Part I: FLP Format & Events

FLP is a binary format used by Image-Line FL Studio, a music production software, to store project files. Instead of using C-style structs entirely, the FLP format has evolved from what once was a MIDI-like format to a really bad and messy combination of Type-length-value encoded "events" and structs.

Specification

An FLP file contains of basically 2 sections or "chunks", one is the header and other is the "data" section, which contains all the "events".

Header chunk

C / C++ | **Python**

```python
class Header:
    magic: str
    size: int
    format: int
    num_channels: int
    ppq: int
```

> See also
>
> `Project.format`, `Project.channel_count`, `Project.ppq`

Data chunk

```c
struct {
    char magic[4];  // 'FLdt'
    uint32_t size;  // Total combined size of events
    void* events;   // Event data
}
```

Event

An event can be thought of as a "flattened" `dict` of attributes composing a class. It can *roughly* be represented as:

C / C++ | **Python**

latest

```
class Event:
    type: int
    value: object
```

Types

There are basically 4 kinds of events depending on the range of `type`:

Event ID	Size of `value` | Total event size	
0-63	1 byte	1 + 1 = **2**
64-127	2 bytes	1 + 2 = **3**
128-191	4 bytes	1 + 4 = **5**
192-255	Length prefixed	>= 2

Note

Length prefixed events

These events store the length of the `value` they contain after `type` in a varint. It can be considered as the only true TLV encoded event type.

```
struct {
    uint8_t type;     // 192-255
    uint8_t* length;  // varint
    void* value;      // string, struct or subevent
}
```

It should be clearer by now how the FLP format is a misfit for the data it represents.

Representation

Event IDs 0-191 are used for storing fixed size data like integers, floats and booleans. IDs from 192-255 are used for storing structs, subevents and strings.