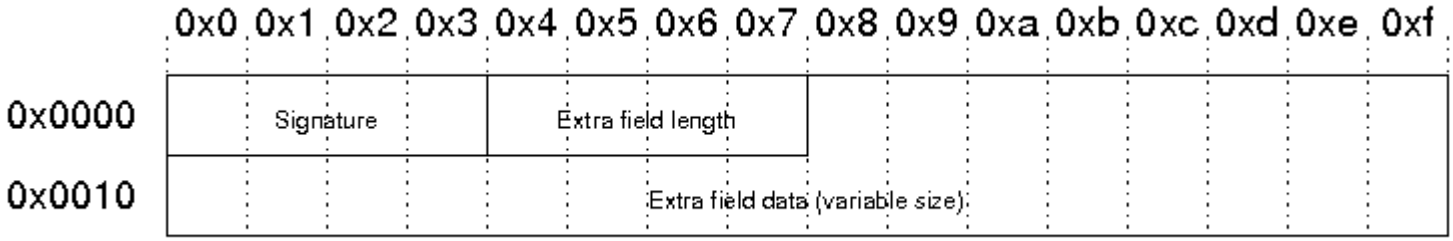
The example file does not contain a data descriptor.

Archive decryption header

This header is used to support the Central Directory Encryption Feature. It is present when the central directory is encrypted. The format of this data record is identical to the Decryption header record preceding compressed file data.

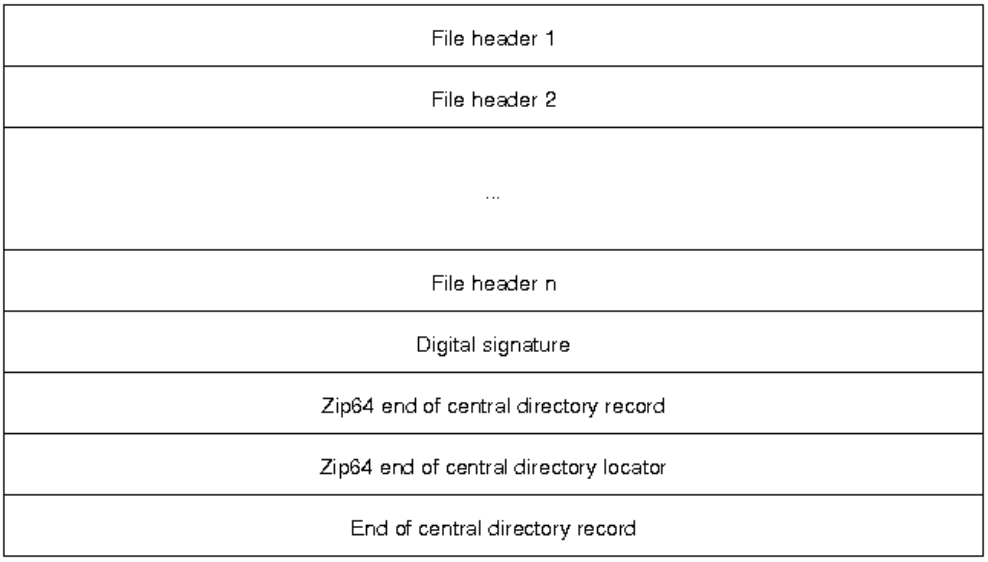
Archive extra data record

This header is used to support the Central Directory Encryption Feature. When present, this record immediately precedes the central directory data structure. The size of this data record will be included in the Size of the Central Directory field in the End of Central Directory record. The structure is as follows:



Central directory

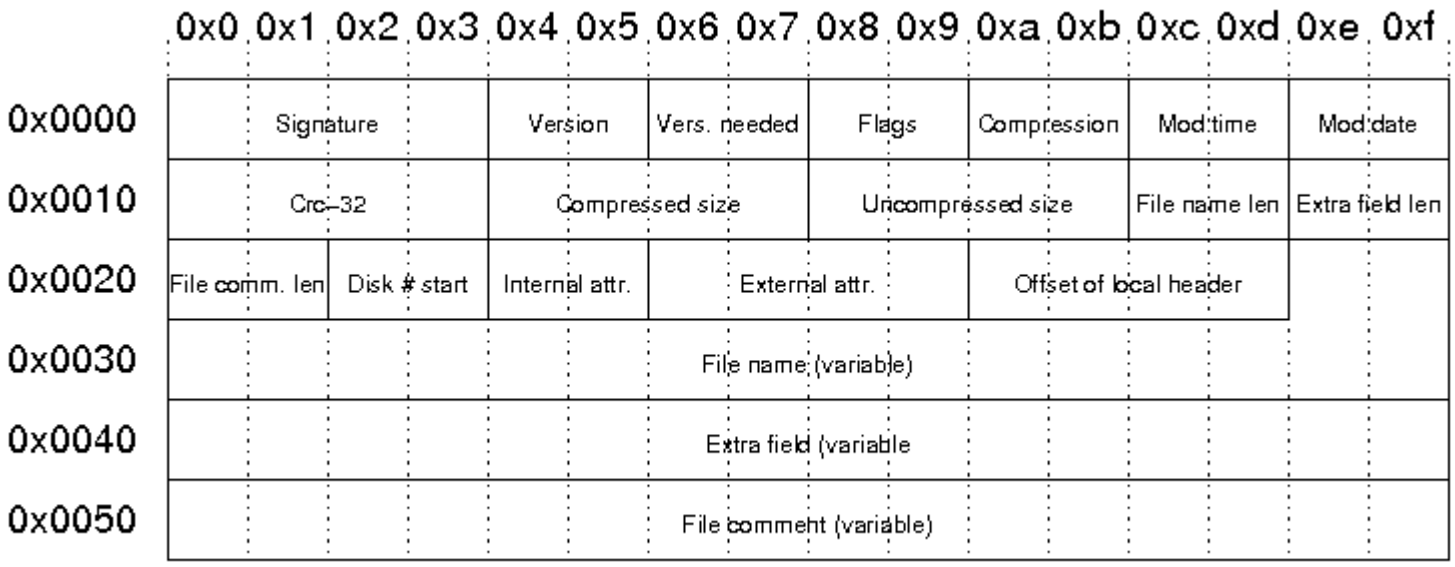
The central directory contains more metadata about the files in the archive and also contains encryption information and information about Zip64 (64-bit zip archives) archives. Furthermore, the central directory contains information about archives that span multiple files. The structure of the central directory is as follows:



The file headers are similar to the local file headers, but contain some extra information. The Zip64 entries handle the case of a 64-bit Zip archive, and the end of the central directory record contains information about the archive itself.

Central directory file header

The structure of the file header in the central directory is as follows:



Signature	The signature of the file header. This is always '\x50\x4b\x01\x02'.
Version	Version made by: upper byte: 0 - MS-DOS and OS/2 (FAT / VFAT / FAT32 file systems) 1 - Amiga 2 - OpenVMS 3 - UNIX 4 - VM/CMS 5 - Atari ST 6 - OS/2 H.P.F.S. 7 - Macintosh 8 - Z-System 9 - CP/M 10 - Windows NTFS 11 - MVS (OS/390 - Z/OS) 12 - VSE 13 - Acorn Risc 14 - VFAT 15 - alternate MVS 16 - BeOS 17 - Tandem 18 - OS/400 19 - OS/X (Darwin) 20 - 255: unused lower byte: zip specification version
Vers. needed	PKZip version needed to extract
Flags	General purpose bit flag: Bit 00: encrypted file Bit 01: compression option Bit 02: compression option Bit 03: data descriptor Bit 04: enhanced deflation

	Bit 05: compressed patched data Bit 06: strong encryption Bit 07-10: unused Bit 11: language encoding Bit 12: reserved Bit 13: mask header values Bit 14-15: reserved
Compression method	00: no compression 01: shrunk 02: reduced with compression factor 1 03: reduced with compression factor 2 04: reduced with compression factor 3 05: reduced with compression factor 4 06: imploded 07: reserved 08: deflated 09: enhanced deflated 10: PKWare DCL imploded 11: reserved 12: compressed using BZIP2 13: reserved 14: LZMA 15-17: reserved 18: compressed using IBM TERSE 19: IBM LZ77 z 98: PPMd version I, Rev 1
File modification time	stored in standard MS-DOS format: Bits 00-04: seconds divided by 2 Bits 05-10: minute Bits 11-15: hour
File modification date	stored in standard MS-DOS format: Bits 00-04: day Bits 05-08: month Bits 09-15: years from 1980
Crc-32 checksum	value computed over file data by CRC-32 algorithm with 'magic number' 0xdebb20e3 (little endian)
Compressed size	if archive is in ZIP64 format, this filed is 0xffffffff and the length is stored in the extra field
Uncompressed size	if archive is in ZIP64 format, this filed is 0xffffffff and the length is stored in the extra field
File name length	the length of the file name field below
Extra field length	the length of the extra field below
File comm. len	the length of the file comment
Disk # start	the number of the disk on which this file exists
Internal attr.	Internal file attributes: Bit 0: apparent ASCII/text file Bit 1: reserved Bit 2: control field records precede logical records Bits 3-16: unused
External attr.	External file attributes: host-system dependent
Offset of local header	Relative offset of local header. This is the offset of where to find the corresponding local file header from the start of the first disk.
File name	the name of the file including an optional relative path. All slashes in the path should be forward slashes '/'.

Extra field	Used to store additional information. The field consistes of a sequence of header and data pairs, where the header has a 2 byte identifier and a 2 byte data size field.
File comment	An optional comment for the file.

Example:

The corresponding file header from our local file header example above starts at byte 0x9a2 in the example file:

```
000009a0  28 f0 50 4b 01 02 17 03  14 00 00 00 08 00 1c 7d  |(.PK...........}|
000009b0  4b 35 a6 e1 90 7d 45 00  00 00 4a 00 00 00 05 00  |K5...}E...J.....|
000009c0  0d 00 1c 00 00 00 01 00  00 00 a4 81 00 00 00 00  |................|
000009d0  66 69 6c 65 31 55 54 05  00 03 c7 48 2d 45 55 78  |file1UT....H-EUx|
000009e0  00 00 74 68 69 73 20 69  73 20 61 20 63 6f 6d 6d  |..this is a comm|
000009f0  65 6e 74 20 66 6f 72 20  66 69 6c 65 20 31 50 4b  |ent for file 1PK|
```

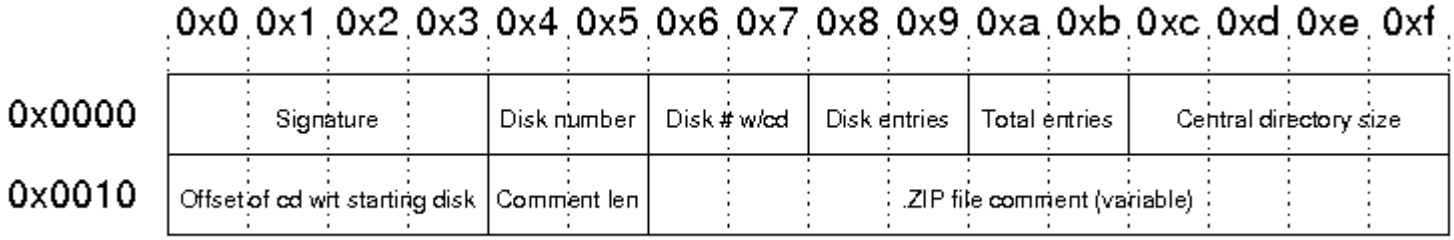
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	01	02	17	03	14	00	00	00	08	00	1c	7d	4b	35
0x0010	a6	e1	90	7d	45	00	00	00	4a	00	00	00	05	00	0d	00
0x0020	1c	00	00	00	01	00	00	00	a4	81	00	00	00	00	66	69
0x0030	6c	65	31	55	54	05	00	03	c7	48	2d	45	55	78	00	00
0x0040	74	68	69	73	20	69	73	20	61	20	63	6f	6d	6d	65	6e
0x0050	74	20	66	6f	72	20	66	69	6c	65	20	31				

Signature	'\x50\x4b\x01\x02'.
Version	0x0317 upper byte: 03 -> UNIX lower byte: 23 -> 2.3
Version needed	0x14 = 20 -> 2.0
Flags	no flags
Compression method	08: deflated
File modification time	0x7d1c = 0111110100011100 hour = (01111)10100011100 = 15 minute = 01111(101000)11100 = 40 second = 01111101000(11100) = 28 = 56 seconds 15:40:56
File modification date	0x354b = 0011010101001011 year = (0011010)101001011 = 26 month = 0011010(1010)01011 = 10 day = 00110101010(01011) = 11 10/11/2006
Crc-32 checksum	0x7d90e1a6
Compressed size	0x45 = 69 bytes
Uncompressed size	0x4a = 74 bytes
File name length	5 bytes
Extra field length	13 bytes
File comment length	28 bytes

Disk # start	0
Internal attributes	Bit 0 set: ASCII/text file
External attributes	0x81a40000
Offset of local header	0
File name	"file1"
Extra field	id 0x5455: extended timestamp, size: 5 bytes Id 0x7855: Info-ZIP UNIX, size: 0 bytes
File comment	"this is a comment for file 1"

End of central directory record

The structure of the end of central directory record is as follows:



Signature	The signature of end of central directory record. This is always '\x50\x4b\x05\x06'.
Disk Number	The number of this disk (containing the end of central directory record)
Disk # w/cd	Number of the disk on which the central directory starts
Disk entries	The number of central directory entries on this disk
Total entries	Total number of entries in the central directory.
Central directory size	Size of the central directory in bytes
Offset of cd wrt to starting disk	Offset of the start of the central directory on the disk on which the central directory starts
Comment len	The length of the following comment field
ZIP file comment	Optional comment for the Zip file

Example:

The end of central directory in out example file starts at byte 0xb36:

```
00000b30  6f 6d 6d 65 6e 74 50 4b  05 06 00 00 00 00 04 00  |ommentPK........|
00000b40  04 00 94 01 00 00 a2 09  00 00 33 00 74 68 69 73  |..........3.this|
00000b50  20 69 73 20 61 0d 0a 6d  75 6c 74 69 6c 69 6e 65  | is a..multiline|
00000b60  20 63 6f 6d 6d 65 6e 74  20 66 6f 72 20 74 68 65  | comment for the|
00000b70  20 65 6e 74 69 72 65 20  61 72 63 68 69 76 65     | entire archive|
```

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0000	50	4b	05	06	00	00	00	00	04	00	04	00	94	01	00	00
0x0010	a2	09	00	00	33	00	74	68	69	73	20	69	73	20	61	0d
0x0020	0a	6d	75	6c	74	69	6c	69	6e	65	20	63	6f	6d	6d	65
0x0030	6e	74	20	66	6f	72	20	74	68	65	20	65	6e	74	69	72
0x0040	65	20	61	72	63	68	69	76	65							

Signature	'\x50\x4b\x05\x06'.
Disk Number	0
Disk # w/cd	0
Disk entries	4
Total entries	4
Central directory size	0x194 = 404 bytes
Offset of cd wrt to starting disk	byte 0x9a2 = byte 2466
Comment len	0x33 = 51 bytes
ZIP file comment	"this is a multiline comment for the entire archive"