FLAC2MP3 batch FLAC to MP3 converter

# Overall process and suggested tools to be used to rip CDs and build a library in FLAC and MP3

The following process has been worked out and found to be useful

Rip CDs with Exact Audio Copy either on Windows or Linux platform and save files as FLAC files along with FLAC and ID3v2 tags.

If Album arts cannot be found with built in freedb access, Google Images almost always provides an image for use

If Album art to be added from Google results or other FLAC ID, or ID3v2 fields are to be modified, Mp3Tag is a powerful utility under Windows.

Postprocessing is done in Linux.

Mp3 Encoding uses Lame encoder and the flac2mp3 script which takes care of the proper coding and adding ID3v2 tags as well as album art.

Cataloging can also be run in Linux using the flaclist script, which produces .csv output which can be formatted e.g. in Excel.

Linux uses UTF-8 character encoding, while Windows is using Unicode. In order to get accented characters displayed properly under Windows, use the utf2uc converter (now running in Windows command prompt)

# EAC settings and other ripping related issues

FLAC optional parameters for use within EAC:

-8 -V -T "ARTIST=%artist%" -T "TITLE=%title%" -T "ALBUM=%albumtitle%" -T "DATE=%year%" -T "TRACKNUMBER=%tracknr%" -T "GENRE=%genre%" -T "COMMENT=%comment%" -T "BAND=%albuminterpret%" -T "ALBUMARTIST=%albumartist%" -T "DISCNUMBER=%cdnumber%" -T "TOTALDISCS=%totalcds%" -T "TOTALTRACKS=%numtracks%" %hascover%--picture="%coverfile%"%hascover% %source% -o %dest%

When ripping metadata fields are to be checked, especially for artist, album title, track title, year, genre, album art.

FLAC files should be appended with FLAC ID3v1 and ID3v2 tags, for compatibility with the widest range of devices. Latest ID3v2 specification is 2.4, however it is not widely used, therefore 2.3 is to be used

For albums with multiple artists, Album artist should be set to “Various Artists” and artist for individual tracks are to be set one by one.

For classical music, the composer is kept as artist (album artist), while the performing orchestra, conductor and soloist is noted in parenthesis after the title of the piece.

For Album art 500x500 pixel resolution preferred (though preferably no more than 100 kBs), alternatively 400x400 or 300x300 is acceptable. In case freedb database does not have proper album art, google images almost always have, which needs to be imported later using Mp3tag or similar utility. Album art is stored to all tracks as well as stored as jpg (or png) file in actual album directory with artist – title.jpg name.

Log files should always be saved to current directory with artist – title.log. (Log files under Windows are saved with Unicode (16 bit) character set, while in Linux, the default is UTF-8 (variable length). The catalog script can handle both)

# Conversion from FLAC to MP3

For the conversion Lame encoder is used with target rate of ~130 kbit/s to provide relatively good quality and significantly smaller file sizes as FLAC for music on the go. During the conversion process ID tags in the FLAC files are to be preserved.

Conversion script flac2mp3 can assist in the conversion and ID tags preservation. This bash script runs on Linux and is able to convert music for one album or several albums at a time. Most of the logic in the script is about ID tag conversions.

Tag mapping definition (FLAC to ID3v1 and ID3v2.3)

From the source files FLAC tags are used, and those will be converted to respective ID3v1 and ID3v2 tags for the MP3 file. Since there are more versions ot ID3v2 definitions, the most commonly used ID3v2.3 specification is used. FLAC tag naming seem to change slightly depending on implementation, so we adhere to the EAC FLAC tags, as that seems to be the most widely used “de-facto standard”. The conversion table for the selected tags used by the flac2mp3 script is as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **FLACtags** | **EAC Flac settings** | **ID3v1** | **ID3v2.3 Frame** | **ID3v2.4 Frame** | **Description** |
| ALBUM | ALBUM | ALBUM | ALBUM | TALB | TALB | Album/Movie/Show title |
| ALBUM ARTIST2 |  | ALBUMARTIST |  | TPE2 | TPE2 | Band/orchestra/accompaniment |
| ARTIST | ARTIST | ARTIST | ARTIST | TPE1 | TPE1 | Lead performer(s)/Soloist(s) |
| BAND2 |  | BAND |  | TPE2 | TPE2 | Band/orchestra/accompaniment |
| COMMENT | COMMENT | COMMENT | COMMENT | COMM | COMM | Comments |
| COMPOSER |  | COMPOSER |  | TCOM | TCOM | Composer |
| DATE | DATE or YEAR | DATE | YEAR | TYER |  | Year |
| DISCNUMBER |  | DISCNUMBER |  | TPOS | TPOS | Part of a set |
| GENRE | GENRE | GENRE | GENRE | TCON | TCON | Content type |
| TITLE | TITLE | TITLE | TITLE | TIT2 | TIT2 | Title/songname/content description |
| TOTALDISCS |  | TOTALDISCS |  | TPOS | TPOS | Part of a set |
| TOTALTRACKS |  | TOTALTRACKS |  | TRCK | TRCK | Total tracks |
| TRACKNUMBER | TRACKNUMBER | TRACKNUMBER | TRACK | TRCK | TRCK | Track number/Position in set |
| N/A |  |  |  | APIC | APIC | Cover image |

Most of the fields are self explanatory. Tags ALBUM, ARTIST, TITLE and TRACKUMBER are always assumed to be present and written to target MP3 files. The rest of the tags are written to destination files if they are present in source FLAC file.

In line with the table above here follows the logic used when converting FLAC tags to ID3 (v1, v2) tags the following logic:

If ALBUMARTIST defined then ALBUMARTIST --> (-, TPE2)

Else if BAND defined then BAND --> (-, TPE2)

Else ARTIST --> (-, TPE2)

ALBUM --> (ALBUM, TALB)

ARTIST --> (ARTIST, TPE1)

TRACKNUMBER --> (TRACK, TRCK)

TITLE --> (TITLE, TIT2)

GENRE --> (GENRE, TCON)

COMMENT --> (COMMENT, COMM)

If DATE defined then DATE --> (YEAR, TYER)

Else if YEAR defined then YEAR --> (YEAR, TYER)

If DISCNUMBER defined then DISCNUMBER --> (-, TPOS)

Album art files are not extracted from FLAC tracks. If album art file is available in the album directory, as a variety of file name and type options, the content will be added to the MP3 file as well (ID3v2 tags only). The search order for cover images: TPE2 - TALB.\*, TALB.\*, cover.\* Cover.\* folder.\* Folder.\* front.\* Front.\* where extension can be jpg, png or gif

# LAME encoder settings

Basic principle that MP3s are created for portable/mobile use and quality listening source should be FLAC.

LAME encoder can provide constant, average and variable bitrate coding. For music, VBR coding yields better quality and should be preferred. [For encoding audiobooks or other low bitrate material CBR coding could be considered, but that is out of scope here]

The LAME VBR options and approximate bitrates:

|  |  |  |  |
| --- | --- | --- | --- |
| Switch | Preset | Target Kbit/s | Bitrate range kbit/s |
| -b 320 | --preset insane | 320 | 320 CBR |
| -V 0 | --preset fast extreme | 245 | 220...260 |
| -V 1 |  | 225 | 190...250 |
| -V 2 | --preset fast standard | 190 | 170...210 |
| -V 3 |  | 175 | 150...195 |
| -V 4 | --preset fast medium | 165 | 140...185 |
| -V 5 |  | 130 | 120...150 |
| -V 6 |  | 115 | 100...130 |
| -V 7 |  | 100 | 80...120 |
| -V 8 |  | 85 | 70...105 |
| -V 9 |  | 65 | 45...85 |

Compromise needs to be achieved between quality and file size. Since MP3 files are meant for mobile use, where file size is important the higher bitrates have less meanings. If quality is important then the original FLAC is to be played.

For our purposes –V5 option is selected, which provides an approximately 80% reduction in file size compared to FLAC files.

# Usage of flac2mp3 script

The script converts all \*.flac files under <src-path> to MP3 format using LAME encoder. Result files are placed to <dest-path>/<src-path>, where the default is ../Mp3/<src-path>. The <src-path> should be a relative path for correct operation.

Usage: flac2mp3 [options] <src-path>

Options: -h Print usage

-d Delete FLAC input file after processing

-D<dest-path> Path to output MP3 files (default is ../Mp3)

-Vn Set the target bitrate (quality) of LAME VBR encoder

where n = 0 target bitrate 245 kbit/s

n = 1 target bitrate 225 kbit/s

n = 2 target bitrate 190 kbit/s

n = 3 target bitrate 175 kbit/s

n = 4 target bitrate 165 kbit/s

n = 5 target bitrate 130 kbit/s (default)

n = 6 target bitrate 115 kbit/s

n = 7 target bitrate 100 kbit/s

n = 8 target bitrate 85 kbit/s

n = 9 target bitrate 65 kbit/s

# Usage of catalog generator script

Flaclist script is provided to create a catalog of the entire music library. It generates a CSV output which is then can be formatted and further processed e.g. with Microsoft Excel.

The CSV file contains the following columns:

Directory Name (i.e. Artist - Album)

File Name (i.e Track) - only .flac and .mp3 files are listed, other known files in the directory (album art, .rip) files are omitted. Additional files are listed as Unknown file.

Rip Source - if Directory Name.rip file exists, first 20 characters included, otherwise set to unknown

Rip Quality indicating the result of the EAC ripping process.

Rip Quality Text, same as the field above, but with textual representation

Album Art - if file exists in directory (same specs as in flac2mp3)

For the Rip Quality and Rip Quality Text the following values are possible:

0 - No Log File

1 - Invalid log file content (could be wrong version of EAC)

2 - Error(s) in Process

3 - No Errors, Not in AutoRip DB

4 - No Errors, different from AutoRip DB

5 - No Errors, Partially Accurate

6 - No Errors, Fully Accurate

The list it generates has header row for each album (directory), with all the fields except the Track field (which is left empty for Album header) and under the Album header row all tracks are listed in separate row (with only the Track field filled in).

# How to generate scripts for mass conversion

The following directory structure is an example to be used for storing sound files:

/Flac

/Flac/Audiobook

/Flac/Classical

/Flac/Jazz

/Flac/PopRock

/Flac/Test

/Flac/World

The same structure exists for Mp3 as well.

For mass conversion of a directory (e.g. Jazz) generate a script:

$ cd “/srv/userdata/$MusicArchive/Flac”

$ ls -1 Jazz >Jazz.sh

$ nano Jazz.sh

*Replace* ^ 🡪 ~/flac3mp3 –D ../../Mp3/Jazz “

*Replace* $ 🡪 “

*Save, exit*

$ chmod 755 Jazz.sh

$ cd Jazz

$ ../Jazz.sh | tee ../Jazz.log

When ready, check result of each CD conversion with

$ grep “Conversion process” Jazz.log or

$ grep –B4 “Conversion process” Jazz.log or

# Flat library or hierarchical

The flat library system containing all albums in directories in the format of ‘Artist – Album’ seemed easy and logical, however when the library grows it becomes harder to manage. Another drawback surfaces when we try to select an album for playback on a typical network audio player, the built in display is usually too short to display the entire long directory entry. Pausing at an item these devices usually start to scroll the entry after a certain amount of waiting time, but this makes the selection process quite cumbersome.

Therefore after using the library on various devices for a while a hierarchical structure seemed to be more useful. So instead of having a flat directory system of ‘Artist – Album’ first a directory level is created for ‘Artist’s and underneath it a second level for ‘Album’s.

The ‘flat2hier.sh’ script is used to do that conversion.

# Usage of flat2hier.sh script

The flat2hier.sh script converts the above flat ‘Artist – Album’ structure to the ‘Artist’ as first level and ‘Album’ as second level structure.

Usage: flat2hier.sh [options] <src-path>

Starting it without options or –h option results to print the usage

flat2hier.sh [1.0] - Music library flat to hierarchical converter

Usage: flat2hier.sh [options] <src-path>

Options: -h Print usage

Converts flat music directory structure to a hierachical one

From: To:

Artist1 - Album1 --> Artist1 --- Album1

Artist1 - Album2 |- Album2

Artist1 - Album3 |- Album3

Artist2 - Album1 Artist2 --- Album1

...

Actual usage to convert the standard flat structure:

$ cd “/srv/userdata/$MusicArchive/Flac”

$ ~/flat2hier.sh Jazz/\* | tee Jazz.log

Search for errors in log file:

$ grep "cannot" Jazz.log

Search for ambiguous naming:

$ grep "Album sub-dir" Jazz.log | grep " - "

Once directory system is converted to hierarchical, the flac2mp3 and flaclist tools will not work properly. Those would need to be modified, but that work is pending.