



链滴

Translate Text in Sphinx Templates and Configurations

作者: [blueset](#)

原文链接: <https://ld246.com/article/1583927736279>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

In both my [previous](https://ld246.com/forward?goto=https%3A%2F%2Fblog.1a23.com%2F2019%2F10%2F06%2Fcustom-sort-order-in-music-libraries-macos-and-android-en-zh-han-ja%2F) and [recent](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Fblueset%2Fproject-lyricova) projects, I have been working with tags (metadata) of music files. One of the reason being I am rather particular about having a nicely organised library with all tag data aligned to the same format. Until recently while I was seeking for a solution to read and write tags of (potentially) all music formats, I only have MP3, FLAC, AIFF and M4A in my library, so that's kinda *all* for me., and I encountered FFmpeg, the Swiss Army Knife of media processing.

FFmpeg has always been my go-to solution for processing media programmatically or in batch, and I have recently found the way to write into the tags of music files using it. The way of doing so might be a little verbose as everything have to fit into the command line interface with other components.

Read tags

To read tags from a music file, we actually need to use `ffprobe` instead of `ffmpeg`. `ffprobe` in this case can produce an output in JSON which is more program-friendly.

```
ffprobe -show_format -print_format json music.mp3
```

```
{
  "filename": "music.mp3",
  "nb_streams": 1,
  "nb_programs": 0,
  "format_name": "mp3",
  "format_long_name": "MP2/3 (MPEG audio layer 2/3)",
  "start_time": 0.000000,
  "duration": 90.160000,
  "size": "464333",
  "bit_rate": "18015",
  "probe_score": 51,
  "tags": {
    "title": "初音ミクの消失 (2018 Remake)",
    "artist": "cosMo@暴走P feat. 初音ミク",
    "album": "初音ミクの消失 -Real and Repeat-",
    "compilation": "1",
    "encoded_by": "Max 0.9.1",
    "title-sort":
```

```

"はつねみくのしょうしつ (2018 Remake)",
</span></span><span class="highlight-line"><span class="highlight-cl">      "artist-sort
: "cosMo@ぼうそうP feat. はつねみく",
</span></span><span class="highlight-line"><span class="highlight-cl">      "album-sor
": "はつねみくのしょうしつ -Real and Repeat-",
</span></span><span class="highlight-line"><span class="highlight-cl">      "TDTG": "2
14-11-03T15:38:58",
</span></span><span class="highlight-line"><span class="highlight-cl">      "encoder":
"Lavf58.29.100"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>

```

The option `-show_format` adds the section `"format"` to the output, which has the metadata of the file format as long as the tags. Omitting this option will result in an output with no useful data.

Besides JSON, `ffprobe` also produce output in CSV, flat key-value pairs, NI, and XML syntax. You can choose whichever format that fits your need better. See <https://ld246.com/forward?goto=https%3A%2F%2Fffmpeg.org%2Fffprobe.html> for more options.

Note that both `ffmpeg` and `ffprobe` prints version number, compile info of the program itself and metadata of the file to `stderr`. Make sure to get rid of it if your program consumes both `stdout` and `stderr` together by default.

Write tags

It is trickier to write tags than just reading them. In the logic of FFmpeg, everything is considered as a *stream*, some file comes as an input stream, some filters are applied, and then one output is produced. This is the same case for writing tags too. It could make the entire procedure a little more complex, since usually when you just want to make changes to tag data, you want to just overwrite the original file. But with the way FFmpeg is designed, [it won't allow you to do so](https://ld246.com/forward?goto=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F28877049%2Fissue-with-overwriting-file-while-using-ffmpeg-for-converting). The best you can go around with this is to let FFmpeg to make a copy of the file, and delete the old one afterwards.

Options used in this command are as follows:

-

-

- `-i aiff.aiff`: path to the of input file.

-

-

- `-map 0`: map both audio and video of the 0th

- input file to the output file, i.e. to keep the media content unchanged.

-

-

- `-y`: overwrite if the destination file exists. *Note that you **annot** write to your input file even with this option enabled, or you may corrupt the file entirely.*

-

-

- `-codec copy`: to keep the codec of the file unchanged, so as to prevent unnecessary re-encoding.

-

<p><code>-write_id3v2 1</code>: quite self-explanatory, only use this option if you want to write the tags as ID3v2. In cases like AIFF, FFmpeg cannot detect the correct tag type to use, so forcing ID3v2 could be sometimes useful.</p>

<p><code>-metadata "tag-name=tag value"</code>: this is where you write/overwrite tags</p>

<p><code>aiffout.aiff</code>: path to the output file.</p>


```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">ffmpeg -i aiff.aiff -map 0 -y -codec copy -write_id3v2 1 -metadata "artist-sort=emon feat ort" aiffout.aiff</span></span><span class="highlight-line"><span class="highlight-cl">ffmpeg version 4.2.1 Copyright (c) 2000-2019 the FFmpeg developers</span></span><span class="highlight-line"><span class="highlight-cl"> built with Apple clang version 11.0.0 (clang-1100.0.33.8)</span></span><span class="highlight-line"><span class="highlight-cl"> configuration: --prefix=/usr/local/Cellar/ffmpeg/4.2.1_2 --enable-shared --enable-pthreads --enable-version3 --enable-avresample --cc=clang --host-cflags='-I/Library/Java/JavaVirtualMachines/adoptopenjdk-13.jdk/Contents/Home/include -I/Library/Java/JavaVirtualMachines/adoptopenjdk-13.jdk/Contents/Home/include/darwin -fno-stack-check' --host-ldflags= --enable-ffplay --enable-gnutls --enable-gpl --enable-libaom --enable-libbluray --enable-libmp3lame --enable-libopus --enable-librubberband --enable-libsndio --enable-libtheora --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --enable-libxvid --enable-lzma --enable-libfontconfig --enable-libfreetype --enable-frei0r --enable-libass --enable-libopencl --enable-libopenm2 --enable-libopenjpeg --enable-librtmp --enable-libspeex --enable-libsoxr --enable-videlabs --disable-libjack --disable-indev=jack</span></span><span class="highlight-line"><span class="highlight-cl"> libavutil 56.31.100 / 56.31.100</span></span><span class="highlight-line"><span class="highlight-cl"> libavcodec 58.54.100 / 58.54.100</span></span><span class="highlight-line"><span class="highlight-cl"> libavformat 58.29.100 / 58.29.100</span></span><span class="highlight-line"><span class="highlight-cl"> libavdevice 58.8.100 / 58.8.100</span></span><span class="highlight-line"><span class="highlight-cl"> libavfilter 7.57.100 / 7.57.100</span></span><span class="highlight-line"><span class="highlight-cl"> libavresample 4.0.0 / 4.0.0</span></span><span class="highlight-line"><span class="highlight-cl"> libswscale 5.100 / 5.100</span></span><span class="highlight-line"><span class="highlight-cl"> libswresample 3.5.100 / 3.5.100</span></span><span class="highlight-line"><span class="highlight-cl"> libpostproc 55.5.100 / 55.5.100</span></span><span class="highlight-line"><span class="highlight-cl">Gussed Channel aayout for Input Stream #0.0 : stereo</span></span><span class="highlight-line"><span class="highlight-cl">Input #0, aiff, from 'aiff.aiff':</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:</span></span></code></pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> title      : sha
e it!
</span></span><span class="highlight-line"><span class="highlight-cl"> artist      : em
n feat. 初音ミク.鏡音リン.鏡音レン
</span></span><span class="highlight-line"><span class="highlight-cl"> album       :
マジカルミライ 2014] OFFICIAL ALBUM
</span></span><span class="highlight-line"><span class="highlight-cl"> compilation :
1
</span></span><span class="highlight-line"><span class="highlight-cl"> encoded_by  :
Max 0.9.1
</span></span><span class="highlight-line"><span class="highlight-cl"> title-sort  : sh
ke it!
</span></span><span class="highlight-line"><span class="highlight-cl"> creation_time :
2014-11-03T15:38:58
</span></span><span class="highlight-line"><span class="highlight-cl"> TDTG       : 2
14-11-03T15:38:58
</span></span><span class="highlight-line"><span class="highlight-cl"> album-sort  :
「まじかるみらい 2014」 OFFICIAL ALBUM
</span></span><span class="highlight-line"><span class="highlight-cl"> artist-sort  :
mon feat. はつねみく.かがみねりん.かがみねれん
</span></span><span class="highlight-line"><span class="highlight-cl"> Duration: 00:03:
7.03, start: 0.000000, bitrate: 2822 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0: A
dio: pcm_s32be, 44100 Hz, stereo, s32, 2822 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Output #0, aiff, to
aiffout.aiff':
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> title      : sha
e it!
</span></span><span class="highlight-line"><span class="highlight-cl"> artist      : em
n feat. 初音ミク.鏡音リン.鏡音レン
</span></span><span class="highlight-line"><span class="highlight-cl"> album       :
マジカルミライ 2014] OFFICIAL ALBUM
</span></span><span class="highlight-line"><span class="highlight-cl"> compilation :
1
</span></span><span class="highlight-line"><span class="highlight-cl"> encoded_by  :
Max 0.9.1
</span></span><span class="highlight-line"><span class="highlight-cl"> title-sort  : sh
ke it!
</span></span><span class="highlight-line"><span class="highlight-cl"> album-sort  :
「まじかるみらい 2014」 OFFICIAL ALBUM
</span></span><span class="highlight-line"><span class="highlight-cl"> TDTG       : 2
14-11-03T15:38:58
</span></span><span class="highlight-line"><span class="highlight-cl"> artist-sort  :
mon feat sort
</span></span><span class="highlight-line"><span class="highlight-cl"> encoder     :
avf58.29.100
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0: A
dio: pcm_s32be (NONE / 0x454E4F4E), 44100 Hz, stereo, s32, 2822 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream mapping:
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0 -&gt;
#0:0 (copy)
</span></span><span class="highlight-line"><span class="highlight-cl"> Press [q] to stop, [
] for help

```

```
</span></span><span class="highlight-line"><span class="highlight-cl">size= 78219kB t
me=00:03:47.02 bitrate=2822.5kb/s speed=2.08e+03x
</span></span><span class="highlight-line"><span class="highlight-cl">video:0kB audio:7
218kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000549%
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
```

In order to provide a uniform interface for writing metadata, FFmpeg has some custom aliases for common tag names that are different from what is actually written to the file. Examples like `<code>title</code>`, `<code>album</code>`, `<code>artist</code>`, and `<code>genre</code>` should be honored in most tag types. But some aliases might not always be mapped to the tag you would expect. Sort key tags of Vorbis Comments in FLAC is not mapped as they are in ID3, and are considered as custom tags by FFmpeg. It is always better to check with `<code>ffprobe</code>` using an already properly tagged file to see if aliases are used. Multimedia Wiki has [provided a list](https://ld246.com/forward?goto=https%3A%2F%2Fwiki.multimedia.cx%2Findex.php%2FFFmpeg_Metadata) of common aliases and the tags they mapped to in actual files.

To write a custom tag, just use the key name of your choice directly. FFmpeg can figure out the proper way to add them to your file, like using `<code>TXXX</code>` in ID3.

As mentioned previously, this command also produces a lot of debug info to `<code>stderr</code>`. In fact, all these output are by default printed to `<code>stderr</code>`, so you will only get a return code 0, and nothing from `<code>stdout</code>`.

Read cover art

In FFmpeg, a music file with a cover art embedded is considered as a 2 input streams — 1 audio stream and 1 single-frame video stream (as a picture). So, to extract the cover art out, what we need to do is similar to stripping off the audio track of a video.

Options used in this command are as follows:

-

- `<code>-i mp3.mp3</code>`: path to the of input file.

- `<code>-an</code>`: drop the audio stream.

- `<code>-vcodec copy</code>`: to keep the codec of the video stream. (That should mean the image format I guess)

- `<code>cover.png</code>`: path to the output file.

-

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">ffmpeg -i mp3.mp3 -an -vcodec copy cover.png
</span></span><span class="highlight-line"><span class="highlight-cl">ffmpeg version 4.2
1 Copyright (c) 2000-2019 the FFmpeg developers
</span></span><span class="highlight-line"><span class="highlight-cl"> built with Apple clang
version 11.0.0 (clang-1100.0.33.8)
</span></span><span class="highlight-line"><span class="highlight-cl"> configuration: --
refix=/usr/local/Cellar/ffmpeg/4.2.1_2 --enable-shared --enable-pthreads --enable-version3 -
enable-avresample --cc=clang --host-cflags='-I/Library/Java/JavaVirtualMachines/adoptopen
dk-13.jdk/Contents/Home/include -I/Library/Java/JavaVirtualMachines/adoptopenjdk-13.jdk/
ontents/Home/include/darwin -fno-stack-check' --host-ldflags= --enable-ffplay --enable-gnu
ls --enable-gpl --enable-libaom --enable-libbluray --enable-libmp3lame --enable-libopus --e
nable-librubberband --enable-libsnpappy --enable-libtesseract --enable-libtheora --enable-libv
dstab --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --enable-libxvid --
nable-lzma --enable-libfontconfig --enable-libfreetype --enable-frei0r --enable-libass --enabl
-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-librtmp --e
nable-libspeex --enable-libsoxr --enable-videotoolbox --disable-libjack --disable-indev=jack
</span></span><span class="highlight-line"><span class="highlight-cl"> libavutil 56. 31
100 / 56. 31.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavcodec 58.
54.100 / 58. 54.100
</span></span>
```



```

</span></span><span class="highlight-line"><span class="highlight-cl"> libavformat 58.
29.100 / 58. 29.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavdevice 58.
8.100 / 58. 8.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavfilter 7. 57
100 / 7. 57.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavresample 4.
0. 0 / 4. 0. 0
</span></span><span class="highlight-line"><span class="highlight-cl"> libswscale 5.
.100 / 5. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libswresample 3
5.100 / 3. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libpostproc 55.
5.100 / 55. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl">Input #0, mp3, fr
m 'mp3.mp3':
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> title : Mel
dy Line
</span></span><span class="highlight-line"><span class="highlight-cl"> artist : Smi
eR feat. 初音ミク
</span></span><span class="highlight-line"><span class="highlight-cl"> track : 02
20
</span></span><span class="highlight-line"><span class="highlight-cl"> album :
elody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> genre : &l
;unknown&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> title-sort : M
elody Line
</span></span><span class="highlight-line"><span class="highlight-cl"> album-sort :
Melody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> artist-sort :
mileR feat. はつねみく
</span></span><span class="highlight-line"><span class="highlight-cl"> Duration: 00:03:
1.09, start: 0.025056, bitrate: 457 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0: A
udio: mp3, 44100 Hz, stereo, fltp, 320 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:1: Vi
deo: png, rgba(pc), 1500x1499, 90k tbr, 90k tbn, 90k tbc (attached pic)
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> comment
Other
</span></span><span class="highlight-line"><span class="highlight-cl">Output #0, image2
to 'cover.png':
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> title : Mel
dy Line
</span></span><span class="highlight-line"><span class="highlight-cl"> artist : Smi
eR feat. 初音ミク
</span></span><span class="highlight-line"><span class="highlight-cl"> track : 02
20
</span></span><span class="highlight-line"><span class="highlight-cl"> album :
elody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> genre : &l

```

```

;unknown>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> title-sort : M
lody Line
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> album-sort :
Melody Line(s)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> artist-sort :
mileR feat. はつねみく
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> encoder :
avf58.29.100
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Stream #0:0: Vi
eo: png, rgba(pc), 1500x1499, q=2-31, 90k tbr, 90k tbn, 90k tbc (attached pic)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Metadata:
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> comment
Other
</span> </span> <span class="highlight-line"> <span class="highlight-cl">Stream mapping:
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Stream #0:1 -&gt;
#0:0 (copy)
</span> </span> <span class="highlight-line"> <span class="highlight-cl">Press [q] to stop, [
] for help
</span> </span> <span class="highlight-line"> <span class="highlight-cl">frame= 1 fps=0.
q=-1.0 Lsize=N/A time=00:00:00.00 bitrate=N/A speed=0.00122x
</span> </span> <span class="highlight-line"> <span class="highlight-cl">video:3371kB aud
o:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
</span> </span> </code> </pre>

```

So, nothing fancy here. It seems like the format the output image would always follow the extension in the output path specified no matter what you have in the file, even when `-vcodec copy` is enabled. I'm not really sure if that option is really needed or not. But what is important here is to **have a proper extension** in the output path.

Write cover art

Similar to reading, writing cover art is the reverse process — combining an audio and a static picture into one file. The static picture will be automatically considered as a cover art.

Options used in this command are as follows:

-

-

- `-i mp3.mp3`: path to the of input audio file.

-

-

- `-i cover.png`: path to the cover art.

-

-

- `-map 0`: map streams of the 0

- `th`

- (first) input to the output.

-

-

- `-map 1:0`: map first stream (image data) of the 1

- `th`

- `mfn1`

- `th`

- is the way of saying the #1 element in a zero-indexed list, it's intended here. (second) input (i.e. the picture) to the output.

-


```

<li>
<p><code>-codec copy</code>: to keep the codec of streams, and prevent unnecessary encoding.</p>
</li>
<li>
<p><code>mp3out.mp3</code>: path to the output file.</p>
</li>
</ul>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">ffmpeg -i mp3.mp3 -i cover.png -map 0 -map 1:0 -codec copy mp3out.mp3
</span></span><span class="highlight-line"><span class="highlight-cl">ffmpeg version 4.2
1 Copyright (c) 2000-2019 the FFmpeg developers
</span></span><span class="highlight-line"><span class="highlight-cl"> built with Apple clang version 11.0.0 (clang-1100.0.33.8)
</span></span><span class="highlight-line"><span class="highlight-cl"> configuration: --refix=/usr/local/Cellar/ffmpeg/4.2.1_2 --enable-shared --enable-pthreads --enable-version3 -
enable-avresample --cc=clang --host-cflags='-I/Library/Java/JavaVirtualMachines/adoptopenjdk-13.jdk/Contents/Home/include -I/Library/Java/JavaVirtualMachines/adoptopenjdk-13.jdk/
ontents/Home/include/darwin -fno-stack-check' --host-ldflags= --enable-ffplay --enable-gnul
s --enable-gpl --enable-libaom --enable-libbluray --enable-libmp3lame --enable-libopus --e
able-librubberband --enable-libsndio --enable-libtheora --enable-libv
dstab --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --enable-libxvid --
nable-lzma --enable-libfontconfig --enable-libfreetype --enable-frei0r --enable-libass --enabl
-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-librtmp --e
able-libspeex --enable-libsoxr --enable-videotoolbox --disable-libjack --disable-indev=jack
</span></span><span class="highlight-line"><span class="highlight-cl"> libavutil 56. 31
100 / 56. 31.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavcodec 58.
54.100 / 58. 54.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavformat 58.
29.100 / 58. 29.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavdevice 58.
8.100 / 58. 8.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavfilter 7. 57
100 / 7. 57.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libavresample 4.
0. 0 / 4. 0. 0
</span></span><span class="highlight-line"><span class="highlight-cl"> libswscale 5.
.100 / 5. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libswresample 3
5.100 / 3. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl"> libpostproc 55.
5.100 / 55. 5.100
</span></span><span class="highlight-line"><span class="highlight-cl">Input #0, mp3, fr
m 'mp3.mp3':
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> title : Mel
dy Line
</span></span><span class="highlight-line"><span class="highlight-cl"> artist : Smi
eR feat. 初音ミク
</span></span><span class="highlight-line"><span class="highlight-cl"> track : 02
20
</span></span><span class="highlight-line"><span class="highlight-cl"> album :
elody Line(s)

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> genre      : &l
;unknown&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> title-sort  : M
lody Line
</span></span><span class="highlight-line"><span class="highlight-cl"> album-sort   :
Melody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> artist-sort   :
mileR feat. はつねみく
</span></span><span class="highlight-line"><span class="highlight-cl"> Duration: 00:03:
1.09, start: 0.025056, bitrate: 457 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0: A
dio: mp3, 44100 Hz, stereo, fltp, 320 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:1: Vi
eo: png, rgba(pc), 1500x1499, 90k tbr, 90k tbn, 90k tbc (attached pic)
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> comment
Other
</span></span><span class="highlight-line"><span class="highlight-cl">Input #1, png_pipe
from 'cover.png':
</span></span><span class="highlight-line"><span class="highlight-cl"> Duration: N/A, bi
rate: N/A
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #1:0: Vi
eo: png, rgba(pc), 1500x1499, 25 tbr, 25 tbn, 25 tbc
</span></span><span class="highlight-line"><span class="highlight-cl">Output #0, mp3, t
'mp3out.mp3':
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> TIT2      : Me
lody Line
</span></span><span class="highlight-line"><span class="highlight-cl"> TPE1      : S
ileR feat. 初音ミク
</span></span><span class="highlight-line"><span class="highlight-cl"> TRCK      : 0
/20
</span></span><span class="highlight-line"><span class="highlight-cl"> TALB      : M
lody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> TCON      : &
t;unknown&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> TSOT      :
elody Line
</span></span><span class="highlight-line"><span class="highlight-cl"> TSOA      :
elody Line(s)
</span></span><span class="highlight-line"><span class="highlight-cl"> TSOP      :
mileR feat. はつねみく
</span></span><span class="highlight-line"><span class="highlight-cl"> TSSE      : La
f58.29.100
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0: A
dio: mp3, 44100 Hz, stereo, fltp, 320 kb/s
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:1: Vi
eo: png, rgba(pc), 1500x1499, q=2-31, 90k tbr, 90k tbn, 90k tbc (attached pic)
</span></span><span class="highlight-line"><span class="highlight-cl"> Metadata:
</span></span><span class="highlight-line"><span class="highlight-cl"> comment
Other
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:2: Vi
eo: png, rgba(pc), 1500x1499, q=2-31, 25 tbr, 25 tbn, 25 tbc
</span></span><span class="highlight-line"><span class="highlight-cl">Stream mapping:

```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:0 -&gt;
#0:0 (copy)
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #0:1 -&gt;
#0:1 (copy)
</span></span><span class="highlight-line"><span class="highlight-cl"> Stream #1:0 -&gt;
#0:2 (copy)
</span></span><span class="highlight-line"><span class="highlight-cl">Press [q] to stop, [
] for help
</span></span><span class="highlight-line"><span class="highlight-cl">frame= 1 fps=0.
q=-1.0 Lq=-1.0 size= 14599kB time=00:03:21.03 bitrate= 594.9kbits/s speed=4.27e+03x
</span></span><span class="highlight-line"><span class="highlight-cl">video:6743kB aud
o:7855kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.009018%
</span></span></code></pre>
```

<p>Again, these output are for <code>stderr</code> only, nothing is printed to <code>std out</code>.</p>

<p>You can manipulate metadata and the cover art in the same command by just adding <code>-metadata key=value</code> options. Note that if you have <code>-map 0:0</code> instead of <code>-map 0</code> for the audio file, you may lose your existing tag data. Only add the extra <code>:0</code> if that is what you want to do.</p>

<hr>

<p>FFmpeg can get rid of the hassle if you are in a hassle looking for an all-in-one solution for music tag reading/writing. It might be a little too heavy if you are using it only for this purpose, but it shouldn't be much of a problem if you already have it installed in your system for some other things.</p>