EWF 2 specification

Expert Witness Compression Format version 2 specification

By Joachim Metz <joachim.metz@gmail.com>

Summary

In EnCase 7 Guidance Software introduced a version 2 of the Expert Witness Compression Format (EWF). Although at high-level both version 1 and 2 are quite similar in the details both versions differ significantly.

This document is intended as a working document for the EWF2 specification. Which should allow existing Open Source forensic tooling to be able to process this file type.

Document information

Author(s): Joachim Metz <joachim.metz@gmail.com>

Abstract: This document contains the EWF file format version 2 specification.

Classification: Public

Keywords: Expert Witness Compression Format, EWF2, EnCase file format

License

Version

Version	Author	Date	Comments
0.0.1	J.B. Metz	July 2012	Initial version
0.0.2	J.B. Metz	July 2012	Additional information about encryption and Lx01.
0.0.3	J.B. Metz	July 2012	Additional information some obtained from Guidance Software.
0.0.4	J.B. Metz	July 2012	Additional information about pattern fill and bzip2 compression.
0.0.5	J.B. Metz	August 2012	Additional information some obtained from Guidance Software and findings.

Table of Contents

1. Overview

In EnCase 7 Guidance Software introduced a version 2 of the Expert Witness Compression Format (EWF). Although at high-level both version 1 and 2 are quite similar in the details both versions differ significantly.

This document will use EWF2 as the name of the format although Guidance named the format EnCase Evidence File Format Version 2, see [ENCASE12]. EWF1 is used to indicate version 1 of the format.

There are 2 different versions of EWF2:

- EWF2-Ex01; "normal" image files to store disk, volume and memory images.
- EWF2-Lx01; logical evidence file to store files and directories.

In EWF2 the data is either compressed or non-compressed, EWF2 no longer distinguishes between multiple compression levels.

In EWF2 support was added to encrypt the data and relevant metadata.

1.1. Test version

The following version of programs were used to test the information within this document:

- EnCase 7.04 (Windows)

2. Segment files

EWF2 stores data in one or more segment files (or segments). Each segment file consists of:

- A file header.
- One or more sections; which [ENCASE12] refers to as link records and data.

EnCase allows for the segment file size to be set at 30 MB at minimum and about 8.8 TB at maximum. Libewf 1 MiB at minimum and about 8 EiB at maximum.

2.1. File header

Each segment file starts with file header, the file header differs for EWF2-Ex01 and EWF2-Lx01.

2.1.1. EWF2-Ex01

[ENCASE12] defines the file header as:

The file header is 32 bytes of size and consists of:

offset	**size**	**value**	**description**
0	8		Signature "EVF2\r\n\x81\x00"
8	1	2	Major version
9	1	1	Minor version

offset	size	value	description
10	2		Compression method See section: 2.1.3 Compression methods
12	4		Segment file number (Series)
16	16		Segment file set identifier Contains a little-endian GUID (version 4)

Version 2.1 is the first version of the EWF2 format. It currently is assumed that EWF1 is considered version 1.0.

2.1.2. EWF2-Lx01

The file header is 32 bytes of size and consists of:

offset	size	value	description
0	8		Signature "LEF2\r\n\x81\x00"
8	1	2	Major version
9	1	1	Minor version
10	2		Compression method See section: 2.1.3 Compression methods
12	4		Segment file number (Series)
16	16		Segment file set identifier Contains a little-endian GUID (version 4)

2.1.3. Compression methods

Value	Identifier	Description
0	COMPRESSION_NONE	No compression
1	COMPRESSION_LZ	LZ compression (deflate, RFC195, zlib)
2	COMPRESSION_BZIP2	BZip2 compression

[ENCASE12] states that “COMPRESSION_NONE will be never used”, even so EnCase 7 does not even seem to supports this compression method and indicates the file header is corrupt.

Note at the moment EnCase 7 does not appear to provide an option to set the compression method to bzip2

2.2. Segment file extensions

2.2.1. EWF2-Ex01

The first segment file has the extension '.Ex01'.

- The next segment file has the extension '.Ex02.
- This will continue up to '.Ex99'.
- After which the next segment file has the extension '.ExAA'.
  - The next segment file has the extension '.ExAA'.
  - This will continue up to '.ExAZ'.
  - The next segment file has the extension '.ExBA'.
  - This will continue up to '.ExZZ'.
  - The next segment file has the extension '.EyAA '.
  - This will continue up to '.EzZZ'. (verify this; and then ?)

libewf supports extensions up to .EzZZ

2.2.2. EWF2-Lx01

The first segment file has the extension '.Lx01'.

- The next segment file has the extension '.Lx02.
- This will continue up to '.Lx99'.
- After which the next segment file has the extension '.LxAA'.
  - The next segment file has the extension '.LxAA'.
  - This will continue up to '.LxAZ'.
  - The next segment file has the extension '.LxBA'.
  - This will continue up to '.LxZZ'.
  - The next segment file has the extension '.LyAA '.
  - This will continue up to '.LzZZ'. (verify this; and then ?)

libewf supports extensions up to .LzZZ

3. The sections

The remainder of the segment file consists of sections. Every section ends with data that describes the section this will be referred to as the section descriptor. In contrast to EWF the section descriptor is at the end of the section and the section descriptor points to its previous section so the sections need to be read from back-to-front.

3.1. Section descriptor

The section descriptor consist of 64 bytes, it contains information about a specific section.

offset	**size**	**value**	**description**
0	4		Section type See section: 3.1.1 Section types
4	4		Data flags See section: 3.1.2 Data flags
8	8		Previous section offset The offset is relative from the start of the segment file 0 if there is no previous section
16	8		Data size

offset	size	value	description
24	4		Section descriptor size
28	4		Padding size The size of the padding after the data to make the sections 16-byte aligned
32	16		Data integrity hash Contains an MD5 of the data including padding If the data is encrypted the integrity hash is calculated of the encrypted data.
48	3 x 4 = 12	0	Padding Reserved empty values
60	4		Checksum Adler-32 of all the previous data within the section descriptor.

Note that the data size includes the padding size. The padding is not always at the end of the section data, it can also be after a table header followed by more section data.

The section can contain additional data not defined by the data size. This was seen in the sector data section of an EWF2 file that was aborted and restarted.

3.1.1. Section types

Value	Identifier	Description
0x00000001		Device information
0x00000002		Case data
0x00000003		Sector data
0x00000004		Sector table
0x00000005		Error table
0x00000006		Session table
0x00000007		Increment data
0x00000008		MD5 hash
0x00000009		SHA1 hash
0x0000000a		Restart data
0x0000000b		Encryption keys
0x0000000c		Memory extents table
0x0000000d		Next
0x0000000e		Final information
0x0000000f		Done
0x00000010		Analytical data

Value	Identifier	Description
0x00000020		Single files data Not defined by [ENCASE12]
0x00000021		Single files unknown table Not defined by [ENCASE12]
0x00000022		Single files MD5 hash table Not defined by [ENCASE12]
0x00000023		Single files unknown table Not defined by [ENCASE12]

3.1.2. Data flags

Value	Identifier	Description
0x00000001	MD5HASHED	The data integrity hash is set
0x00000002	ENCRYPTED	The data is encrypted

3.2. Device information

The device information section can be found:

- in every segment file after the file header in EWF2-Ex01
- in every segment file after section 0x00000020 in EWF2-Lx01 (TODO check multi Lx01)

When encryption is enabled the device information is encrypted.

The device information section contains a serialized file object string that consist of:

Line	Value	Description
1	1	Number of objects
2	“main”	Object name
3		Attribute tags
4		Attribute values
5		Empty line

3.2.1. Attribute tags

Identifier	Type	Description
sn	Text	Drive serial number EnCase 7 generated strange values for this in the test
md	Text	Drive model
lb	Text	Drive label
ts	Integer 64-bit	Number of sectors

Identifier	Type	Description
hs	Integer 64-bit	Number of sectors of the HPA protected sectors
dc	Integer 64-bit	Number of sectors of the DCO protected sectors
dt	Enumeration	Drive type See section: 3.2.2 Drive type
pid	Integer 32-bit	Process identifier Set when the memory of an individual process is acquired
rs	Integer 32-bit	Number of sectors of a PALM RAM device
ls	Integer 32-bit	Number of sectors in the SMART or ATA general logs The latter is returned by the ATA READ_LOG_EXT command
bp	Integer 32-bit	Bytes per sector
ph	Boolean	Is physical

3.2.2. Drive type

Value	Identifier	Description
a		RAM disk
c		Optical disc (CD-ROM)
f		Fixed
l		Single files (Logical evidence)
m		Memory
p		PALM
r		Removable

3.3. Case data

The case data section can be found:

- in every segment file after the device information section in EWF2-Ex01
- in every segment file after the file header in EWF2-Lx01 (TODO check multi Lx01)

When encryption is enabled the case data is encrypted.

The case data section contains a serialized file object string that consist of:

Line	Value	Description
1	1	Number of objects
2	“main”	Object name
3		Attribute tags
4		Attribute values
5		Empty line

3.3.1. Attribute tags

Identifier	Type	Description
nm	Text	Name Similar to Description in EWF version 1. libewf treats them as equivalent.
cn	Text	Case number
en	Text	Evidence number
ex	Text	Examiner name
nt	Text	Notes
av	Text	Application version The version of the application used for acquisition
os	Text	Operating system The operating system used used for acquisition
tt	Timestamp	Target time Date and time of the system used for acquisition in UTC Similar to Acquired date in EWF version 1
at	Timestamp	Actual time User provided date and time Similar to System date in EWF version 1 [ENCASE12] defines this as in UTC, but if this is user provided can UTC still be guaranteed?
tb	Integer 64-bit	Number of chunks (blocks)
cp	Integer 32-bit	Compression method See section: 2.1.3 Compression methods The value is empty, not 0 when the compression method is no compression Note that to match the compression method in the segment file header only be 16-bit of this value can be used
sb	Integer 32-bit	Number of sectors per chunk (block)
gr	Integer 32-bit	Error granularity
wb	Integer 32-bit	Write-blocker type

Note that EnCase 7 only provides the following number of sectors per chunk: 64, 128, 256, 512, 1024 which is referred by the application as block size. The thorough error granularity in EnCase 7 corresponds to 1 sector.

3.3.2. Write-blocker type

Value	Identifier	Description
1		FastBloc
2		Tableau

3.4. Sector data

The first sector data section can be found in every segment file after the case data section. Successive sector data sections are found after the sector table section.

When encryption is enabled the sector data is encrypted. TODO verify this.

The sector data is stored in chunks. [ENCASE12] states that each chunk must be stored 16-byte aligned and padded with 0-byte values if necessary. Although it can read non 16-byte aligned chunks.

If the sector compression method defined in case data section is set the chunk is compressed and the chunk data flag COMPRESSED is set. The checksum intrinsic to the compression method is used to verify the integrity of the chunk data. The chunk data flag CHECKSUMED is not set.

If a chunk is not compressed an Adler32 checksum of the data is stored after the chunk data and the chunk data flag CHECKSUMED is set.

Pattern fill seems to be a special case of compression and the COMPRESSED flag is set in combination with the PATTERNFILL flag. In EnCase pattern fill is not used when writing files and the compression is set to none. Libewf, when reading files, ignores the PATTERNFILL flag if the corresponding COMPRESSED flag is not set.

If the PATTERNFILL flag is set the chunk data size in the sector table entry is set to 0 and the chunk data offset contains a 64-bit pattern to fill the chunk data.

3.5. Sector table

The sector table is stored as an array of sector table entries (chunk descriptor or block offset). It defines the location of the chunk data in the segment file.

The sector table section can be found in every segment file after the sector data section. Every sector data section should be followed by a section table section.

When encryption is enabled the sector table is encrypted.

The sector table consists of:

- the sector table header
- an array of sector table entries
- the sector table footer

3.5.1. Sector table header

The sector table header is 20 bytes of size and consists of:

offset	size	value	description
0	8		First chunk number The first chunk number in the table 0 is the first chunk number for the entire image
8	4		Number of entries

offset	size	value	description
12	4	0	Padding Reserved empty values
16	4		Checksum Adler-32 of all the previous data within the sector table header.

The sector table header should be followed by 12 bytes of alignment padding.

TODO does EnCase support non-contiguous images?
Does EnCase write about 1600 entries per section ?

3.5.2. Sector table entry

A sector table entry is 16 bytes of size and consists of:

offset	size	value	description
0	8		Chunk data offset or fill pattern if corresponding flag is set
8	4		Chunk data size
12	4		Chunk data flags

3.5.3. Chunk data flags

Value	Identifier	Description
0x00000001	COMPRESSED	The chunk is compressed
0x00000002	CHECKSUMED	The chunk is followed by an Adler32 checksum
0x00000004	PATTERNFILL	The chunk is sparse and the value in the chunk data offset is used to fill the chunk data at run-time.

3.5.4. Sector table footer

The sector table footer is 4 bytes of size and consists of:

offset	size	value	description
0	4		Checksum Adler-32 of all the previous data within the sector table entries.

The sector table footer should be followed by 12 bytes of alignment padding.

3.6. Error table

The error table is stored as an array of error table entries. It defines the sector ranges that could not be read correctly during acquisition.

The error table section is optional, it does not need to be present. If it does it resides in the last segment file before the MD5 hash section.

When encryption is enabled the error table is encrypted. TODO verify this.

The error table consists of:

- the error table header
- an array of error table entries
- the error table footer

3.6.1. Error table header

The error table header is 20 bytes of size and consists of:

offset	**size**	**value**	**description**
0	4		Number of entries
4	12		Unknown (Empty value)
16	4		Checksum Adler-32 of all the previous data within the error table header.

The error table header should be followed by 12 bytes of alignment padding.

This differs from what [ENCASE12] specifies.

3.6.2. Error table entry

An error table entry is 16 bytes of size and consists of:

offset	**size**	**value**	**description**
0	8		Start sector
8	4		Number of sectors
12	4	0	Padding Reserved empty values

3.6.3. Error table footer

The error table footer is 4 bytes of size and consists of:

offset	**size**	**value**	**description**
0	4		Checksum Adler-32 of all the previous data within the array of error table entries.

The error table footer should be followed by 12 bytes of alignment padding.

3.7. Session table

The session table is stored as an array of session table entries. It defines the sessions of the optical disc stored in the set of segment files.

The session table section is optional, it does not need to be present. If it does it resides in the last segment file before the error table section.

When encryption is enabled the session table is encrypted. TODO verify this.

The session table consists of:

- the session table header
- an array of session table entries
- the session table footer

3.7.1. Session table header

The session table header is 20 bytes of size and consists of:

offset	**size**	**value**	**description**
0	4		Number of entries
4	12		Unknown (Empty value)
16	4		Checksum Adler-32 of all the previous data within the session table header.

The session table header should be followed by 12 bytes of alignment padding.

This differs from what [ENCASE12] specifies.

3.7.2. Session table entry

A session table entry is 32 bytes of size and consists of:

offset	**size**	**value**	**description**
0	8		First sector
8	4		Session flags
12	5 x 4	20	Padding Reserved empty values

For a CD the first session sector is stored as 16, although the actual session starts at sector 0. Could this value be overloaded to indicate the size of the reserved space between the start of the session and the ISO 9660 volume descriptor.

3.7.3. Session flags

Value	**Identifier**	**Description**
0x00000001		If set the track is an audio track otherwise the track is a data track

EnCase stores the data of audio tracks of an optical disc as 0-byte data with a sector size of 2048. It is therefore assumed that the format is only to support data tracks with a sector size of 2048.

3.7.4. Session table footer

The session table footer is 4 bytes of size and consists of:

offset	size	value	description
0	4		Checksum Adler-32 of all the previous data within the array of session table entries.

The session table footer should be followed by 12 bytes of alignment padding.

3.8. Increment data

The increment data section contains a serialized file object string that consist of:

TODO location in segment files, affected by encryption?

3.9. MD5 hash

The MD5 hash section contains the MD5 hash of the data stored in the set of segment files.

The MD5 hash section is optional, it does not need to be present. If it does it resides in the last segment file before the SHA1 hash section.

When encryption is enabled the MD5 hash is encrypted.

The MD5 hash data is 20 bytes of size and consists of:

offset	size	value	description
0	16		MD5 hash
16	4		Checksum Adler-32 of the MD5 hash.

The MD5 hash data should be followed by 12 bytes of alignment padding.

3.10. SHA1 hash

The SHA1 hash section contains the SHA1 hash of the data stored in the set of segment files.

The SHA1 hash section is optional, it does not need to be present. If it does it resides in the last segment file before the analytical data section.

When encryption is enabled the SHA1 hash is encrypted.

The SHA1 hash data is 24 bytes of size and consists of:

offset	size	value	description
0	20		SHA1 hash

offset	size	value	description
20	4		Checksum Adler-32 of the SHA1 hash.

The MD5 hash data should be followed by 8 bytes of alignment padding.

3.11. Restart data

The restart data section is optional, it does not need to be present. If it does it resides in the last segment file before the done section.

TODO is the restart data stored after or before the encryption keys?

Note that the “main” and “rl” object tags are not explicitly defined in the string.

The restart data section contains a serialized file object string that consist of:

Line	Value	Description
1		Unknown
2		Attribute tags
3		Unknown

```
1       1
p       d       sr      sp
0       1

0       1
5
0       0
                        1216
```

TODO

3.11.1. Attribute tags

Identifier	Type	Description
p	Integer 32-bit	Properties According to Guidance Software this value is used to store saved stated. In this context the value should always set to 0 but can contain other values in different contexts.
d	Timestamp	Start date and time Date and time the acquisition process was (re-)started
sr	Integer 64-bit	First sector The first sector acquired in the acquisition process
sp	Integer 64-bit	Last sector The last sector acquired in the acquisition process

3.12. Encryption keys

In EWF2 the data and some of the metadata can be encrypted, the encrypted keys section contains information necessary for decrypting the data.

The encryption keys section is optional, it does not need to be present. If it does it resides in the last segment file before the done section.

TODO is the encryption keys stored after or before the restart data?

The encryption keys is variable of size and consists of:

offset	**size**	**value**	**description**
0	4		Size Including the padding size
4	4		Unknown (Checksum?)
8	8	2	Unknown (Algorithm ID?) 2 => AES-256
16	...		Unknown (Encrypted data?)

The encryption keys should be followed by 12 bytes of alignment padding.

[ENCASE12] “Please refer to the document outlining the encryption support for Ex01 for further detail.” Where is this document?

3.13. Memory extents table

The memory extents table is stored as an array of memory extents table entries. It defines the extents of memory stored in the set of segment files.

TODO location in segment files, affected by encryption?
TODO does this table also come with a table header and footer ?

3.13.1. Memory extents table entry

A memory extents table entry is 16 bytes of size and consists of:

offset	**size**	**value**	**description**
0	8		Start page
8	8		Number of pages

3.14. Next

The next section is without data and marks the end of the segment file indicating more segment files are in the set. It should be the last section in a segment file, other than the last segment file.

3.15. Final information

[ENCASE12] defines this section as currently unused.

TODO location in segment files, affected by encryption?

3.16. Done

The done section is without data and marks the end of the segment file indicating this is the last segment file in the set. It should be the last section in the last segment file.

3.17. Analytical data

The analytical data section is optional, it does not need to be present. If it does it resides in the last segment file before the restart data section.

When encryption is enabled the analytical data is encrypted.

The analytical data section contains a serialized file object string that consist of:

Line	Value	Description
1	1	Number of objects
2	“main”	Object name
3		Attribute tags
4		Attribute values
5		Empty line

[ENCASE12] does not define the format of this section in detail.

3.17.1. Attribute tags

Identifier	Type	Description
tps	Integer 64-bit	The (total) number of bytes not written for use of pattern fill

3.18. Single files data

The single files data section is only present in EWF2-Lx01.

The single files data section can be found in the last segment file after the last sector table section. TODO what about non-closed LEF files.

This section has the section integrity hash set.

The single files data section contains a non-compressed serialized file object data which is similar to the EnCase 7 ltree data in EWF-L01.

3.19. 0x00000021 table

TODO

The 0x00000021 table consists of:

- the 0x00000021 table header
- an array of 0x00000021 table entries
- the 0x00000021 table footer

3.19.1. 0x00000021 table header

The 0x00000021 table header is 20 bytes of size and consists of:

offset	**size**	**value**	**description**
0	4		Number of entries
4	12		Unknown (Empty value)
16	4		Checksum Adler-32 of all the previous data within the 0x00000021 table header.

The 0x00000021 table header should be followed by 12 bytes of alignment padding.

3.19.2. 0x00000021 table entry

An 0x00000021 table entry is 8 bytes of size and consists of:

offset	**size**	**value**	**description**
0	8		TODO Start offset in the data?

3.19.3. 0x00000021 table footer

The 0x00000021 table footer is 4 bytes of size and consists of:

offset	**size**	**value**	**description**
0	4		Checksum Adler-32 of all the previous data within the array of 0x00000021 table entries.

The 0x00000021 table footer should be followed by 12 bytes of alignment padding.

3.20. Single files MD5 hash table

TODO

The single files MD5 hash table consists of:

- the single files MD5 hash table header
- an array of single files MD5 hash table entries
- the single files MD5 hash table footer

3.20.1. single files MD5 hash table header

The 0x00000021 table header is 20 bytes of size and consists of:

offset	size	value	description
0	4		Number of entries
4	12		Unknown (Empty value)
16	4		Checksum Adler-32 of all the previous data within the single files MD5 hash table header.

The single files MD5 hash table header should be followed by 12 bytes of alignment padding.

3.20.2. single files MD5 hash table entry

A single files MD5 hash table entry is 8 bytes of size and consists of:

offset	size	value	description
0	16		MD5 hash

3.20.3. single files MD5 hash table footer

The single files MD5 hash table footer is 4 bytes of size and consists of:

offset	size	value	description
0	4		Checksum Adler-32 of all the previous data within the array of single files MD5 hash table entries.

The single files MD5 hash table footer should be followed by 12 bytes of alignment padding.

3.21. 0x00000023 table

TODO

The 0x00000023 table consists of:
- the 0x00000023 table header
- an array of 0x00000023 table entries
- the 0x00000023 table footer

3.21.1. 0x00000023 table header

The 0x00000023 table header is 20 bytes of size and consists of:

offset	size	value	description
0	4		Number of entries
4	12		Unknown (Empty value)
16	4		Checksum Adler-32 of all the previous data within the

offset	size	value	description
			0x00000023 table header.

The 0x00000023 table header should be followed by 12 bytes of alignment padding.

3.21.2. 0x00000023 table entry

An 0x00000023 table entry is 8 bytes of size and consists of:

offset	size	value	description
0	8		TODO Start offset in the data?

3.21.3. 0x00000023 table footer

The 0x00000023 table footer is 4 bytes of size and consists of:

offset	size	value	description
0	4		Checksum Adler-32 of all the previous data within the array of 0x00000023 table entries.

The 0x00000023 table footer should be followed by 12 bytes of alignment padding.

Note if the number of table entries is odd the alignment padding is only 4 bytes.

4. Serialized file object data

The serialized file object data is stored as a compressed UTF-16 string with byte-order-mark. Commonly the string is encoded in little-endian. The compression method is defined in the file header of the segment file.

The serialized file object data consists of:

- the first line containing the number of objects in the string
- the object data

The file object serialization format uses the following special character values:

Value	Identifier	Description
0x0001		Escaped line feed
0x0002		Escaped carriage return
0x0003		Escaped tab
0x0009		Value delimiter
0x000a		Line delimiter

Note: [ENCASE12] states line feed (0x000d) as line delimiter this should be line feed (0x000a).

4.1. Object

An object consists of multiple lines:

Line	Value	Description
1		Object name
2		Attribute tags

4.2. Data types

Identifier	Type	Description
	Boolean	Boolean defined as: false => (empty) true => a single character containing “1”
	Enumeration	Single character that represent a value in an enumeration
	Array of Integer 64-bit	A space separated list of 64-bit unsigned integers
	Integer 32-bit	Decimal representation of a 32-bit unsigned integer
	Integer 64-bit	Decimal representation of a 64-bit unsigned integer
	Object	Sub or child object
	Text	Text EnCase limits the string to 3000 characters.
	Timestamp	Decimal representation of a 32-bit unsigned integer containing a timetamp, which contains the number of seconds since Jan 1, 1970 00:00:00 UTC.

5. Encryption

TODO:
Encryption keys section:

- the data integrity hash is set and the corresponding data flag in the section descriptor

Padding gets encrypted as well

Other sections:

- the data integrity hash is set and the corresponding data flag in the section descriptor
- the data is encrypted and the corresponding data flag in the section descriptor

This also applies to sections that contain no data. So what is the MD5 calculated on? The entire section without the MD5?

Password derivation/key file?
Unlocking the data?

6. Corruption scenarios

TODO

EWF2-Ex01, EWF2-Lx01
* corrupt chunk table
 - chunk data flags
 - with pattern fill
* corrupt chunk
  - uncompressed
  - compressed
* metadata

how does encase deal with out of order sector table sections?

7. Notes

7.1. .PublicKey file

```
00000000  41 43 46 09 0d 0a ff 00  02 00 00 00 65 6b 65 79  |ACF.........ekey|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000060  00 01 00 00 00 00 00 00  01 00 00 00 40 00 00 00  |............@...|
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000080  00 00 00 00 00 00 00 00  01 00 00 00 b0 03 00 00  |................|
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000f0  00 00 00 00 00 00 00 00  00 00 00 00 91 04 4e e2  |..............N.|
00000100  6b 65 79 73 00 00 00 00  01 00 00 00 32 cb 26 1d  |keys........2.&.|
00000110  40 01 00 00 00 00 00 00  ab 03 00 00 00 00 00 00  |@...............|
00000120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000130  00 00 00 00 00 00 00 00  00 00 00 00 ed 03 02 c4  |................|
00000140  01 9c 11 06 04 00 da 4b  9f d2 22 d1 4b ce 2f 3b  |.......K..".K./;|
...
```

Appendix A. References

[ENCASE12]
Title: EnCase Evidence File Format Version 2
Author(s): Guidance Software
Date: January 2012
URL: http://www.guidancesoftware.com/

Appendix B. GNU Free Documentation License

Version 1.3, 3 November 2008

```
Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.
```

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at

least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.