Serato External Metadata

Specification

# Contents

Contents 2

Overview 3

Location 3

Overview file 4

Metadata file 5

<Name> 5

<Artist> 5

<Album> 6

<Genere> 6

<Length> 6

<Size> 6

<ByteSize> 6

<Bitrate> 6

<Samplerate> 7

<BPM> 7

<Comments> 7

<Key> 7

<Grouping> 7

<Remixer> 8

<Composer> 8

<Year> 8

<Rating> 8

<GaindB> 8

<AutoGain> 9

<AutoGainAdj> 9

<Label> 9

<Markers> 9

<Loops> 10

<BeatGrid> 11

# Overview

This document specifies the song metadata storage format that Serato software uses when it is not possible or desired to embed the metadata into the media file itself. Serato software will embed metadata whenever it is able to do so. For example when dealing with writable mp3 files, metadata will be stored in the ID3 tags. At least four reasons exist why this is sometimes not possible:

* The format does not support embedding metadata
* Serato software doesn’t support embedding metadata for that particular file format
* The file is not writable
* A special file format, where Serato has chosen not to embed the data

When embedded metadata is not possible or desired, the software provides a way of saving metadata information separately from the media file. This document describes the location and file format used for this system.

# Location

Serato software stores the external metadata in a folder inside the \_Serato\_ library folder. As described in the Library Specification document, this folder is available on every drive attached to the system. On the primary drive the metadata folder is stored on a per-user basis:

Or:

|  |
| --- |
| C:\Users\[username]\Documents\Music\\_Serato\_\**Metadata** |

|  |
| --- |
| /Users/[username]/Music/\_Serato\_/**Metadata** |

On an external/secondary drives this is off the root folder:

|  |
| --- |
| D:\\_Serato\_\**Metadata** |

Or:

|  |
| --- |
| /Volumes/MyDrive/Music/\_Serato\_/**Metadata** |

For each media file that metadata is stored for, two files are saved in a location off the above-mentioned directory. The two files are:

|  |  |  |
| --- | --- | --- |
| File | Extension | Description |
| Metadata | .xml | This file contains the common metadata for a media file, such as the title, artist, bpm, loops…etc. |
| Overview | .ovb | This file contains the overview of the audio in a special Serato format |

The exact location of these files, given the path of the original media file, is a concatenation of the above root location, the absolute path of the original media file, and the extension (which is either .xml or .ovb).

For example if the original media file sits on a Windows machine in this location:

|  |
| --- |
| C:\SomeFolder\SomeSong.mp4 |

The corresponding metadata and overview files will be located here:

And here:

|  |
| --- |
| C:\Users\[username]\Documents\Music\\_Serato\_\Metadata\SomeFolder\SomeSong.mp4.**xml** |

|  |
| --- |
| C:\Users\[username]\Documents\Music\\_Serato\_\Metadata\SomeFolder\SomeSong.mp4.**ovb** |

# Overview file

The overview file stores a graphical representation of the file’s audio at a low resolution. The software gathers this information when the file is analysed or loaded onto a virtual deck, and then saves it on disk it for quick loading. Here is a graphical representation of one such overview.

Macintosh HD:Users:goran:Desktop:Overview.png

While the storage format of this file is quite simple, the process of arriving at the result (from raw audio input) is complex. Therefore it is impractical to try to replicate the process, and thus of little use to describe the file format either. If you are trying to rebuild the Serato library from external sources, it is best to let the software go through the file analysis process once, harvest (back up) the results, and then recall the files when you wish to rebuild the library.

You may notice that in the software the cue points are also displayed on this overview. The cue points are not stored in this file. They are subsequently overlaid at runtime. The track gain levels set in Serato software also do not affect this overview. So in general, it is sufficient to go through the analysis process once, the first time a file is imported into a library. However if the actual audio stored in the file had somehow changed (for example if it was run through an audio level normalization process external to the Serato application), then it would be necessary to repeat the analysis and subsequent harvesting for accurate representation.

# Metadata file

This file format stores certain metadata information about a media file. The file is automatically written out by Serato software upon analysis, or upon loading a file to a deck, or when the user changes a certain field such as the comment field.

The metadata file is a standard XML 1.0 file, **UTF-8** encoded. However the encoding is not specified at the start of the file. In fact, the file has no xml declaration at the start.

All files have the SSLMetadata as the root element. All other elements must be embedded inside this element, for example:

|  |
| --- |
| <SSLMetadata>  <Name>Fifth Symphony</Name>  <Artist>Beethoven</Artist>  <BPM>108.00</BPM>  …  </SSLMetadata> |

The following elements/tags are allowed inside SSLMetadata:

### <Name>

**Description:** The title of the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Name>Poker Face</Name> |

### <Artist>

**Description:** The artist name

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Artist>Lady Gaga</Artist> |

### <Album>

**Description:** The name of the album

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Album>The Fame</Album> |

### 

### <Genere>

**Description:** The genre of the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Genre>Pop</Genre> |

### <Length>

**Description:** The length of the song

**Format:**

[hh:][mm:]ss.ss format.

hh = hours

mm = minutes

ss = seconds

dd = decimal part of seconds

**Example:**

|  |
| --- |
| <Length>03:36.28</Length> |

### <Size>

**Description:** The size of the file as a human readable string

**Format:** A UTF-8 string usually representing the size in megabytes

**Example:**

|  |
| --- |
| <Size>2.1MB</Size> |

### <ByteSize>

**Description:** The size of the file in bytes

**Format:** An integer

**Example:**

|  |
| --- |
| <ByteSize>20997010</ByteSize> |

### <Bitrate>

**Description:** The bit rate of the song

**Format:** A standard shorthand string representation for bit rate, such as 320kbps

**Example:**

|  |
| --- |
| <Bitrate>320kbps</Bitrate> |

### <Samplerate>

**Description:** The file sample rate

**Format:** A standard shorthand string representation of the sample rate, such as:

44.1k

48k

**Example:**

|  |
| --- |
| <Samplerate>44.1k</Samplerate> |

### <BPM>

**Description:** The BPM of the audio

**Format:** A decimal representation with 2 d.p.

**Example:**

|  |
| --- |
| <BPM>119.00</BPM> |

### <Comments>

**Description:** The comments for the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Comments>Sampled off vinyl</Comments> |

### <Key>

**Description:** The key of the song

**Format:** UTF-8 string. There are numerous standards an non-standards used, such as “C#m”, “1A”, “C”…etc

**Example:**

|  |
| --- |
| <Key>G#m</Key> |

### <Grouping>

**Description:** The group tag for the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Grouping>Club banger</Grouping> |

### <Remixer>

**Description:** The name of the remixer

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Remixer>DJ butcher</Remixer> |

### <Composer>

**Description:** The composer of the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Composer>Lady Gaga & Red One</Composer> |

### <Year>

**Description:** The year the song came out

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Year>2008</Year> |

### <Rating>

**Description:** The rating of the song

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Rating>1</Rating> |

### <GaindB>

**Description:** Serato’s user settable manual gain level. This value is used by Serato software to boost/cut the default track level. Note that this setting is overridden if Auto Gain is used (selected in the setup screen) and the track was analysed to determine the average audio level.

**Format:** A decimal (6 d.p.) representing the track gain to be applied on playback in decibels. As an example of the physical outcome, given a 6dB value, the amplitude of the resulting audio will increase by a factor of around 2, or more accurately

**Example:**

Decrease the playback level by 12dB

|  |
| --- |
| <GaindB>-12.00</GaindB> |

### <AutoGain>

**Description:** This value represents the audio level of the track as (automatically) calculated by Serato’s Auto Gain file analysis. This value is calculated when the track is analysed or placed on a deck for the first time. This value is not editable by the user. The user can change the <AutoGainAdj>. The value will be used to apply gain on the track based on the target value set in the setup screen at runtime.

**Format:** A decimal representation (6 d.p.) of the track’s level. However this is in non-standard units and should not be generated. You should let the software calculate it for you. It is safest to treat the value as opaque.

**Example:**

|  |
| --- |
| <AutoGain>2.03</AutoGain> |

### <AutoGainAdj>

**Description:** Represents the user adjustment (refinement) to the automated gain adjustment applied to a track at runtime. This is often used if auto gain calculation was incorrect or unsuitable.

**Format:** A decimal (6 d.p.) representing a further gain boost/cut in decibels to be applied to the level

**Example:**

|  |
| --- |
| <AutoGainAdj>-6.00</AutoGainAdj> |

### <Label>

**Description:** The name of the label that released the track

**Format:** UTF-8 string

**Example:**

|  |
| --- |
| <Label>Streamline Records</Label> |

### <Markers>

**Description:** Contains the saved cue-points

**Format:** This element should contain a collection of <Marker> elements.

Each <Marker> element specifies one cue point, with the following attributes:

* type – Specifies the type of the marker. There is only one valid value, which is “1”
* index – The index of the cue point (0..n). Note that there is a limit in Serato DJ of 8 cue points.

Each <Marker> element has the following sub-elements:

|  |  |  |
| --- | --- | --- |
| Name | Description | Format |
| Position | The position of the marker | An integer representing the number of milliseconds from start of the track |
| Color | The colour of the marker | An integer representing a 24-bit RGB value (don’t use hex, use integer) |
| LpPos | Deprecated – don’t use | Deprecated |
| Label | A human readable label for the marker | A UTF-8 string |

**Example:**

|  |
| --- |
| <Markers>  <Marker type=”1” index=”0”>  <Position>503</Position>  <Color>2599649</Color>  <LpPos>-1</LpPos>  <Label>Song Start</Label>  </Marker>  <Marker type=”1” index=”1”>  <Position>2573</Position>  <Color>13369344</Color>  <LpPos>-1</LpPos>  <Label>Cue1</Label>  </Marker>  </Markers> |

### <Loops>

**Description:** Contains the saved manual loops

**Format:** This element should contain a collection of <Loop> elements.

Each <Loop> element specifies one loop, with the following attributes:

* type – Specifies the type of the loop. There is currently only one valid value, which is “3”
* index – The index of the loop (0..n). Note that there is a limit in Serato DJ of 8 loops.

Each <Loop> element has the following sub-elements:

|  |  |  |
| --- | --- | --- |
| Name | Description | Format |
| Position | The start position of the loop | An integer representing the number of milliseconds from start of the track |
| Color | The colour of the loop | An integer representing a 24-bit RGB value (don’t use hex, use integer) |
| LpPos | The end position of the loop | An integer representing the number of milliseconds from start of the track |
| LpEnable | Indicates whether the loop is active | An integer with two valid values:  “0” – inactive  “1” – active |
| LpLocked | Indicates whether the loop is locked (for edits) | An integer with two valid values:  “0” – not locked  “1” – locked |

### <BeatGrid>

**Description:** Contains the saved beat grid for the track

**Format:** This element should contain a collection of <BeatGridMarker> elements.

Each <BeatGridMarker> element specifies one marker in the beat grid. The markers must be chronologically sorted

* Each <BeatGridMarker> element must have a <Position> element.
* Each <BeatGridMarker> elements **except the last one** must have a <BeatsTillNextMarker> element.
* The last <BeatGridMarker> element must have a <BPM> element.

|  |  |  |
| --- | --- | --- |
| Name | Description | Format |
| Position | The position of the marker | A decimal (6 d.p.) representing the number of seconds from start of the track. |
| BeatsTillNexMarker | The number of beats between this marker and the next one. | An integer representing the number of beats |
| BPM | The tempo after the last beat grid marker | A decimal (6 d.p.) representing the BPM |

**Example:**

|  |
| --- |
| <BeatGrid>  <BeatGridMarker>  <Position>0.503000</Position>  <BeatsTillNexMarker>32</Label>  </BeatGridMarker>  <BeatGridMarker>  <Position>225.503000</Position>  <BPM>100.00</Label>  </BeatGridMarker>  </BeatGrid> |