

# The structure of a PKZip file

by Florian Buchholz

General structure Overview

Local file headers Data descriptor Archive decryption header

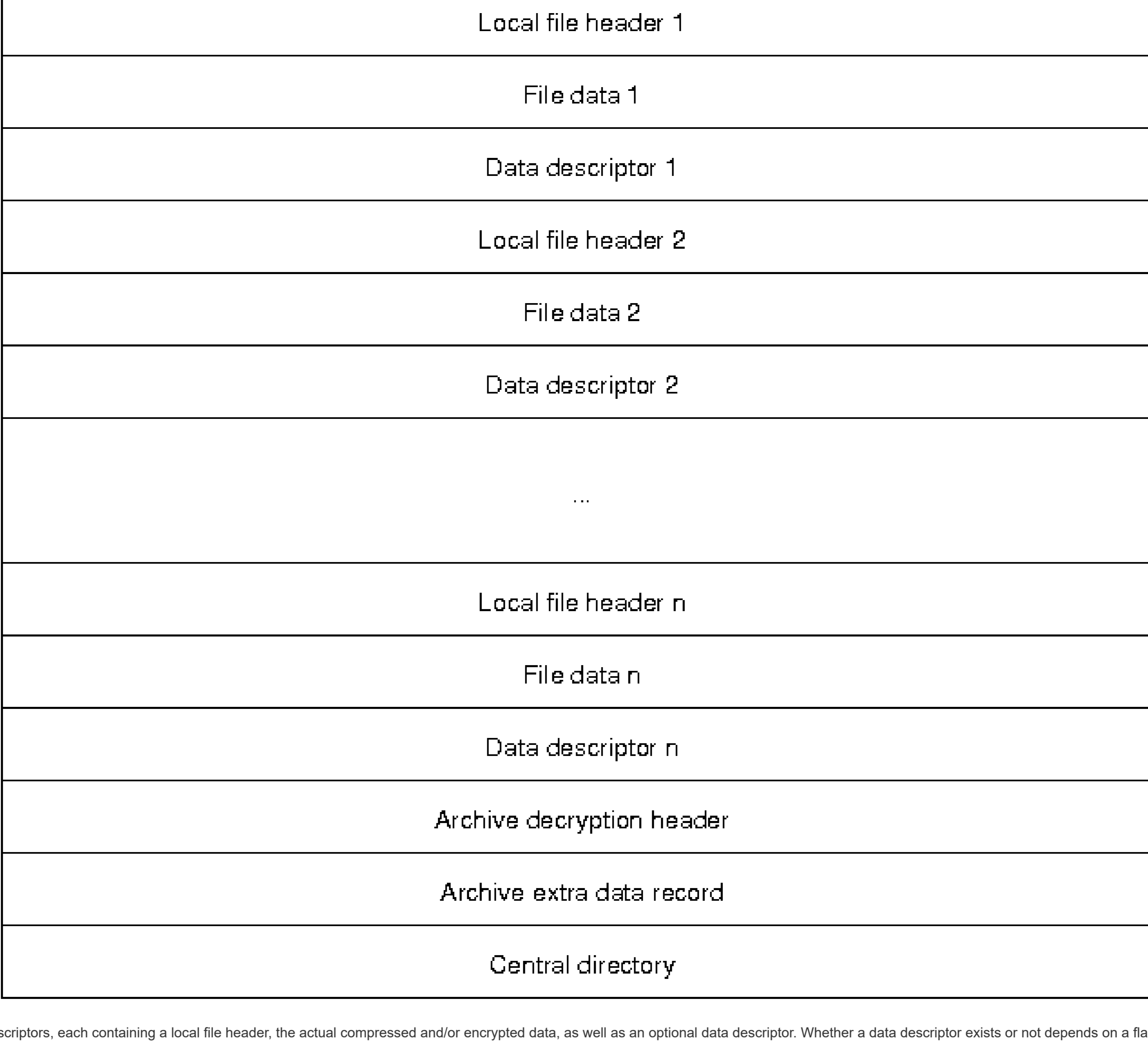
Archive extra data record Central directory

Printable Version

This document describes the on-disk structure of a PKZip 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:

0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf

0x0000	Signature	Version	Flags	Compression	Modtime	Moddate	Crc-32
0x0010	Crc-32	Compressed size	Uncompressed size	File name len	Extra field len		
0x0020	File name (variable size)						
0x0030	Extra field (variable size)						

Signature: The signature of the local file header. This is always 0x41414141.

Version: PKZip version needed to extract

Flags: General purpose bit flag: Bit 00: encrypted file; Bit 01: reserved; 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: compressed using BZIP2; 07: reserved; 08: PKWARE DCL imploded; 09: reserved; 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; 08: PPMd version 1, 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" 0x05050505 (little endian)

Compressed size: if archive is in ZIP64 format, this field is 0xffff and the length is stored in the extra field

Uncompressed size: if archive is in ZIP64 format, this field is 0xffff 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 consists 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 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00000010 90 7d 45 00 00 00 00 00 00 00 05 00 15 00 66 69
00000020 6c 65 31 55 54 09 00 03 c7 48 2d 45 c7 48 2d 45
00000030 55 78 04 00 f5 01 f5 01
```

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	00	00	1c	7d	4b	35	a6	e1
0x0010	90	7d	45	00	00	00	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: 0x41414141

Version: 0x14 = 20 -> 2.0

Flags: no flags

Compression method: 08: deflated

File modification time: 0x761c = 011110100011100 hour = 011110100011100 = 15 minute = 011110100011100 = 40 second = 011110100011100 = 28 = 56 seconds 15:40:56

File modification date: 0x35a6 = 0010101010010111 year = 0010101010010111 = 26 month = 0010101010101011 = 10 day = 0010101010101011 = 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: "file"

Extra field: id 0x5455: extended timestamp, size: 8 bytes; id 0x7855: Info-ZIP UNIX, size: 4 bytes

## Data descriptor

The data descriptor is only present if bit 3 of the bit flag 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:

0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb

0x0000	Crc-32	Compressed size	Uncompressed size
--------	--------	-----------------	-------------------

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

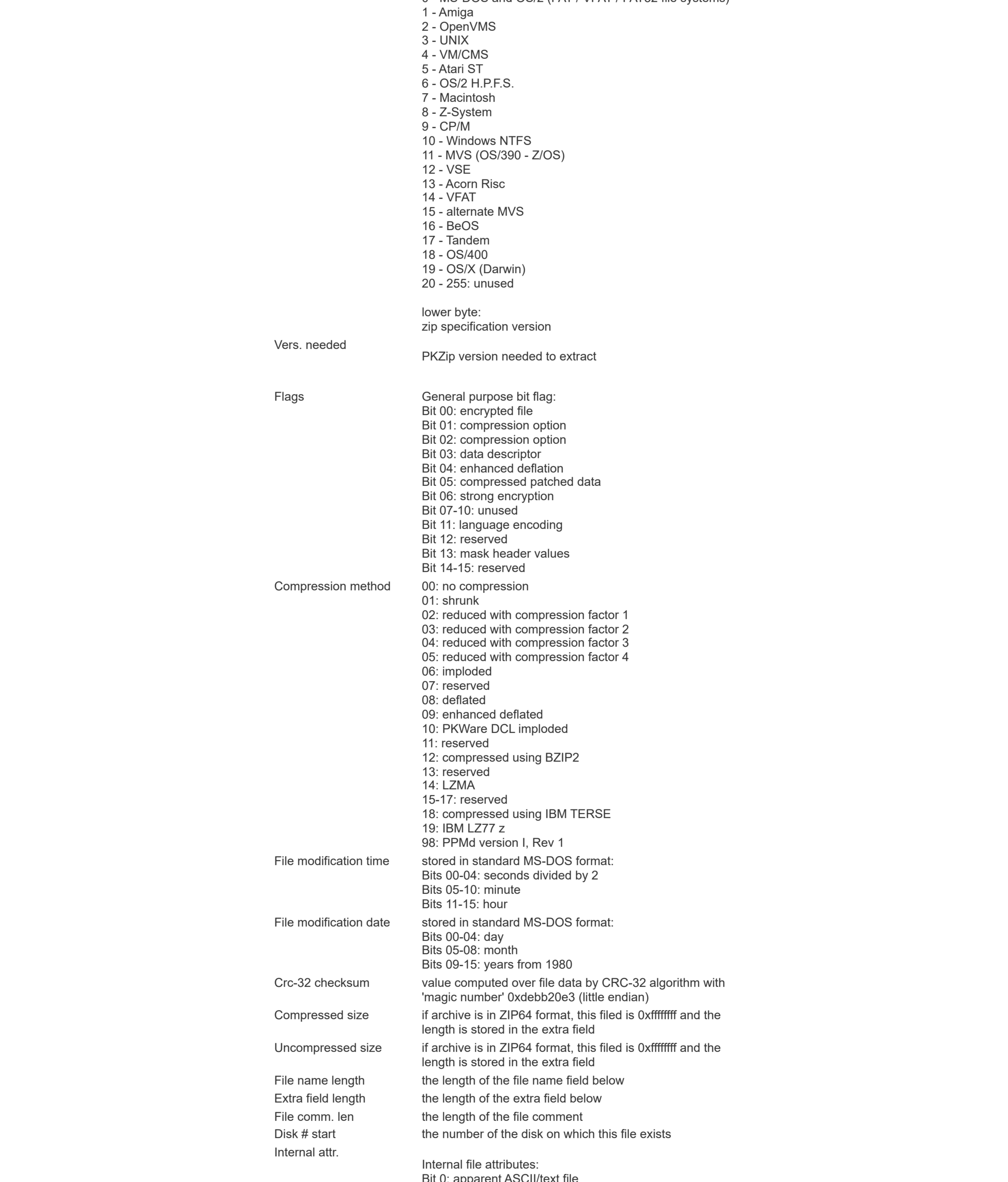
This header is used to support the Central Directory Encryption Feature. When present, this record immediately precedes the central directory data structure. The size of this data record will be included in the Size of the Central Directory field in the End of Central Directory record. The structure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

0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf

0x0000	Signature	Extra field length
0x0010	Extra field data (variable size)	

## 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:

0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf

0x0000	Signature	Version	Version needed	Flags	Compression	Modtime	Moddate
0x0010	Crc-32	Compressed size	Uncompressed size	File name len	Extra field len		
0x0020	File comment len	Disk # start	Internal attr.	External attr.	Offset of local header		
0x0030	File name (variable)						
0x0040	Extra field (variable)						
0x0050	File comment (variable)						

Signature: The signature of the file header. This is always 0x41414141.

Version: zip specification version

Version 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: PKWARE DCL imploded; 09: reserved; 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; 08: PPMd version 1, 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" 0x05050505 (little endian)

Compressed size: if archive is in ZIP64 format, this field is 0xffff and the length is stored in the extra field

Uncompressed size: if archive is in ZIP64 format, this field is 0xffff 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 comment 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 3-10: 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 consists 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.

## End of central directory record

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

0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf

0x0000	Signature	Disk number	Disk # w/ol	Disk entries	Total entries	Central directory size
0x0010	Offset of cd wrt starting disk	Comment len		ZIP file comment (variable)		

Signature: The signature of end of central directory record. This is always 0x504b504b.

Disk Number: The number of this disk (containing the end of central directory record)

Disk # w/ol: 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: An optional comment for the Zip file

Example

The end of central directory in our example file starts at byte 0xb30:

```
00000b30 6f 6d 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f
00000b40 80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98 99
00000b50 a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8 b9
00000b60 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8 d9
00000b70 e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8 f9
```

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	05	06	00	00	00	00	04	00	00	94	01	00	00
0x0010	a2	09	00	00	33	00	74	68	69	73	20	69	73	20	61
0x0020	0a	6d	75	6c	74	69	6c	69	6e	65	20	63	6f	6d	65
0x0030	6e	74	20	66	6e	6e	6e	6e	65	20	65	6e	74	69	72
0x0040	65	20	61	72	62	68	69	74	65						

Signature: 0x504b504b

Disk Number: 0

Disk # w/ol: 0

Disk entries: 4

Total entries: 4

Central directory size: 0x194 = 404 bytes

Offset of cd wrt to starting disk: 0xb30 = byte 2468

Comment len: 0x33 = 51 bytes

ZIP file comment: multiline comment for the entire archive