

Flac Faq

[FLAC](#) has now been used widely at etree.org and is the best choice for new seeds or re-seeds. Check out the pages [FLAC](#), [FlacFrontend](#), [FlacFingerprint](#), [NamingStandards](#) and [SeedingGuidelines](#). If your question is not covered, ask it here on this question page. Questions and answers may later be edited for clarity, and may be incorporated into the other pages. Mark off each item with 4 dashes.

Making Files

Q: Should I change all my shns to 16-bit flac?

A: No, there's no point. The shns are still perfectly fine. Keep trading them as they are. Simply duplicating them as flac sets will only cause confusion in the trading pool.

Q: Should I change all my shns to 24-bit flac?

A: No, shns are already 16-bit. It's not possible to gain audio quality.

Q: What settings should I use for flac or in [FlacFrontend](#)?

A: A typical user might want to use Level 6 - the compression level / time tradeoff seems good based on [\[this analysis\]](#) at [\[Hans Heijden's site\]](#) and [\[Speek's site\]](#). Higher levels get slightly smaller file sizes, if you have a fast computer or more time. Remember that seeds are encoded once, but decoded many times - so like all the seeding guidelines, it's worth doing right. And one of the nice features of flac is that decode times are the same fast speed, regardless of the level of compression.

Check the Verify option for seeds to add a little extra check against seeding shows with an encoding error.

Q: What about aligning on sector boundaries?

A: It's probably best to seed and track your show to avoid sector boundary errors - see the discussion at [SeedingGuidelines](#). But using the flac align feature does seem to be a workable way to fix SBEs - see [\[this etree forums thread on SBEs\]](#). **If you use Align On Sector Boundaries, you have to be absolutely certain that the files sort in the right order.** Otherwise, parts of the tracks will be moved to the wrong other tracks.

Q: What about tags? What extra data can be stored in FLAC files?

A: See [FlacMetadata](#).

Q: What's that [ReplayGain](#) setting for?

A: Using [ReplayGain](#) is an option in flac and other file formats. The idea of Replay Gain is to calculate a volume adjustment that players (hardware and software) can use to even out jarring volume differences between songs.

The short answer on Replay Gain is, it's your call. If you have a fast computer, it doesn't add much processing time. If you do decide to use Replay Gain when you're encoding a show, be sure to use the "Treat input files as one album" option, and encode the whole show at once, to preserve the volume levels of each track relative to others in the show.

If you'd rather, you can safely ignore that option and leave it un-checked. Most people won't benefit from it - it's only used when flacs are played from the flac file, and a quick twist of the volume control does just as well for those listening to a whole show. It's really more use for mixes of single tracks.

[FlacFingerprint](#) Issues

Q: A common practice for SHN files is to verify the MD5's right after download or after creating a newly burned disc. Is there an equivalent procedure with FLACs and FLAC fingerprints?

A: Yes, just run drag the files into [FlacFrontend](#) and run Test (or use "flac -t" from a command line) to test flac files on a CD, DVD or hard drive. If they pass this test, they are good copies. It may take longer than a typical md5 check. Alternatively, you can just decode the files; flac tests them as they decode, and will alert you if there is an error.

If the flac copies come with md5s or a par2 file, you can of course use those for a check. Similarly, all complete [BitTorrent](#) downloads are automatically verified to confirm that you have an exact copy of the original files. Unless there is an error in the source files, these approaches will also confirm that you have a good copy.

Q: Scroll off during testing: When I load in some tracks into [FlacFrontend](#) and hit Test,

another DOS window pops up and the flac starts testing. The window that pops up does not let me scroll up and down, and after a track is tested and flac moves on, that line 'ok' scrolls up and is gone after the next few tracks. If I am not watching the program, how do I know if there is a problem with one track early on?

A: This problem can be fixed by adjusting the properties of your DOS window (that black window that pops up when you click Test). When the black window appears, right click on the upper left corner and select Properties. Under Screen Buffer Size, increase the Height value to something large, the maximum is 9999.

Q: Why isn't the [FlacFingerprint](#) file used to automatically verify the uniqueness of a seed? The end-user should not have to manually export and visually compare a [FlacFingerprint](#) file for each and every seed that is download to verify that the audio contained therein has not been altered.

also

Q: What is the point of a [FlacFingerprint](#) file? When verifying with "flac -t", it uses the checksum values inside of the flac files (not the [FlacFingerprint](#) file). If you want to "see" the fingerprints you can just run metaflac. Keeping this around in a text file seems completely pointless since these values are already stored in the files themselves.

A: The ffp.txt can be used in a quick visual comparison to those stored in the [ShnDatabase](#), as an identifier to see which seed you might have *versus others*. That's why folks have taken to calling it a fingerprint. If the seeder has made this file already, it saves people a step. The file only takes up ~1KB, and people who don't care can always ignore it.

In addition, the flac fingerprint uses a checksum of only the music data in the file, the same calculation used by shntool md5. This provides a format-independent way to compare files or shows, which should be the most useful method in the long term.

Extra benefit to having ffp.txt files for the [ShnDatabase](#): The [\[Live Music Archive\]](#) show importing software can automatically refer to them in the database. This means faster turnaround on getting music uploads made public, and far less burden on archive.org admins. That same fingerprint crossreferencing also enables the "Download this show from the Llama" links you see at db.etree.org.

Q: When I generate a [FlacFingerprint](#) file (via metaflac --show-md5sum), should I remove all the path information in the resulting text file, or leave it as is?

A: It's best to have any path information be relative to the top level .flac16/.flac24 directory, so when creating these files all you should need to do is to run "metaflac --show-md5sum *.flac >

showffp.txt" from the directory where the flac files live, and there should be no extraneous path information in the output.

Q: Should I generate md5s for my flacs, too?

A: It is not required, and might not be all that useful. (For example, [BitTorrent](#) automatically verifies that you have all the files in a torrent and that they are all complete and exact copies before saying that a download has reached 100% complete.) However, for some users it may still be beneficial to have md5s for verifying files on remote servers that lack flac tools, or to speed/smooth processing of contributions to the [\[Live Music Archive\]](#). See [FlacFingerprint](#) for compare and contrast.

Q: What if I find that the md5s someone generated don't match after all?

A: It is possible that someone, during later circulation, has simply changed the compression ratio of the file or its [FlacMetadata](#), which does not affect the audio. If the FFP is ok, then the integrity of the audio is intact, and the metadata or the compression ratio is the only thing that has changed.

Decompressing Files

Q: As I drag and drop files, everything is fine. When I hit decode, it says: Got error code 0: Flac stream decoder error status lost sync. Error while decoding metadata state =4 end of stream. What does this mean and how can I decode without getting this error message?

A: Error codes generally mean your file is corrupt. Try performing flac -t (Test in [FlacFrontend](#)). Get a fresh copy if there is a problem.

If you used [BitTorrent](#) to get the files, decoding errors are often an indication that your copy did not reach 100% completion. Try re-opening the torrent and having your [BitTorrent](#) software re-check the files. If they are not 100% complete, re-start the torrent and try to complete the download.

Q: I recently downloaded FLAC frontend and sent some .flac16.torrent files to be decompressed, however the program keeps giving me a box that says "Only FLAC and Ogg-FLAC files can be decoded!" I don't know why this won't work, as my files seem to be FLAC files. Any suggestions?

A: Any files ending with extension .torrent are support files for running the [BitTorrent](#) protocol to download the set of files. That is, the small .torrent file has only a list of the filenames and sizes, and verification information - it doesn't have all the music!

When you run a .torrent file, it normally creates a file set on the hard drive and then begins to download pieces of the file. After a [BitTorrent](#) download is 100% completed, look on your hard drive for the files ending with .flac, then decompress those.

Normally you will not have any complete tracks until the torrent reaches 100% completion - pieces of each file will be missing - although some [BitTorrent](#) software will allow you to try to complete certain tracks first by making those a priority.

Other Questions

Q: Why is etree.org using FLAC instead of APE (Monkey's Audio)? Both support 24-bit files.

A: There are a few reasons. First, the FLAC project is completely free and open source and has been so from the start; the Monkey's Audio code has only recently been released to the public, but this release is incomplete (no GUI) and support for it could be dropped at any time. Also, the FLAC code has been ported to more platforms than the Monkey's Audio code, which ensures that more people can access the music in that format.

That said, it appears that the author of APE is making a good effort at cross platform compatibility and releasing source code, so it could be a viable alternative later on.

Q: How do I get [WinAmp](#) to play FLAC files ?

A: You need a small "plugin" file to help [WinAmp](#) decode the flac files, to play them from a hard drive or disc. You can get the plugin a few different ways:

- If you have [WinAmp](#) installed, run the [FlacFrontend](#) installer and it will also install the flac plugin.
- The official [\[FLAC binary distribution\]](#) for Windows contains an input plugin that works in both Winamp 2 and 5. Copy the in_flac.dll file into your [WinAmp](#)\plugins folder and restart [WinAmp](#).
- Or, download of the unofficial in_flac.dll plugin at [\[the Rarewares lossless page\]](#)

There is also an installer containing an obsolete version of this plugin on the [\[WinAmp website\]](#).

The plugin works with [WinAmp](#) 2 and 5. There is no known way to play flac files using [WinAmp](#) 3.

Q: My Flac Front End used to work. Now, whenever I start the program, a window blips on the screen for a tiny fraction of a second and disappears. The GUI interface never starts. I have tried everything I can think of, including uninstalling and reinstalling, shutting off all firewalls, etc, but nothing is working. Any suggestions or ideas would be appreciated. Thanks!

A: I recommend you search for answers in the FAQs here, and if you don't see what you're looking for, check the forums: <http://forums.etree.org> There is a whole section there dedicated to help topics where more people may feel able to respond -Damien

Q: How can I find more specific information for Linux ?

I am hoping someone well versed with FLAC under Linux will edit this page. But if I can find the answers on my own I will do my best to update this page myself. Peace - Tim

Q: I understand FLAC typically compresses 16-bit WAV files by 50% or so. How does it do with 24-bit files from live recordings, that are not necessarily made with the very quietest mics and preamps?

A: It depends entirely on the music and recording method. You'd really have to try it and see.

Roughly speaking, compression depends on the variety and complexity of the music (as captured) and the correlation between the channels. File sizes of 55-65% of the wav files are probably more typical for recent 16-bit recordings (with a full range of frequencies and true stereo sound). Older sources with a less-than-full range of frequencies, that are mono or nearly so, might compress to as little as 30-40% of the size of the wav files.

Q: What's up with converting FLAC to SHN??? Is this safe and lossless??? How should I handle this in my text file and lineage to avoid any confusion among the various seeds??? Is there a one-step method to do this, or do I have to decompress my SHNs to WAV then recompress to FLAC???

A: There will be no quality loss going from SHN to FLAC or vice versa.

Re: trading, so long as you make a note in your .txt file like: "This is SHN ID # _____, converted to FLAC on [date] by [me]" then most folks won't be too concerned. The .ffp data should match up to the .md5 data for the original SHN seed, anyway, but you can also add to the bottom of the .txt file something like "The original .md5s were: [list them all]" which will also give people comfort that you know what you're doing.

Re: doing it all in one step, shntool will do it with the "-o flac" parameter. The command would look something like this: "shntool conv -o flac *.shn".

Taken from a thread on the ETREE.FORUMS:

<http://forums.etree.org/viewtopic.php?t=4615&highlight=shn+wav+flac>

(written by MGoldey)

This Page Last Changed: Jun 13, 2005 12:57:22

[Edit this page](#) · [View page history](#)

All content written by members of the etree.org community, for the etree.org community.
© 1998-2007, All Rights Reserved.