

STEM FILE



Specification



The information in this document is subject to change without notice and does not represent a commitment on the part of Native Instruments GmbH. The software described by this document is subject to a License Agreement and may not be copied to other media. No part of this publication may be copied, reproduced or otherwise transmitted or recorded, for any purpose, without prior written permission by Native Instruments GmbH, hereinafter referred to as Native Instruments.

“Native Instruments”, “NI” and associated logos are (registered) trademarks of Native Instruments GmbH.

Mac, Mac OS, GarageBand, Logic, iTunes and iPod are registered trademarks of Apple Inc., registered in the U.S. and other countries.

Windows, Windows Vista and DirectSound are registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners and use of them does not imply any affiliation with or endorsement by them.

Document authored by: Chad Carrier, Stewart Walker

Document version: 1.0 (12/2015)

Special thanks to the Beta Test Team, who were invaluable not just in tracking down bugs, but in making this a better product.

NATIVE INSTRUMENTS GmbH

Schlesische Str. 29-30
D-10997 Berlin
Germany
www.native-instruments.de

NATIVE INSTRUMENTS North America, Inc.

6725 Sunset Boulevard
5th Floor
Los Angeles, CA 90028
USA
www.native-instruments.com

NATIVE INSTRUMENTS K.K.

YO Building 3F
Jingumae 6-7-15, Shibuya-ku,
Tokyo 150-0001
Japan
www.native-instruments.co.jp

NATIVE INSTRUMENTS UK Limited

18 Phipp Street
London EC2A 4NU
UK
www.native-instruments.com



© NATIVE INSTRUMENTS GmbH, 2015. All rights reserved.

Table of Contents

1	Introduction	6
2	Stem File Design Goals	7
3	Stem File Specification	9
4	Stem Master Dynamics	11
4.1	Playback of Stems via Stem Master Dynamics	11
4.2	Creating Stem Content with Stem Master Dynamics	11
5	Authoring Guidelines	13
5.1	Stem Part Arrangement	13
5.2	Labeling	14
5.3	Colors	14
5.4	Example Layouts	14
5.4.1	Trance and Synth-Heavy Music	14
5.4.2	Percussive Techno Without Notable Song Structure or Melody	15
5.4.3	Vocal House/Disco/Pop	15
5.4.4	"Minimal"/Esoteric Music	15
5.5	Audio Quality	15
5.6	Mastering Stems with Hardware	16
6	"ni-stem" Stem Creator Tool	17
6.1	Installing on Mac	17
6.2	Getting Started on Mac	17
6.3	Installing on Windows	18
6.4	Getting Started on Windows	18
6.5	Main Options	19
6.6	Examples	19
6.7	Advanced Options and Defaults	20

7	Programmers' Reference	22
7.1	File Structure	22
7.2	Stem File Identification	24
7.2.1	Using mp4v2	24
7.2.2	Using mp4box	24
7.3	Metadata Structure	25
7.4	Verifying a Stem File	26
7.5	Stem File Playback	27
7.6	Code Example - Extracting Metadata from the iTunes Library on iOS	29

1 Introduction

Native Instruments is proud to introduce an open standard for stem content for DJs in the form of a single **Stem File**, and an associated toolset consisting of a **Stem Creator** tool and a **Stem Master Dynamics** DSP library. This document outlines the structure of the Stem File, how it is created, and various standards for the design. This is an open format which can be adopted freely in any music software, for playback or creation. The details of the Stem Creator tool and the Stem Master Dynamics are provided for proper authoring and playback of the Stem File.

2 Stem File Design Goals

The Stem File has been designed to satisfy a number of goals for DJ use. These are:

- (1) A single file for easy file management, distribution, and sale.
- (2) A file format free from any licensing fees.
- (3) Backward compatibility with 2-channel players.
- (4) Clear labeling of the stem components.
- (5) Compatible with mobile operating systems, including iOS.
- (6) The user has the ability to isolate key components of the music, specifically the drums, bassline, and vocals.
- (7) The audio quality of stem playback matches the original Stereo Master as closely as possible.

Stem Files are created using an **MP4 container** in order to satisfy goals (1), (2), (3), and (5). This single file is standardized; doesn't carry any license fees for creation, distribution, or use; accepts audio tracks of varying compressed audio qualities; is already supported by numerous 2-channel player applications and devices; and is also compatible with iOS (which has the most stringent/limited native support for file formats).

Stem Files contain audio encoded using **AAC** format in order to satisfy goals (2), (3), and (5). AAC codecs are available on both Mac OS X, iOS, and Windows platforms (beginning with Windows 7) and AAC audio files have no license fees for creation, distribution, or use. An option to use **Apple Lossless Codec** (ALAC) for audio encoding (available natively in OS X and Windows 10) is available for lossless audio compression at the cost of increased file size.

A simple **JSON format** is used to label the stems to satisfy goal (4). This is a human-readable format making it simple to implement.

A Stem File consists of **four Stem Part audio files** in order to satisfy goal (6), and also relates to goal (5). By limiting the Stem Part audio files to four, it focuses the Stem Parts to only essential instrument groups in the track (like "drums" and "bass", rather than wasting seven tracks just for drums, another two for bass, and so on). This makes the Stem Parts easy to use in a live DJ context, allows for consistent presentation from track to track, and also limits the file size (essential for mobile devices and computers using smaller SSD drives).

Instead of using a surround format to encode the Stem Parts, **individual audio tracks** are used for each Stem Part. This is done for a couple of reasons: First, by using discrete audio tracks for each Stem Part, there are no undesirable audio side effects such as correlation and cross-talk artifacts which can happen with surround encodings – with individual audio tracks, each Stem Part remains completely discrete. Secondly, the behavior of 2-channel players is more consistent when using discrete channels, unlike surround formats which have inconsistent playback behaviors in different systems. The **Stereo Master File** is marked as the active track so that it will be played by 2-channel players while the stem tracks are ignored to satisfy goal (3).



At the time of this writing, Windows Media Player appears to ignore the active/inactive flags of the audio tracks and tends to just play the last stem of the Stem File. You can, however, change the track played by Windows Media Player by clicking "Play > Audio and language tracks > Audio Track 5". This will then play the stereo master track from within the Stem File.

Finally, metadata is included in the Stem File that sets parameters of the **Stem Master Dynamics** processor which are to be applied in real-time during playback of the Stem Parts to help reproduce the original, refined sound of the content as required by goal (7). The Master Dynamics processor is available for free from www.stems-music.com in order to support proper playback of Stem Files in software and hardware.

3 Stem File Specification

- A Stem File is an MP4 container which contains five stereo audio files:
 - The first track is the Stereo Master of the song and is marked as "enabled" so that it is played by normal stereo MP4 players.
 - The remaining 4 tracks are the individual Stem Parts and are marked as "disabled" so that they are ignored by normal stereo MP4 players that aren't designed for Stem Part playback.
- The audio files can be encoded in two different ways:
 - **AAC 256kbps VBR**: this encoding is preferred due to its high quality and minimum file size.
 - **Apple Lossless Codec (ALAC)**: this encoding is for those users that want completely lossless audio, though the file size is significantly larger.
- All five tracks must be encoded using the same format (AAC or ALAC) and must have the same sample rate.
- The Stem Part audio Files must be created in such a way that playing them all simultaneously from their start will result in the correct playback timing of the Stem Parts. For example, a vocal part may not start until 1 minute into the track; the vocal Stem Part will therefore include 1 minute of silence at the start to ensure that the vocal starts at the correct location. VBR encoding used by AAC and ALAC optimizes compression for this silence to ensure the overall file size is kept as small as possible.
- The first track (the Stereo Master) must also be time-aligned with the Stem Parts so that they're all perfectly in sync when playing them all from their starting positions at the same time. This alignment allows things such as cue markers and beat grids to be shared between the Stem Parts and the Stereo Master.
- The individual Stem Part audio files are created in such a way that playing them at unity gain results in the correct audio mix of the files; in other words, the Stem Part audio files should not be level normalized after exporting from the DAW, nor should the levels of the Stem Parts be adjusted individually in any mastering process. When the user puts the volume faders for each Stem Part at maximum (unity gain), they should hear the original mix as intended by the content creator.

- Each Stem Part is given a name (e.g. "Drums", "Bass", "Synths", "Melody", "Atmosphere", "FX", "Vocals", "Hook", etc.) as well as an RGB color definition to be used when displaying Stem Parts on devices that support color.
- The settings of the Stem Master Dynamics are written to the metadata of the Stem File in order to be used during real-time playback of the Stem Parts. This Stem-specific metadata, including the Stem Part names described in the previous point, goes into a special "stem" box within the MP4 container in order to keep it separate from normal user metadata.

4 Stem Master Dynamics

When an artist creates a track, it typically goes through a final mastering stage where an engineer applies additional EQ, compression, and other techniques to make the music sound balanced, controlled, and often with maximum perceived volume. This mastering process is the final stage of the music creation, but Stem Files allow a user to change the audio mix of a track which happens *before* the mastering stage. Therefore, the normal process of mastering will not work for Stem Files and part of it needs to be replicated in real-time during playback. This is the purpose of the **Stem Master Dynamics** which is a high-quality stereo Compressor followed by a Limiter.

The Stem Master Dynamics is provided as a pre-compiled library which can be included in projects. It can be downloaded as part of the Developer Tools. This same Stem Master Dynamics processor must be used to ensure that the Stem Part playback sounds as the content creator intended – omitting the Stem Master Dynamics can result in the system providing sub-standard audio playback quality of Stem Parts.

4.1 Playback of Stems via Stem Master Dynamics

When a Stem File is created, metadata for the settings of the Stem Compressor and Limiter will be included so that the playback system can set the Stem Master Dynamics correctly when the content is loaded for playback. The Stem Master Dynamics therefore applies the final stage of "glue" and "control" to the mix, but still adapts to real-time changes in the mix created by the user. The full mix of stems as created by the user must be fed through the Stem Master Dynamics using the parameter values stored within the Stem File metadata.

4.2 Creating Stem Content with Stem Master Dynamics

In order to create Stem Files, the settings of the Stem Master Dynamics must be included in the Stem File metadata. If making a tool that creates Stem Files, that tool should allow the user to preview the sound of the stem mix through the Stem Master Dynamics and allow the user to adjust the parameters of the Stem Master Dynamics to taste. The tool then needs to write the settings of the Stem Master Dynamics into the JSON metadata for the file as descri-

bed earlier in this document. Failure to write Stem Master Dynamics parameters into the Stem File can result in no dynamics being applied during playback (resulting in sub-par audio quality) or can cause some systems to reject the Stem File entirely.

5 Authoring Guidelines

While the Stem File format specification is rather minimal and flexible, a number of authoring techniques and guidelines were employed by Native Instruments during the creation of the format in order to provide a consistent, simple, and quality user experience with the content. These guidelines are provided here in order to encourage consistency between all Stem Files such that users find familiar track layouts in their various Stem Files, thus making Stems easier to use.

5.1 Stem Part Arrangement

While every song is different, it is beneficial to have a consistent layout of the stem content for instant usability by DJs. As not every song can conform to one arrangement of stems, there are guidelines to isolate key components in the music:

- The first Stem Part is for Drums. If the track has lots of other parts, this will likely be all percussion parts in one track. If the song has a more minimal structure, this might be only the kick drum/low drums.
- The second Stem Part is for Bass, normally acoustic or synthesized bass.
- The third Stem Part is for the Main instrumentation of the track. This Stem Part will be some kind of strong hook or melody. If not, this would be all of the musical elements outside of drums, bass, and vocals.
- The fourth Stem Part is for Vocals. If the track does not have vocals, you can use this Stem Part for the remaining musical parts like the accompaniment, harmonies, additional percussion, effects, etc.

With this arrangement, you should always be able to isolate or remove the drums, the bass, and the vocals (if present). Their location within the file layout should be consistent from track to track.

5.2 Labeling

Each Stem Part should have a label that clearly describes its contents. It is recommended to use short, descriptive names for these labels, preferably a name taken directly from the following list whenever possible:

- Drums, Kick, HiHat, Snare, Toms, Clap, Tops, Loop, Bass, Bassline, SubBass, Reece, Synths, Pads, Chords, Drone, Comp, Guitar, Melody, Hook, Lead, Stabs, Hits, Acid, Donk, Noise, Atmos, Vocals, Voices, SFX, FX

5.3 Colors

Each Stem Part also has an RGB color definition to be used by software and hardware as an additional identifier for the content. In general, the color chosen is based on the subjective frequency content of the Stem. For example, low-frequency / bassy stems are shades of red while high-frequency / treble stems are shades of blue.

5.4 Example Layouts

Below are some examples of possible Stem Part Arrangements and Color Assignments based on various types of music and structures that could be encountered:

5.4.1 Trance and Synth-Heavy Music

Elements can be grouped cleanly into orchestration elements.

- Track 1 contains Drums (Colors: Reds)
- Track 2 contains Basses (Colors: Pink/Purple spectrum)
- Track 3 contains Harmonic Synths (Colors: Green spectrum)
- Track 4 contains Melodic Synths (Colors: Blue Spectrum)

5.4.2 Percussive Techno Without Notable Song Structure or Melody

Generally keep bass-heavy sounds to the left and bright sounds to the right.

- Track 1 contains Kicks & Toms (Colors: Reds)
- Track 2 contains Sub-bass (Colors: Pink/Purple spectrum)
- Track 3 contains Snares & HiHats (Colors: Green spectrum)
- Track 4 contains Ambiences and additional percussion (Colors: Blue Spectrum)

5.4.3 Vocal House/Disco/Pop

Harmonies and Melodies must be grouped to allow soloed a cappellas.

- Track 1 contains Drums (Colors: Reds)
- Track 2 contains Basses (Colors: Pink/Purple spectrum)
- Track 3 contains Melodies and Harmonies (Colors: Green spectrum)
- Track 4 contains Vocals (Colors: Orange Spectrum)

5.4.4 "Minimal"/Esoteric Music

Unique sounds should be discretely spaced across tracks to allow effective soloing.

- Track 1 contains Drums and percussive elements (Colors: Reds)
- Track 2 contains bassy elements (Colors: Pink/Purple spectrum)
- Track 3 contains sounds with sharp plucked attacks (Colors: Green spectrum)
- Track 4 contains sounds with soft attacks like pads and ambience (Colors: Blue Spectrum)

5.5 Audio Quality

When preparing stem content, the Stem Parts should be processed using the same mastering toolchain as the Stereo Master. This is easiest to achieve by preparing the four Stem Parts before the mastering session, and then having the Mastering Engineer master the final mix fol-

lowed by the four Stem Parts all within one session. This will ensure that the four Stem Parts have most of the same mastering tweaks as the Stereo Master, such as EQ changes, harmonic exciting, additional reverb, etc., and the Stem Master Dynamics would be used to replicate the final stage of compression and limiting applied to the Stereo Master.

5.6 Mastering Stems with Hardware

If you use external equipment to master your music, it is still possible to use this equipment when preparing Stem Part audio files. You will need to play each of the Stem Parts through the equipment, and then you'll need to re-import these processed Stem Parts back into a DAW along with the Stereo Master. You'll then need to edit the Stem Part audio files so that they're time-aligned with the Stereo Master. Sometimes the easiest way to do this is to align the Stem Parts to the Stereo Master and then re-export all from the DAW so the resulting audio files are all the same length. These files will then be ready to be loaded into Stem Creator.

6 “ni-stem” Stem Creator Tool

During development of the Stem File format, Native Instruments designed a simple command-line Stem Creator tool for creating the Stem File (.stem.mp4) from the individual Stem Part audio files and Stereo Master File using Python scripts and a few open-source libraries. You are free to use this tool to create Stem Files of your own, or to just gain a deeper understanding of the creation method and structure of the file.

You will receive the tool `ni-stem` as an archive file.

OS	Archive
Mac OS X	<code>ni-stem-1.0.tar.gz</code>
Windows	<code>ni-stem.Win.1.0.zip</code>

Installation is done by unpacking the archive.

6.1 Installing on Mac

1. Choose a directory where you want to install `ni-stem` . e.g.
`/Users/joe.coder/myStems`
2. Copy the archive file `ni-stem-1.0.tar.gz` to the installation folder
3. Unpack the archive. This can be done using by right-clicking the archive in the OS X Finder and then choosing the standard option for opening archives.
4. The unpacker installs `ni-stem` into the folder `ni-stem-1.0`

6.2 Getting Started on Mac

1. Open a Terminal window.
2. Change directory to `ni-stem-1.0` inside your installation folder, e.g.
`$ cd myStems/ni-stem-1.0`
3. You may check your position in the file system with `pwd` (print working directory)
`$ pwd /Users/joe.coder/myStems/ni-stem-1.0`

4. Call `$./ni-stem -h` to check if `ni-stem` will be executed.
5. Copy source files (WAV or AIFF) to folder
`/Users/joe.coder/myStems/ni-stem-1.0`
6. You should be able now to create stem files.

6.3 Installing on Windows

1. Choose a directory where you want to install `ni-stem`. e.g.
`C:\Users\joe.coder\myStems.`
2. Copy the archive file `ni-stem-1.0.Win.zip` to the installation folder.
3. Unpack the archive. This can be done using by right-clicking the archive in the Windows Explorer and then choosing the standard option for opening archives: "extract all".
4. The unpacker installs `ni-stem` into the folder `ni-stem-Win.1.0`.

6.4 Getting Started on Windows

1. Open a Command Prompt window.
2. Change directory to `ni-stem` inside your installation folder, e.g.
`> cd myStems\ni-stem.Win.1.0.`
3. You may check your position in the file system (shown in the console prompt)
`C:\Users\joe.coder\myStems\ni-stem.Win.1.0>.`
4. Call `ni-stem.exe -h` to check if `ni-stem` will be executed.
5. Copy source files (WAV or AIFF) to folder
`C:\Users\joe.coder\myStems\ni-stem.Win.1.0.`
6. You should be able now to create Stem Files.

6.5 Main Options

Command	Description
ni-stem -h ni-stem --help	Get help
ni-stem create	Create a Stem File
ni-stem info	Dump stem metadata to file

6.6 Examples

The main calls are explained best via example. The following example assumes that you are calling `ni-stem` from the directory where your four Stem Files and Stereo Master File are located. Note that backslash characters “\” must be omitted if `ni-stem` is used on the command line. Backslashes “\” are required when calling `ni-stem` from within a shell script:

```
./ni-stem create -s AfroDrums.mp4 \  
                  WoodBass.mp4 \  
                  Marimbaphone.mp4 \  
                  Vocals.mp4 \  
-x MyAfroMaster.mp4 \  
-m MyAfroNamesAndColors.json \  
-o MyAfro
```

Main call example 1

You may also read files from different directories using paths. There are also long forms for the command line options:

```
./ni-stem create --stems      ./MyAfroProject/AfroDrums.mp4 \
                             ./MyAfroProject/WoodBass.mp4 \
                             ./MyMarimbaStems/Marimbaphone.mp4 \
                             ./MyVocals/Vocals.mp4 \
--mixdown                    ./MyMixes/MyAfroMaster.mp4 \
--metadata                   ./MyStemMetaData/MyAfroNamesAndColors.json
--output                     ./MyStemFiles/Afro
```

Main call example 2

Calling `ni-stem info` allows you to check if Stem Part audio files and metadata have been successfully written into the Stem File.

- `./ni-stem info -s MyAfro.stem.mp4` gives a summary about the Stem Part audio files in your Stem File, in this case `MyAfro.stem.mp4`.
- `./ni-stem info -s MyAfro.stem.mp4 -m MyAfro.json` writes a JSON file containing the metadata as JSON expression.

6.7 Advanced Options and Defaults

The following table contains all options of command `ni-stem`. A short and long version of options, e.g. `-h --help`, is available to be chosen from.

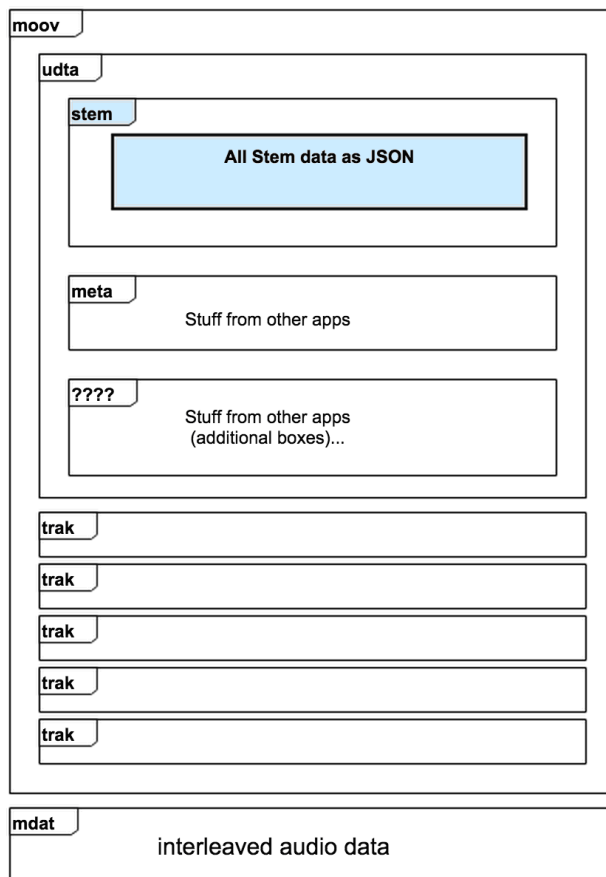
Command	Opt	Option	Default	Required	Description
	-h	--help			List of top level options
create	-h	--help			Help on the create option
info	-h	--help			Help on the info option
create	-s	--stems		Yes	4 paths for four Stem-Part audio files as WAV or AIFF. WAV and AIFF files will be re-encoded automatically by the tool.

Command	Opt	Option	Default	Required	Description
	-x	--mixdown		Yes	1 path for the Stereo Master File as WAV or AIFF. WAV and AIFF files will be re-encoded automatically by the tool.
	-m	--metadata	Yes		Path to metadata JSON file. If omitted, default names and colors are used.
	-o	--output	Yes		Path to the Stem Sile to be created. If omitted, the name of the base name of the retail master file is suffixed by “.stem.mp4”
info	-s	--stem		Yes	Stem File to be read from
	-m	--metadata			File path to a JSON file to be written to
	-r	--report			File path to a text file to be written to

7 Programmers' Reference

7.1 File Structure

The structure of the Stem File follows the specification for MP4 containers given by ISO/IEC 14496-12 and ISO/IEC 14496-14. The Stem File is formed as a hierarchical structure of objects called **boxes** (sometimes called **atoms**). The box type is given using a 4-byte character code. The Stem File format mostly relies upon already existing box types and introduces only one new box called `stem`, which is marked blue in the following diagram:



File structure diagram

The outermost box is the file type box `ftpy` followed by the movie metadata box `moov`. The stem metadata is stored in a `stem` box inside the user data box `udta`. While it would make sense to place the box outside of the user data, placing it in `udta` makes it more accessible through existing APIs such as the AVFoundation on iOS. This compatibility to existing APIs is vital, as it allows Stem Files to be read and played which are stored in the iTunes Library on an iOS device, to which developers have only limited access through said APIs.

7.2 Stem File Identification

There are two criteria every Stem File **must** fulfill—failure to fulfill these criteria means the file in question is **not** a Stem File:

- The file **must** be a valid MP4 file (ISO base media file format: ISO/IEC 14496-12) with the “.mp4” file extension. Preferably the file name should end with “.stem.mp4”, but just “.mp4” is also valid.
- The MP4 file **must** contain a box (i.e. data container) of type `stem` as a sub-box of `moov.udta`. `moov` is the root box of the MP4 file, as shown in the Stem File Structure diagram in section [↑7.1, File Structure](#).

Several command line tools exist that allow you to inspect the container structure of an MP4 file:

7.2.1 Using mp4v2

The `mp4v2` package available from <https://code.google.com/p/mp4v2/> contains the “mp4file” tool.

Calling

```
$ mp4file --dump MyStemFile.stem.mp4
```

will list the box structure of the MP4 file and, for a valid Stem file, **must** contain the line:

```
"MyStemFile.stem.mp4": type stem (moov.udta.stem)
```

7.2.2 Using mp4box

This command line tool is available from <https://gpac.wp.mines-telecom.fr/mp4box/>

Calling

```
$ mp4box --dump-udta 0:stem MyStemFile.stem.mp4
```

will create a “MyStemFile.stem_stem.udta” file with the content of the `stem` box. The tool **must not** give the error `No UDTA for type stem found`, as that would imply a missing `moov.udta.stem` box.

7.3 Metadata Structure

The `stem` box in `udta` contains the stem data for all tracks in JSON format, which is outlined below:

```
{
  "version" : 1,
  "mastering_dsp" :
  {
    "compressor" :
    {
      "enabled"      : true,
      "input_gain"   : 6,
      "output_gain"  : 4.4,
      "threshold"    : -2.4,
      "dry_wet"       : 100,
      "attack"        : 0.001,
      "release"       : 0.2,
      "ratio"         : 1.5,
      "hp_cutoff"    : 50
    },
    "limiter" :
    {
      "enabled"      : true,
      "threshold"    : 0.0,
      "ceiling"       : -0.35,
      "release"       : 0.05
    }
  },
  "stems" :
  [
    { "name" : "Afro Drums",    "color" : "#fd4a4a" },
    { "name" : "Wood Bass",    "color" : "#ffff00" },
    { "name" : "Marimbaphone",  "color" : "#00e8e8" },
    { "name" : "Vocals",       "color" : "#ad65ff" }
  ]
}
```

Stem metadata structure in JSON format

Read-only metadata for the Stereo Master File and the four Stem Part audio files (i.e. ID, sample rate, and other technical parameters) are contained in 5 standard `trak` boxes as defined by the MP4 standard. `trak` box 1 refers to the Stereo Master File. `trak` boxes 2 through 5 refer to Stem Part audio files.

7.4 Verifying a Stem File

To fully verify a Stem File (i.e. make sure that it can be played back properly), it is necessary to inspect its content.

- The MP4 file **must** contain exactly five audio tracks.
- The audio tracks **must** have the same kind of audio data (matching sample rate, codec, ...).
- The compressor and limiter values stored in the JSON-formatted data of the `moov.udta.stem` box **must** lie in the valid ranges as shown in the table below:

Compressor parameter values must be within these ranges:

Parameter	Description	Range
<code>enabled</code>	Activate/deactivate the compressor	true, false
<code>input_gain</code>	Gain of signal going into the compressor	[-12, 12] dB
<code>output_gain</code>	Gain of output from the compressor going into the limiter (after the dry/wet mix)	[-12, 12] dB
<code>threshold</code>	Compressor threshold	[-40, 0] dB
<code>dry_wet</code>	Mix of dry signal vs. compressed signal	[0, 100] %
<code>attack</code>	Compressor attack time	[0.0001, 0.03] sec
<code>release</code>	Compressor release time	[0.1, 1.6] sec
<code>ratio</code>	Compressor ratio (stepped value)	[1.5, 2, 3, 4, 5, 10] :1
<code>hp_cutoff</code>	Cutoff frequency of High Pass Filter in Sidechain input to compressor	[20, 500] Hz – a value less than 20 will bypass the sidechain.

Limiter parameter values must be within these ranges:

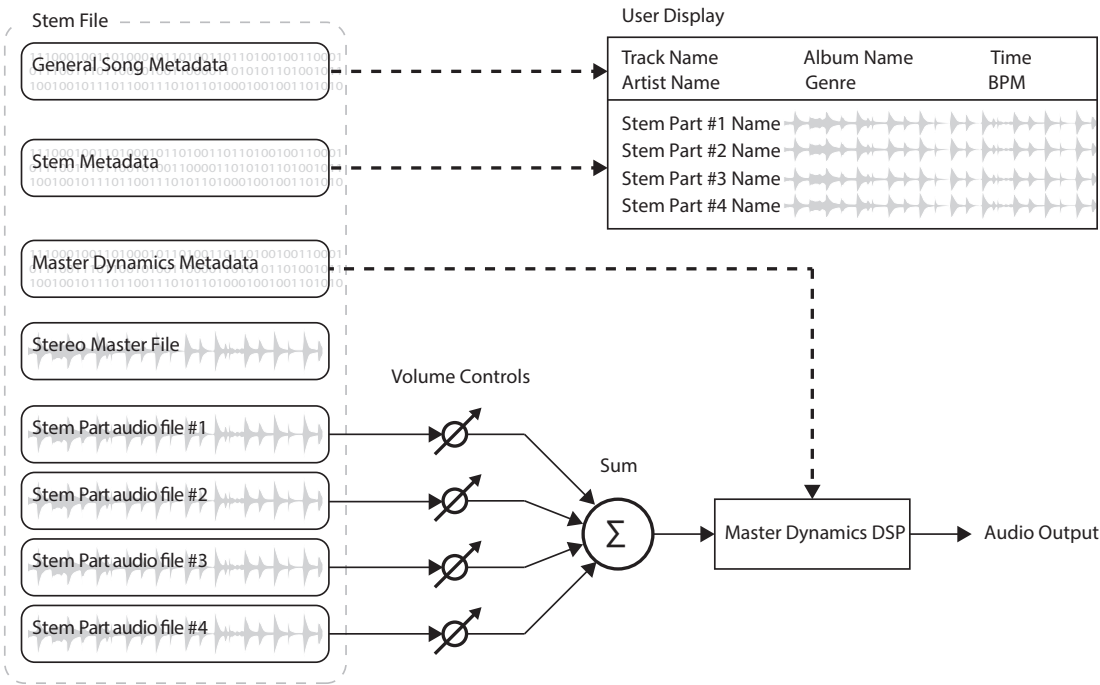
Parameter	Description	Range
enabled	Activate/deactivate the limiter	true, false
threshold	Limiter threshold	[-40, 0] dBFS
ceiling	Output ceiling level	[-40, 0] dBFS
release	Limiter release time	[0.001, 1.0] sec

7.5 Stem File Playback

In order to play back Stems Files, the software or hardware must perform the following minimum feature set:

- Access the four Stem Part audio files within the Stem File.
- Access the JSON Stem metadata within the Stem File.
- Play a mix of the Stem Parts created by the user where no Stem Part is amplified – only attenuation should be available for each Stem Part audio File.
- The mix of the Stem Parts is fed into the Stem Master Dynamics whose parameters are set according to the values within the JSON Stem metadata.
- The output of the Stem Master Dynamics is what is heard by the user during Stem File playback – this signal can then be routed elsewhere in the system for mixing with other sources, for example.

Below is a diagram illustrating how the playback system should work:



Stem File playback schematic

7.6 Code Example - Extracting Metadata from the iTunes Library on iOS

The following code example shows how to extract Stem Metadata from Stem Files that are in the iTunes Library on an iOS device using Objective-C:

```
MPMediaQuery *everything = [[MPMediaQuery alloc] init];
NSArray      *itemsFromGenericQuery = [everything items];
for (MPMediaItem *song in itemsFromGenericQuery) {
    NSURL *songUrl = [song valueForKeyProperty:MPMediaItemPropertyAssetURL];
    AVAsset *asset = [AVAsset assetWithURL:songUrl];

    for (AVMetadataItem *metaItem in asset.metadata){
        NSNumber *key = (NSNumber*)metaItem.key; // on iOS >= 8 you could also use
        metaItem.identifier, which should be @"udta/stem"
        if(key.intValue == 'stem'){
            NSData *rawJsonData = (NSData*)metaItem.value;
            NSString *stemJson = [[NSString alloc] initWithData:rawJsonData
            encoding:NSUTF8StringEncoding] autorelease];

            NSLog(@"Stem JSON:%@", stemJson);
        }
    }
}
```

Code example for extracting metadata from iTunes library on iOS