Unix & Linux Stack Exchange is a question and answer site for users of Linux, FreeBSD and other Un\*x-like operating systems. Join them; it only takes a minute:

Join

Here's how it works:

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

# Unix & Linux

## Trim audio file using start and stop times

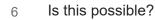


I have an FFmpeg command to trim audio:

24 ffmpeg -ss 01:43:46 -t 00:00:44.30 -i input.mp3 output.mp3

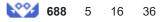


The problem I have with this command is that option -t requires a duration (in seconds) from 01:43:46. I want to trim audio using start/stop times, e.g. between 01:43:46 and 00:01:45.02.



audio ffmpeg trim





- 1 So you want to extract a section of an audio file using start and stop times instead of a start time and a duration, is that correct? user86969 Feb 3 '15 at 9:35
- 1 @Nasha That's correct. I'll edit the post to make that clearer. user8547 Feb 3 '15 at 9:51
- 1 I have rephrased your question accordingly. Indeed ffmpeg doesn't seem to provide anything else than a start time and a duration. And mplayer doesn't either. user86969 Feb 3 '15 at 10:13

#### 2 Answers



ffmpeg seems to have a new option -to

34

-to position (output)
 Stop writing the output at position. position must be a time duration
specification, see the Time
 duration section in the ffmpeg-utils(1) manual.



-to and -t are mutually exclusive and -t has priority.

### Sample command with two time formats

```
ffmpeg -i file.mkv -ss 20 -to 40 -c copy file-2.mkv ffmpeg -i file.mkv -ss 00:00:20 -to 00:00:40 -c copy file-2.mkv
```

This should create a copy (file-2.mkv) of file.mkv from the 20 second mark to the 40 second mark.

answered Aug 10 '16 at 5:59



2 Tx, that works nicely. I find that if I don't remove the copy part the audio/video is out of sync, so I remove that and add -async 1 and it works like a charm. Of course it requires reencoding, so an improvement would be a command that would allow for the copy but without the sync issues. –

```
.... - Avner Barr Jul 17 18 at 11:47
```

@AvnerBarr I don't know if this is possible solely with FFmpeg, but I know it's possible with Mencoder. This works, using the -to option: ffmpeg -i mmm.mp4 -ss 00:04:04 -to 00:04:35 -c copy output1.mp4 && ffmpeg -i mmm.mp4 -ss 00:05:47 -to 00:06:05 -c copy output2.mp4 && mencoder -ovc copy -oac mp3lame -idx output1.mp4 output2.mp4 -o complete.mp4 - user8547 Oct 31 '18 at 15:15



If you have PHP installed, make it a script:



TimeDiff.php contents:



```
<?php
// Create datetime objcects
$dt1 = new DateTime($argv[1]);
$dt2 = new DateTime($argv[2]);

// Conver difference to seconds
$dt3 = $dt2->format('U') - $dt1->format('U');

// echo $dt3."\n";
$h = (int)($dt3 / 3600);
$dt3 %= 3600;
$m = (int)($dt3 / 60);
$dt3 %= 60;
$s = $dt3;

// Dump as H:M:S
echo $h . ":" . $m . ":" . $s;

?>
```

audiochop.sh contents:

```
#!/bin/bash
INFILE=$1
START=$2
STOP=$3
OUTFILE=$4

OFFSET=`php TimeDiff.php "$START" "$STOP"`
```

#### Usage:

```
./audiochop.sh [input.mp3] [startchop] [stopchop] [output.mp3]
```

Where [startchop] and [stopchop] are both absolute timestamps from the beginning of the track.

NB: Script(s) may need tweaking depending on platform version etc...

answered Feb 8 '15 at 2:17



**8** 1

So does the command look like the following example? audiochop.sh input.mp3 00:01:30 00:01:50 output.mp3 - user8547 Feb 8 '15 at 5:01

Yes, exactly that. - Litch Feb 9 '15 at 5:39

This is actually the full command: ./audiochop.sh input.mp3 00:01:30 00:01:50 output.mp3 I had forgotten to put the period and slash in front of the command, which is why it wasn't working. — user8547 Feb 9 '15 at 7:40

Incidentally, I found that this php/bash script also works on mp4 videos, so I assume it will work with any video format. - user8547 Feb 10 '15 at 1:04

1 It should work with whatever format your ffmpeg supports. – Litch Feb 10 '15 at 10:42