

FAAC

Freeware Advanced Audio Coder - FAAC 1.24+

Usage: FAAC.EXE [options] infile ...

Quality-related options:

- q <quality> Set default variable bitrate (VBR) quantizer quality in % (default: 100, averages at approx. 120 kbps VBR for a normal stereo input file with 16 bit and 44.1 kHz sample rate; max. value 500, min. 10).
- b <bitrate> Set average bitrate (ABR) to approximately <bitrate> kbps (default: VBR mode; max. value 152 kbps/stereo with a 16 kHz cutoff, can be raised with a higher -c setting).
- c <freq> Set the lowpass cutoff in Hz (default: automatic, i.e. adapts maximum value to input sample rate according to the Nyquist theorem; changes with different -q settings, but can also be set manually; 0 means no cutoff, only for testing purposes).

Input/output options:

- <stdin/out> If you simply use a hyphen/minus sign instead of an input file name, FAAC can encode directly from stdin, thus enabling piping within other applications like foobar2000. The same works for stdout as well, so FAAC can pipe its output to programs like mp4live (streaming live AAC content).
- o X Set output file name to X (only for one input file; you can use *.aac, *.mp4, *.m4a or *.m4b as file extension, and the correct file format MPEG-2 AAC ADTS or MPEG-4 AAC in the MP4 container will be chosen automatically then).
- P Raw PCM input mode (default: off, i.e. expecting a WAV header; necessary for input files or bitstreams without a header; using only -P assumes the default values for -R, -B and -C in the input file).
- R Raw PCM input sample rate in Hz (default: 44100 Hz, max. 96 kHz).
- B Raw PCM input sample size (default: 16, also possible 8, 24, 32 bit fixed or float input).
- C Raw PCM input channels (default: 2, max. 33 + 1 LFE).
- X Swap raw PCM input byte order (default: bigendian)
- I <C[,LFE]> Input multichannel configuration (default: 3,4 which means Center is third and LFE is fourth like in 5.1 WAV, so you only have to specify a different position of these two mono channels in your multichannel input file or bitstream if they haven't been reordered already, e.g. in the decoder).

MP4 specific options (muxing and tagging):

- w Wrap AAC data in MP4 container (default: *.m4a file extension, *.mp4 or *.m4b also possible; enables direct MP4 file writing without a separate file muxer; necessary for gapless MP4 files and for multiplexing FAAC output with MPEG-4 Systems content in MP4Box from the GPAC project).
- artist "X" Set artist to X, always use quotation marks around tags with spaces.

FAAC

--writer "X" Set writer to X.
--title "X" Set title to X.
--genre "X" Set genre to X.
--album "X" Set album to X.
--track "X" Set track to X (number/total).
--disc "X" Set disc to X (number/total).
--year "X" Set year to X.
--comment "X" Set comment to X.
--cover-art "X" Read cover art from file X (including path), supported image formats are GIF, JPG and PNG.
--compilation Set compilation flag.

Expert options:

--mpeg-vers X AAC MPEG version, X can be 2 or 4 (default: MPEG-2 for *.aac files, using the -w switch or the *.mp4/*.m4a/*.m4b file extension automatically defaults to MPEG-4).
--obj-type X AAC object type, X can be LC (Low Complexity, default), Main or LTP (Long Term Prediction).
-r Raw AAC output mode (only for testing purposes, creates AAC bitstream without ADTS or other headers similar to the one in a MP4 container, so AAC file properties will not be detected correctly in FAAD2 then).
--no-tns Disable Temporal Noise Shaping.
--shortctl X Enforce block type (default: both; 1 = short only; 2 = long only).
--no-midside Don't use M/S matrixing.

Documentation:

--license Show the FAAC license.
--help Show abbreviated help.
--long-help Show complete help.

Sound-related recommendations

The VBR bitrate depends on -q **AND** -c, so you should only vary the default setting -q 100 -c 16000 if you know what you're doing and/or want to experiment with other cutoff frequencies at a given quality setting.

This behaviour is also used for the ABR setting with -b which results in an average bitrate without a bit reservoir. It adjusts the outcome by using a specific cutoff frequency for each -b setting and varying the quantizer quality internally until the desired overall bitrate is reached. Nevertheless the usual

FAAC

remarks about ABR vs. VBR apply here, too, so this setting is only recommended for special purposes where a predictable bitrate is more important than sound quality, e.g. live streaming or obtaining exact file sizes. An exception to this rule ("Don't use ABR if you don't have to!") is the handling of typical problem samples like harpsichord or other solo instruments that usually drop too low in bitrate when using the default -q 100 and can sound a little bit better with -b 128 for example.

The following table should give some orientation for useful -q and -c settings (the latter changes automatically with -q now, but can still be set individually), based on FAAC v1.24+. The VBR bitrates refer to an average sounding stereo input file with 16 bit sample size and 44.1 kHz sample rate, i.e. [ct_reference.wav](#) in this case. Multiplexing these AAC files to [MP4](#) will result in a **~3 kbps lower bitrate** because of the stripped [ADTS](#) headers, unless you use FAAC's -b switch together with -w (or -o with *.mp4 file extension) which already calculates this overhead.

recommended setting:	without resampling:	resampled at:	results in:
-q 150 (-c 22000)	~172 kbps *.aac		
-q 125 (-c 19000)	~146 kbps *.aac		
-q 100 (-c 16000)	~120 kbps *.aac	32 kHz	~112 kbps *.aac
-q 75 (-c 13000)	~93 kbps *.aac	32 kHz	~88 kbps *.aac
-q 65 (-c 11800)	~84 kbps *.aac	24 kHz	~73 kbps *.aac
-q 50 (-c 10000)	~68 kbps *.aac	22.05 kHz	~58 kbps *.aac

When combining the SSRC resampling module in foobar2000 with either FAAC and the CLI encoder or foo_faac.dll (the GUI implementation) it's possible to directly **downsample the input file** while encoding. This way the coding efficiency will be increased = the average bitrate of the same -q setting can be further reduced for cutoff frequencies below 17 kHz with approximately the same sound (i.e. with higher frequency vs. lower time resolution). This is especially true when the lower sample rate exactly halves the original input sample rate and matches the cutoff frequency according to the Nyquist theorem, e.g. input with 44.1 kHz down to 22.05 kHz and cutoff at 11 kHz or 48 -> 24 -> 12 kHz in AC-3 transcodings from DVD. As usual you should test which -q settings with almost equal bitrates and cutoffs (e.g. -q 90 downsampled to 24 kHz or an unchanged -q 70) sound better to you, and don't forget to upsample your test WAV files to 44.1 kHz again for a fair comparison. You shouldn't use any dithering during this test either, because it could affect the sound of the downsampled files.

You should also test if those very **high cutoff frequencies** for the higher quality settings make sense to you, i.e. if you can really hear a difference. Otherwise it might be smarter to save some bits with a lower cutoff and/or gain a better overall quality at the same bitrate with a higher -q value instead (e.g. -q 150 -c 16000 = ~142 kbps/stereo instead of ~172 kbps/stereo).

Some useful FAAC settings for direct **transcoding of multichannel AC-3** movie soundtracks from a DVD to MP4 can be found in [this thread](#) on Doom9.org (using foo_faac.dll in foobar2000 then). A nice <http://forum.doom9.org/showthread.php?s=&threadid=84915> guide with foobar2000 screenshots for transcoding DTS to MP4 is also available there (using foo_dts.dll instead of foo_ac3.dll as input plugin then).

These settings (e.g. -q 75 resulting in ~230 kbps/6ch) are also recommended for FlashROM-based **portables** like the Diva GEM or newer Nokia mobile phones, because they still sound good on these devices, especially when using open headphones in a noisy environment while saving a lot of space resulting in ~80-100 kbps/stereo VBR then.

Two **useless settings** would be the maximum and minimum -q values, but they are mentioned here to show their limits:

-q 500 -c 22050	~277 kbps *.aac	resampled at:	results in:
-q 10 -c 3000	~26 kbps *.aac	8 kHz	~9 kbps *.aac

Especially the lowest quality setting doesn't make sense anymore, because the sound is unusable, at least with the input file properties described above. If you want to try lower quality settings than -q 75, you should always add the SSRC resampling module from foobar2000 (see further explanation below).

An exception to this is **speech coding** with FAAC, because you should not use stereo 16 bit input WAV files or PCM bitstreams sampled at 44.1 kHz then, but rather mono 16 or 8 bit PCM input with 8 kHz sample rate. These properties are usually considered as "original sound" when comparing specialized speech codecs, so you should also set them in your input file if you want to use FAAC for this purpose. This will increase the coding efficiency a lot, because many input bits are discarded due to the low sample rate. So an useful setting for speech coding (~10 kbps/mono) would be:

FAAC

-b 10 -c 3500 input.wav -o output.mp4

The usage of -b rather than -q is recommended here, because normally you'll only need these low bitrates for audio live-streaming in a network where an almost constant bitrate is more important than a better sounding VBR quality.

Other recommendations

- Since the implementation of direct MP4 file writing and tagging in FAAC a lot of handling issues have become easier, e.g. which [MPEG](#) flag to use in the AAC ADTS header or "what did Apple and Nullsoft know" and so forth. Now it's possible to create AAC and MP4 files with the internal FAAC options that will most likely play with your preferred player, be it software or hardware, but not necessarily with all available AAC/MP4 players around due to the possible variants in the MPEG-4 Audio standard.
- The default output file format of FAAC not using any switches is still MPEG-2 AAC ADTS LC which will play on most hardware AAC players like Philips Expanium, Diva GEM and mobile phones from Nokia or other manufacturers.
- This does not include the iPod of course that only reads MPEG-4 AAC LC bitstreams in the MP4 container with the file extensions *.m4a, *.m4b and *.m4p (but not the standard *.mp4 extension). See the [MP4](#) Wiki page for a short explanation of these different proprietary extensions. As it neither decodes MPEG-2 AAC LC nor the Main, LTP or HE AAC profile, not even in a MP4 container, you have to stick to the LC profile and the MP4 file format. So using either the -w or -o switch with appropriate extensions is mandatory, also in GUIs that offer related file format or profile options. Gapless playback is not possible on the iPod or in iTunes yet.
- [Winamp](#) 2.95 came with an own "preliminary AAC support" via Nullsoft's in_mp3.dll which didn't know MPEG-4 AAC or MP4 files either. From v5.02 on Winamp also supports playing audio-only MP4 files with an additional file splitter having the same name as the input plugin from Audiocoding.com (in_mp4.dll) as well as ripping CDs to *.m4a files with a codec from Dolby. They also added a HE AAC decoder from Coding Technologies recently.

So AAC LC and MP4 files produced with FAAC should be playable in new Winamp versions, although the MP4 tags are not supported yet (only ID3 tags on AAC files). HE AAC bitstreams in MP4 files done with Nero will only be decoded without the SBR part if you don't substitute Nullsoft's in_mp4.dll with the one from Audiocoding.com which would also enable gapless playback and MP4 tag reading in Winamp.

Furthermore seeking within an AAC file doesn't work, so your best option is to encode to *.m4a right from the start which will also keep the necessary info for gapless playback in the MP4 file header. You can also do this with the Winamp output plugin out_aac.dll which is based on FAAC v1.24+ and enables direct transcodings from any input format unlike Nullsoft's AAC codec which only works for CD ripping.

- MP4Box, a new MP4 file muxer from the open source [GPAC project](#) is the first command line app that can create MPEG-4 [Systems](#) content without an installed Java VM (unlike mp4tool from the ENST or the IBM Toolkit for MPEG-4). But it cannot multiplex normal AAC files to MP4 like [mp4creator](#), so you have to create a MP4 file with FAAC's -w or -o option first.
 - If you want to play FAAC files on your Xbox player console, you have to use -o and the *.mp4 file extension, too, because it cannot play plain AAC files either.
-