Foster Package Format

Version 2

Introduction

The entire Foster package format is redesigned in order to store information more efficiently and safely. Unlike the previous version, this version contains the name, the creation date, last modified date and size of each file in a new section named “File Info Table” (FIT). This table is uncompressed and the length of it is stored in the header. Since Foster stores the information about files without the data somewhere different, it should take significantly small amount of time to get the information just to see the file names and the sizes. This version of the package format also allows Foster to encrypt the information and extract only a single file since none of the files are encrypted and/or compressed together.

Due to new features of the packaging method, the file size might increase a little.

Structure

Foster packages (file ends with .fp file extension) are built with 3 parts; The Header, The File Information Table and The Content.
The Header contains the information about the package itself.
The File Information Table contains the file information such as the name and size.
The Content contains the data un/compressed and/or un/encrypted data of each file.
Foster still uses UTF-16 Little Endian encoding for all texts and all texts start with 0x02 byte and ends with 0x03 byte.

The Header

The header is similar to the version 1 header.
The header contains these informations (in order):

- Compression Algorithm (Text)
- Foster Package Version (Int32, 4 bytes long)
- The Flags Byte (1 byte long)
- The FIT size (Int64, 8 bytes long)
- The Encryption Algorithm (Text)
- Information about the encryption (if the encryption algorithm is not *none*) (Byte Array)

Note: No passphrase for decrypting the package must be stored inside the package.

The Flags Byte

Here's a table of hexadecimal values associated with the flags:

Value	Flag(s)
0x00	None
0x01	Delta Package
0x02	Foster File

Note: Later versions might add new flags.

The File Information Table (FIT)

The information about all files without the data is stored in here. There can only one section as FIT even on multi-packages as there's no need since the size of this table ranges from 0 to the 64-bit integer limit.

The information that is stored is the file name, the file size and the pointer which tells where does the file end on the file.

Since we already know where does the content starts and the index of the file and the where does it end on table, we can figure out which part of the content does store the file we need.

On Delta packages, this table does not exist and every single delta operation is located inside the "Content".

The Content

The Content contains the data of each file compressed and encrypted next to each other. Each compression and encryption is unique and file gets separated for each 1GiB if the encryption and/or compression algorithm has a limit. For example, a 4GiB file that will be encrypted with AES won't be separated but if that file is going to be compressed with GZip, it will be separated into 4 chunks and processed that way.

Packing

At first, the header will be generated. Then a half-done FIT will be generated. Then each file will be divided into 1GiB chunks and compressed and encrypted. The end of the compressed file will be written to FIT and finally, the FIT will be regenerated again.

Unpacking

At first, the header will be read. Then the FIT will be read. Then starting from the beginning of the content, each file will be first decrypted and uncompressed and jointed back. Later then the file information is written back to file.