

Chart File Format Specifications

Authored by FireFox

Contents

1. Introduction and History
2. Chart Format
 - a. Header Tags and Data Sections
 - b. [Song]
 - i. Mandatory Metadata - Resolution
 - ii. Other Metadata
 - c. Tick = Key Value [...] Layout
 - d. Tick System and Time Synchronisation
 - e. [SyncTrack]
 - i. B - BPM
 - ii. TS - Time Signature
 - iii. A - Anchors
 - f. [Events]
 - i. E - Global Events
 1. Text Events
 2. Sections
 3. Lyrics
 - g. Tracks
 - i. Difficulties/Instruments
 - ii. N - Notes
 - iii. S - Special
 1. S 2 ## - Starpower
 2. S 0 ## and S 1 ## - GH1/2 Co-op
 - iv. E - Track Events
3. Trivia

Introduction and History

The .chart file format is a text-based format designed to store per-song game data originally for Guitar Hero 1/2 custom songs, and later various other rhythm games including Guitar Hero 3 via the [Guitar Hero Three Control Panel \(also known as GHTCP\)](#) and [Clone Hero](#).

In the early days of Guitar Hero, a user by the name of Turkeyman as well as several members of the [Score Hero](#) community worked together to reverse engineer the original Guitar

Hero 1 archive format. Soonafter, a user by the name Katamakel developed a user interface to rebuild these archives while Turkeyman worked on an editor to modify the data contained within said archives, which later turned into the [Feedback Editor](#). Support for this editor dropped at around the release of Guitar Hero 3 due to Turkeyman developing the program into a fully standalone game. Remnants of this can be seen with the inclusion of the Drums and Keys track, which were implemented before Guitar Hero: World Tour and Rock Band were announced. There was also a secondary version of the .chart file format in development for this game known as .chart-v2. This format would have addressed the current limitations that came with designing the original .chart format for GH1/2, but were never adopted by the community due to it only being available in Turkeyman's extended game, which was never released. Modding for Guitar Hero 3 also hit a standstill at this time due to increased security within the game's code. It wasn't until the PC release of GH3 that the modding scene picked up again, spawning GHTCP which became the main modding tool for the game. Soon after GH:WT and Rock Band were announced, development for Feedback dropped entirely.

The format was later updated in order to support [GH3 mods by ExileLord](#), such as Tap notes and Open notes. Due to Feedback's outdated nature, these mechanics had to be implemented using workarounds that were extremely tedious and prone to errors and crashes. It was also around this time that [Moonscraper Chart Editor](#) began development by FireFox in order to resupport the .chart format and the new mods with it, as well as make the charting process easier and more accessible.

Yet still, some people still found exhaustion in the entire modding process for GH3, and soon a fangame called [Clone Hero](#) began development by Srylain, which read the entire .chart file format directly into the game. This made playing custom Guitar Hero songs much more accessible as it avoided any complicated game modification process completely redundant. Clone Hero would go on to implement its own mechanics into the .chart file format, such as the idea for lyrics by Mdsitton and Guitar Hero: Live track data.

**Special thanks to Turkeyman for providing the information on the early history of the .chart file format*

The latest full feature set of the .chart file format is currently supported by:

- [Moonscraper Chart Editor](#)
- [Editor on Fire](#)
- [Clone Hero](#)

A legacy feature set is also supported by

- Feedback Editor
- Guitar Hero 2
- Guitar Hero 3
- GH3+ mods

Chart Format

Header Tags and Data Sections

Similar to the .ini format, data within chart files is divided up into sections. To create a section, begin with a header tag contained within square brackets []. On a new line open a curly bracket {. Start writing the data on a new line. Close the section by placing a closed curly bracket } on a new line.

Chart files always have a [Song] section to define metadata, a [SyncTrack] section to set up time synchronisation events, and an [Events] section that lists various types of miscellaneous events. The rest of the sections are used to define individual tracks of various difficulties and instruments.

[Song]

The [Song] tag is the first line of any chart file. It lists various metadata values used to define the chart. Metadata options are written in the format

[KEY] = [VALUE]

Values may or may not be surrounded by quotation marks to indicate a string. If not, they may be a number type or specific enum types depending on the key (see "Other Metadata").

Mandatory Metadata - Resolution

Resolution is the amount of ticks per quarter note of the entire chart. In layman's terms, it's the density of ticks that can fit within a given timeframe. It is also known as ["ticks per quarter note"](#) or ["pulses per quarter note"](#) for the MIDI file format.

The standard values used are either 192 or 480. Charts designed for Guitar Hero games usually use a resolution of 192, however charts that were converted from a Rock Band midi file, or chart that will be converted to a Rock Band midi will use a resolution of 480.

How this is used to determine the position of a particular event will be further discussed in the section "Tick System and time synchronisation".

Other Metadata

Other metadata options exist to set up specific options that were originally designed for Guitar Hero/Feedback/GHTCP era, but various others have been added since. Most of these options are optional. These options include:

- Name = "5000 Robots"
 - The name of the song. Various games may read the song name from here, the name of the file, or the name of the folder the file is listed in.
- Artist = "TheEruptionOffer"

- The musical artist of the song.
- Album = "Get Smoked"
 - The album the song came from
- Year = ", 2018"
 - The year the song came out. This is listed with a comma and space at the start as it saved time importing into GHTCP
- Charter = "TheEruptionOffer"
 - The person or people who have charted out this particular chart file.
- Offset = 0
 - Offset is the amount of seconds in time because tick position 0 is reached.
[“Basically, this is how far into the fretboard you need to go before the first note is played.”](#) **It is HIGHLY recommended to ignore this parameter and leave this value at 0** as there are better methods of accounting for this delay in your charts and is very much a legacy option
- Resolution = 192
 - See above
- Player2 = bass
 - Used by GH3 to define whether the co-op guitar chart should be a bass or rhythm player. Note that there are no quotation marks surrounding this value.
- Difficulty = 0
 - The perceived difficulty of the song. There is no standard for this, up to the charter's discretion
- PreviewStart = 0
- PreviewEnd = 0
- Genre = "rock"
- MediaType = "cd"
- Audio streams
 - Defines the file location of the audio tracks to play during the game. Some games will not read these and instead choose to search for specifically-named audio files in the same directory as the chart file itself. Audio streams can include:
 - MusicStream = "5000 Robots.ogg"
 - GuitarStream = "guitar.ogg"
 - RhythmStream = "rhythm.ogg"
 - BassStream = "bass.ogg"
 - DrumStream = "drums_1.ogg"
 - Drum2Stream = "drums_2.ogg"
 - Drum3Stream = "drums_3.ogg"
 - Drum4Stream = "drums_4.ogg"
 - VocalStream = "vocals.ogg"
 - KeysStream = "keys.ogg"
 - CrowdStream = "crowd.ogg"

Various other game-specific metadata may be stored in a file called "song.ini". Examples of games that use this file are Phase Shift and Clone Hero but are outside the scope of this document.

Tick = Key Value [...] Layout

The primary layout for the rest of the chart file sections are laid out in the format Tick = Key Value + other values depending on the key.

Tick- A positive integer used to represent the position of an event outside, ignoring any time synchronisation from bpm events. This is NOT the position in real time. See [Tick System and time synchronisation] for more information.

Key- An id value value used to represent a specific event. An example of this is the note event, which uses a key of 'N' to define itself.

Value [...] - One or more values for the event as specified by the key.

Tick System and Time Synchronisation

The calculation of time in seconds from one tick position to the next at a constant BPM is defined as follows-

```
(tickEnd - tickStart) / resolution * 60.0 (seconds per minute) / bpm
```

Therefore to calculate the time any event takes place requires precalculation of the times between all the BPM events that come before it. BPM events are defined in the [SyncTrack] section as defined below.

[SyncTrack]

B - BPM

The "B" key for an event is used to define a BPM marker, or [beats per minute](#).

It has 1 value, a positive integer which is equal to the bpm rate * 1000. For example, 120 bpm will be stored as [0 = B 120000]. This means that bpm is limited to a decimal precision of 3 with this format.

TS - Time Signature

The "TS" key represents a [Time Signature](#) event. Time signatures in various Guitar Hero-style games are often used to draw beat lines on the highway, or to determine how long the Starpower mechanic will last for.

Time Signatures must contain at least 1 value, but may contain up to 2. The first value represents the numerator value of a given time signature. The optional second parameter is

stored as the **logarithm of the denominator in base 2**. If there is no denominator defined, the value is 4 by default.

For example, if you wanted to store a time signature of 3/8, it would be written as [0 = TS 3 3].

A - Anchors

The last part of the synctrack section is the rarely used “anchor” event. These are always paired up with BPM markers and are only used by chart editors, games reading this format can ignore this event.

Anchors have 1 value. This value is the real-time position of the paired BPM stored in microseconds. It is used to lock a certain bpm to a certain time, so that if you start editing previous synctrack events, the bpm should be readjusted to lock back to that exact point in time to prevent the notes ahead of it from going out of sync with the music.

[Events]

E - Global Events

Text Events

These events are simply a string to indicate something at a certain tick.

They always contain 1 value, and that value is always surrounded by quotes “”. Any quote characters within the event phrase will result in the event loading incorrectly, depending on the program loading it.

Sections

Sections are a subtype of text events and are commonly used by games in practice modes to outline a certain section of notes to play over and over.

It's value is always prefixed with the string “section”, followed by the title of the section. For example, a section labeled “Solo 1” would be stored as [112118 = E “section Solo 1”].

Lyrics

Similar to sections, lyrics are simply the same but it has the prefix “lyric” instead of “section”. For example, a lyric would be stored as [112118 = E “lyric OOOoooo”]. Lyrics events according to the Clone Hero spec need to start with a “phrase_start” event. A “phrase_end” event isn't strictly required as a new “phrase_start” event will automatically end the previous phase.

Tracks

Difficulties/Instruments

The following lists the available combinations of difficulties and instruments. The header tag is a combination of these two components to form Header Tag = Difficulty + Instrument.

Difficulties:

- Easy
- Medium
- Hard
- Expert

Instruments

- Single
- DoubleGuitar
- DoubleBass
- DoubleRhythm
- Drums
- Keyboard
- GHLGuitar
- GHLBass
- GHLCoop
- GHLRhythm

Examples:

- [ExpertSingle]
- [MediumDrums]

** The track "SingleBass" may be present in some older charts. This was technically supported by the Feedback editor but was never used in an actual game and thus is considered to not be supported.*

N - Notes

Notes are indicated by the "N" key and have 2 values. The first value is generally the fret number. The second value is the length, or "sustain" of the note in ticks. Some note flags may also use the "N" key because some bright spark thought it was a brilliant idea to use N 5 0 as to flag all notes on the same tick as a forced note and the pattern stuck.

Unconfirmed, but going forward we plan to divide the N key up into clear and separate groups for different game modes, in order to prevent issues like having Black Lane 2 have a fret

value of 4 and Black Lane 3 have a non-sequential fret value of 8. The division will proceed as follows:

"N 0 ####" through "N 31 ####" for standard GH3 notes
"N 32 ####" through "N 63 ####" for Expert+ notes (like double kick) or other extra GH related mods
"N 64 ####" through "N 95 ####" for Rock Band/Pro Drums stuff (like cymbal/tom toggles)
"N 96 ####" through "N 127 ####" for any CH related stuff they want to add

If a new game or game mode comes along that needs custom N events they should reserve the next set of 32.

The following is a list of fret values and their meaning in the context of their track/game mode:

Guitar + other non-listed instruments

0 ## - Lane 1, sustain ##
1 ## - Lane 2, sustain ##
2 ## - Lane 3, sustain ##
3 ## - Lane 4, sustain ##
4 ## - Lane 5, sustain ##
5 0 - Forced Flag
6 0 - Tap Flag
7 ## - Open, sustain ##

Drums - *Note that sustain here is dependent on the game*

0 ## - Open/Kick, sustain ##
1 ## - Lane 1, sustain ##
2 ## - Lane 2, sustain ##
3 ## - Lane 3, sustain ##
4 ## - Lane 4, sustain ##
5 ## - Lane 5, sustain ##
32 ## - Double Kick, sustain ##

34 0 - Lane 1 accent flag
35 0 - Lane 2 accent flag
36 0 - Lane 3 accent flag
37 0 - Lane 4 accent flag
38 0 - Lane 5 accent flag

40 0 - Lane 1 ghost flag
41 0 - Lane 2 ghost flag

42 0 - Lane 3 ghost flag
43 0 - Lane 4 ghost flag
44 0 - Lane 5 ghost flag

66 0 - Pro Drums Cymbal toggle Lane 2
67 0 - Pro Drums Cymbal toggle Lane 3
68 0 - Pro Drums Cymbal toggle Lane 4

**Note that notes are toms by default and are manually toggled to cymbals. This is the opposite of how the RBN midi spec handles cymbals.*

Guitar Hero Live

0 ## - White Lane 1, sustain ##
1 ## - White Lane 2, sustain ##
2 ## - White Lane 3, sustain ##
3 ## - Black Lane 1, sustain ##
4 ## - Black Lane 2, sustain ##
5 0 - Forced Flag
6 0 - Tap Flag
7 ## - Open, sustain ##
8 ## - Black Lane 3, sustain ##

S - Special

S 2 ## - Starpower

Within Guitar Hero style games, this event is used to dictate the position and range of a starpower phase. It has 2 values, the first is 2 to flag it as a starpower special marker, and the second value indicates it's length in ticks, the same way a note sustain would.

S 0 ## and S 1 ## - GH1/2 Co-op

S 0 ## and S 1 ## may be present in some older charts. This was used to mark out a Guitar Hero 1 and 2 co-op gamemode that was later dropped by future games. Notes marked with S 0 ## were given to the primary player to play while notes marked S 1 ## were given to the secondary player. It is recommended to avoid using either of these markers in modern charting.

S 64 ## - Drum Fills

S 64 is used to represent drum fills for activating sp on drum tracks for Rock band style overdrive management. These have length and should not overflow with S 2 ## markers as you cannot get starpower/overdrive at the same time as a fill. See the [rbn authoring docs](#) for further reference. Note that although midi authoring uses 5 notes to display fills, the chart format only

displays the same result using one marker as there is only one valid combination of the 5 notes markers in midi that were ever used.

S 65 ## - Drum Rolls (Standard)

Defines the area for a single lane drum roll. If there are notes across multiple lanes, the roll is only active on the lane of the first note that's in the roll, or the left most lane if there are multiple notes on the same tick. See the [rbn authoring docs](#) for further reference.

S 66 ## - Drum Rolls (Special)

Defines the area for double lane drum rolls. If there are notes across multiple lanes, the roll is only active on the lanes of the first two notes that's in the roll, or the left most lanes if there are multiple notes on the same tick. See the [rbn authoring docs](#) for further reference.

E - Track Events

The same as Global Events noted above, but only apply to the track they are positioned in. Hardly used nowadays, but a good resource to use for game-specific text-based events. The main difference from Global Events is that they are written out without quotation marks surrounding the event. Thus track events only have 1 value and the event cannot have spaces in it. Example - [168960 = E soloend]

Worth noting that disco flip events reuse the same format as listed in the [RBN docs](#).

Trivia

- Some of the original Scorehero members who worked on the charting mods were hired by Neversoft to work on GH3/4.
- It is also believed that due to struggles with internal charting tools, Feedback was used by some Neversoft developers themselves to improve their workflow. When Turkeyman approached some of these developers due to Feedback not being licensed for commercial use, they "got angry at [him] for enabling a community to mod their game and encourage piracy".