

Crop MP3 to first 30 seconds

Original Question

91

I want to be able to generate a new (fully valid) MP3 file from an existing MP3 file to be used as a preview -- try-before-you-buy style. The new file should only contain the first n seconds of the track.



Now, I know I could just "chop the stream" at n seconds (calculating from the bitrate and header size) when delivering the file, but this is a bit dirty and a real PITA on a VBR track. I'd like to be able to generate a proper MP3 file.



39

Anyone any ideas?

Answers

Both `mp3split` and `ffmpeg` are both good solutions. I chose `ffmpeg` as it is commonly installed on linux servers and is also [easily available for windows](#). Here's some more good command line parameters for generating previews with `ffmpeg`

- `-t <seconds>` chop after specified number of seconds
- `-y` force file overwrite
- `-ab <bitrate>` set bitrate e.g. `-ab 96k`
- `-ar <rate Hz>` set sampling rate e.g. `-ar 22050` for 22.05kHz
- `-map_meta_data <outfile>:<infile>` copy track metadata from infile to outfile

instead of setting `-ab` and `-ar`, you can copy the original track settings, as Tim Farley suggests, with:

- `-acodec copy`

mp3

ffmpeg

edited Dec 16 '17 at 7:28



Sridhar Sarnobat

9,379 8 63 63

asked Sep 4 '08 at 14:35



Cheekysoft

26.9k 17 62 82

Just a thought: you may want to skip the beginning of the original song. Say, you can use 30 seconds piece starting at the *third* of the song. In some songs, the first 30 seconds doesn't tell you much as it's just a "setting up the scene" part - for instance *Pink Floyd's* [Shine On You Crazy Diamond](#). –

[Tomas Sedovic](#) Sep 4 '08 at 15:18

Just as note, I had a really bad time with ffmpeg, mptsplit and mp3cutter on CentOS, all of them reporting "Header Missing". So, I had to go with MP3 Class and some maths to cut by filesize. – [Gardner](#) Feb 28 '13 at 19:36

I didn't realize ffmpeg could do that. Thank you. – [Sridhar Sarnobat](#) Dec 16 '17 at 7:28

8 Answers



126



I also recommend ffmpeg, but the command line suggested by John Boker has an unintended side effect: it re-encodes the file to the default bitrate (which is 64 kb/s in the version I have here at least). This might give your customers a false impression of the quality of your sound files, and it also takes longer to do.

Here's a command line that will slice to 30 seconds without transcoding:

```
ffmpeg -t 30 -i inputfile.mp3 -acodec copy outputfile.mp3
```

The -acodec switch tells ffmpeg to use the special "copy" codec which does not transcode. It is lightning fast.

NOTE: the command was updated based on comment from Oben Sonne

edited Dec 15 '14 at 22:01



[Jiri Kremser](#)

8,883 5 34 64

answered Sep 4 '08 at 15:42



[Tim Farley](#)

9,937 4 25 30

19 Recent ffmpeg versions (at least since 0.10.6) interpret options differently based on position. Input-related options must appear before -i ... and output-related options must appear after -i ... This means ffmpeg -t 30 -i inputfile.mp3 -acodec copy outputfile.mp3 would now be the correct answer. – [Oben Sonne](#) Dec 30 '12 at 20:41

Can i extract 30 second audio from 15 second input file, it means if file grater then 30 second then extract 30 second but if file 15 second then repeat and extract complete 30 second. – [Girish Patidar](#) Jan 11 '16 at 10:01

5 If you want to set starting time (offset) you can use `-ss hh:mm:ss[.xxx]`. Example: `ffmpeg -t 30 -ss 00:00:15.500 -i inputfile.mp3 -acodec copy outputfile.mp3` will slice to 30 seconds starting from 00h 00m 15s 500ms. – [patryk.beza](#) May 9 '16 at 15:08

1 for some reason above command didn't worked for me but if i changed it to: `ffmpeg -i inputfile.mp3 -t 30 -acodec copy outputfile.mp3` then it's working perfectly – [Krishnendu](#) Jun 28 '16 at 5:27

For me the -t and -s options must be used separately in 2 commands. But it works, thanks! – [Yan King Yin](#) Jan 21 '17 at 14:25

This command also works perfectly. I cropped my music files from 20 to 40 seconds.

9

-y : force output file to overwrite.

```
ffmpeg -i test.mp3 -ss 00:00:20 -to 00:00:40 -c copy -y temp.mp3
```

edited Mar 12 at 21:20



petezurich

4,162 8 20 36

answered Jun 14 '17 at 6:03



Rahul Chauhan

533 6 10

1 Perfect! Thanks for sharing this. – [HerrimanCoder](#) Aug 26 '18 at 23:39

This is the correct answer. – [Viktor Joras](#) Nov 25 '18 at 11:50

I have got an error while doing the same

0

Invalid audio stream. Exactly one MP3 audio stream is required.
Could not write header for output file #0 (incorrect codec parameters ?): Invalid
argumentStream mapping:

Fix for me was:

```
ffmpeg -ss 00:02:43.00 -t 00:00:10 -i input.mp3 -codec:a libmp3lame out.mp3
```

edited Jul 22 '18 at 2:23



JakeGould

21.2k 9 54 79

answered Dec 2 '16 at 2:41



Mithun Cheriya

1 1

If you wish to REMOVE the first 30 seconds (and keep the remainder) then use this:

```
ffmpeg -ss 30 -i inputfile.mp3 -acodec copy outputfile.mp3
```

50

edited Jul 25 '15 at 10:39



Tim

259 2 13

answered May 27 '10 at 23:30



the.jxc

2,651 15 19

you can use mp3cut:

11

```
cutmp3 -i foo.mp3 -o 30s.mp3 -a 0:00.0 -b 0:30.0
```

It's in ubuntu repo, so just: `sudo apt-get install cutmp3`.

answered Sep 19 '11 at 0:15



Michał Šrajer

22.2k 4 44 68

You might want to try [Mp3Splt](#).

5

I've used it before in a C# service that simply wrapped the mp3splt.exe win32 process. I assume something similar could be done in your Linux/PHP scenario.

answered Sep 4 '08 at 14:45



Ryan Duffield

14.1k 5 34 38

2 Used MP3Split with success for mp3 cut – [Freshblood](#) Apr 10 '14 at 19:20

try:

17

```
ffmpeg -t 30 -i inputfile.mp3 outputfile.mp3
```

answered Sep 4 '08 at 14:45



**John Boker****69.7k** 15 85 127

I haven't used it for this specific purpose, but I bet [ffmpeg](#) can do it.

3

answered Sep 4 '08 at 14:38

**grapefrukt****19.8k** 4 41 67