



Reversing FLP format

You should first take a look at [what events are](#). A decent knowledge of the topics mentioned there as well as Python itself is assumed.

One could use a hex editor, but its too tedious. I have a simpler solution:

FLP Edit [FL 20.8.4.flp]		—	□	×
File Operations Compatibility		Inspector Reload		
▼ Misc				
BeatDivision	96			
> DataChunkID	Char[] Array			
DataChunkLength	190067			
EventCount	3829			
▼ Events	FLP_Event[] Array			
> [0]	ID_Project_Version = "20.8.4.2576" [11 chars]			
> [1]	159 = 2576 [4b]			
> [2]	ID_Project_Registered = 1 [1b]			
> [3]	37 = 1 [1b]			
> [4]	ID_Project_RegName = "pbckeg_["^VR]WX[a=A;@:959" [24 chars]			
> [5]	ID_Project_FineTempo = 69420 [4b]			
> [6]	ID_Project_Current_Pattern = 5 [2b]			
> [7]	ID_Project_LoopActive = 1 [1b]			
> [8]	ID_Project_Shuffle = 0 [1b]			
> [9]	ID_Project_MainPitch = 0 [2b]			
> [10]	ID_Project_TimeSig_Num = 4 [1b]			
> [11]	ID_Project_TimeSig_Beat = 4 [1b]			
> [12]	ID_Project_129_NewUnknown = 1 [1b]			
> [13]	ID_Playlist_Selection Byte Array [8b]			
> [14]	ID_Project_PanVolTab = 0 [1b]			
> [15]	ID_Project_TruncateClipNotes = 1 [1b]			
> [16]	ID_Project_ShowInfo = 1 [1b]			
> [17]	ID_Project_Title = "PyFLP Test FLP" [14 chars]			
> [18]	ID_Project_Genre = "Testing..." [10 chars]			
> [19]	ID_Project_Author = "demberto" [8 chars]			
> [20]	ID_Project_DataPath = "" [0 chars]			
> [21]	ID_Project_Comment = "This is a testing FLP used by PyFLP - An FL Studio project file parser. Notes for contributors:1. M			
> [22]	ID_Project_URL = "https://github.com/demberto/PyFLP" [33 chars]			
> [23]	ID_Project_Time_StartTime:16-09-2022 20:47:12 , Work Time:02:29:21.9970000			
> [24]	ID_ChannelFilter_Name = "Audio" [5 chars]			
> [25]	ID_ChannelFilter_Name = "Generators" [10 chars]			
> [26]	ID_ChannelFilter_Name = "Unsorted" [8 chars]			
> [27]	ID_ChannelFilter_CurrentNum = -1 [4b]			
> [28]	FLP_CtrlRecChan Byte Array [0b]			
> [29]	ID_Pattern_New = 3 [2b]			
> [30]	ID_Pattern_Note_Events = 48 Notes => Rack Channel(s): 9, 5, 1, 10, 6, 2, 11, 7, 3, 12, 8, 4			
> [31]	FLP_Ctrl_Remote_MIDI Structure [5 DWORDs]			
> [32]	FLP_Ctrl_Remote_MIDI Structure [5 DWORDs]			
> [33]	FLP_Ctrl_Remote_MIDI Structure [5 DWORDs]			
> [34]	ID_Channel_New = 0 [2b]			
> [35]	ID_Channel_Type = 2 [1b]			
> [36]	ID_Plugin_Default_Name = "BooBass" [7 chars]			
> [37]	ID_Plugin_New Byte Array [52b]			
> [38]	ID_Plugin_Name = "BooBass" [7 chars]			
> [39]	ID_Plugin_Icon = 0 [4b]			
> [40]	ID_Plugin_Color = #5C656A			
> [41]	ID_Plugin_Parameters Byte Array [16b]			

FLPEdit, an event view for FLP (and related formats) files.

Download it [here](#).

This is an unmaintained software, written actually in C#. Event ID names are different but the file attached above has source code as well. Check the `FLPFileFormat/FLP_File.cs` file for a list of event IDs and compare them to the ones from `pyflp._events.EventEnum` in PyFLP.

Events

Which event needs to be inspected can only be understood when you observe the ordering of events,

whether they occur for default values or not as well as a general knowledge of new features and changes occurring inside FL Studio.

Check [this discussion](#) for a list of unknown / undiscovered events.

Struct fields

Structs whose field names are prefixed with a `_u` are undiscovered fields. Wherever possible, I have added helpful comments right next to them.

Also, throughout PyFLP's codebase, there are a number of `# TODO` comments. Some of these can have additional information about them.

My workflow

1. Create a new test FLP or a preset and save it.
2. Parse the file with PyFLP and record the initial values.
3. Turn knobs / faders all the way to their extremes, save and repeat (2)

Hint

WhatsNew.rtf

FL Studio's changelog file `WhatsNew.rtf` exists in its install folder. It is a very helpful source for understanding which features were added when.