Hacha file format specification

Files

File.0, File.1, File.2, etc.
File.0: this is the first segment. It contains a header with useful stuff and the first chunk of data.
The rest are segments containing all the data.

Header

There are four blocks containing stuff, delimited by groups of 5 characters "?".
The **first block** are 4 bytes and can contain three different things:
- Older formats: length of rest of header
- Newer formats:
  - Nothing, usually ASCII 7, plus 3 nulls.
  - CRC, the crc checksum of the original file.

The **second block** contains the name of the file.
The **third block** contains the size of the file (ASCII).
The **fourth block** contains the size of segments.
For example, in a hacha file with CRC we'll find the following structure (ASCII):

```
?????####?????Filename?????55555?????1111?????
```

The rest of the data in the first segment is a chunk equivalent to the size of the segment. In this example, the first file ("Filename.0") will have a size of 1059 bytes: 48 bytes of header + the first 1111 bytes of the original file data.
The last segment will contain the rest of the data. In this example, we have the first segment (1059 bytes) + 49 segments (* 1111 bytes) + last segment (5 bytes) = header + 55555 bytes.

Example
We have a file called “life.txt” whose contents are the following: “The life is red”.
We will segment it in chunks of 6 bytes.

“life.0”

	Bytes	Meaning
	?????	delimiter
new format	****	nothing (hex “07000000”)
new w/ CRC32	****	CRC32 (hex “FF2F3C21”)
old format	****	length of rest of header (ascii 31 + 3 ascii 0)
	?????	delimiter
	life.txt	original file name
	?????	delimiter
	17	size of original file
	?????	delimiter
	6	size of chunks
	?????	delimiter
	The li	6 bytes of data (first segment)
?????****?????life.txt?????17?????6?????The li		

“life.1”

fe is	6 more bytes

“life.2”

red	3 bytes (rest of data)

HJSplit file format specification

Files

File.001, File.002, File.003, etc.

Header

There is not header. Data is distributed in the different segments.

Example

We have the file “foo.txt”, containing the data “We are the champions”. We will split it in segments of 10 bytes.

“foo.txt.001”

We are the	10 first bytes

“foo.txt.002”

champions	9 bytes (rest of data)

Usually, the segments will keep the extension of the original archive, so it can be used later, when joining them.

WinSplit file format specification

Files

File.000, File.001, File.002, etc. + File.Split

Header

There is not header. Data is distributed in the different segments. There is a ".Split" file which keeps relevant info.

Example

We have the file "foo.txt", containing the data "We are the champions". We will split it in segments of 10 bytes.

"foo.000"

We are the	10 first bytes

"foo.001"

champions	9 bytes (rest of data)

Now, we have an information file, called as the original one, using the extension ".Split":

"foo.Split"

```
<ORIGINALNAME>foo.txt[CRLF]
<#FILES>2[CRLF]
<SIZE>19
```

"[CRLF]" means that you'll find a CR (ASCII 13) and a LF (ASCII 10), typical windows newline mark.

The format of this ".Split" file is self explaining. In each line you'll find the name of the original file, the number of segments and the total size in bytes of the original file.

At join time, WinSplit will use this file.

FastSplit file format specification

Files

File.001, File.002, File.003, etc.

Header

There is not header. Data is distributed in the different segments. However, there is a first file ".000", which keeps relevant info.

Example

We have the file "foo.txt", containing the data "We are the champions". We will split it in segments of 10 bytes.

"foo.001"

We are the	10 first bytes

"foo.002"

champions	9 bytes (rest of data)

Now, we have a file called "foo.000", which keeps some information about the original file. This file is used by FastSplit to join the segments:

"foo.000"

<table><tr><td>19|2|.txt[CRLF]</td></tr></table>

The structure of this file meand the following: size in bytes of the original file, separator ("|"), number of segments, separator ("|"), and extension of the original file; plus a CRLF, which is (ASCII 13 + ASCII 10).

MasterSplitter file format specification

Files

File.001, File.002, File.003, etc.

Header

There is not header. Data is distributed in the different segments. However, there is a first file ".000", which keeps relevant info.

Example

We have the file "foo.txt", containing the data "We are the champions". We will split it in segments of 10 bytes.

"foo.txt.001"

We are the	10 first bytes

"foo.txt.002"

champions	9 bytes (rest of data)

Now, we have a file called "foo.txt.000", which keeps some information about the original file. This file is used by MasterSplitter to join the segments:

-The first byte is the length of the original file name.

-Next bytes are the DOS name of the original file.

-Now, there are 12 bytes, which are a sequence of three bytes, which I don't know what mean.

-Now, we find 5 nulls (ASCII 0).

-Now, 4 bytes more, which I don't know what mean.

-Now, 1 byte, the number of segments.

-Now, a bunch of bytes which complete the 50 bytes header, and also don't know what mean.

Perhaps some of these bytes contain the checksum of the original file, but I didn't check it (it's so bored).