

---

# Quod Libet

Nov 29, 2019



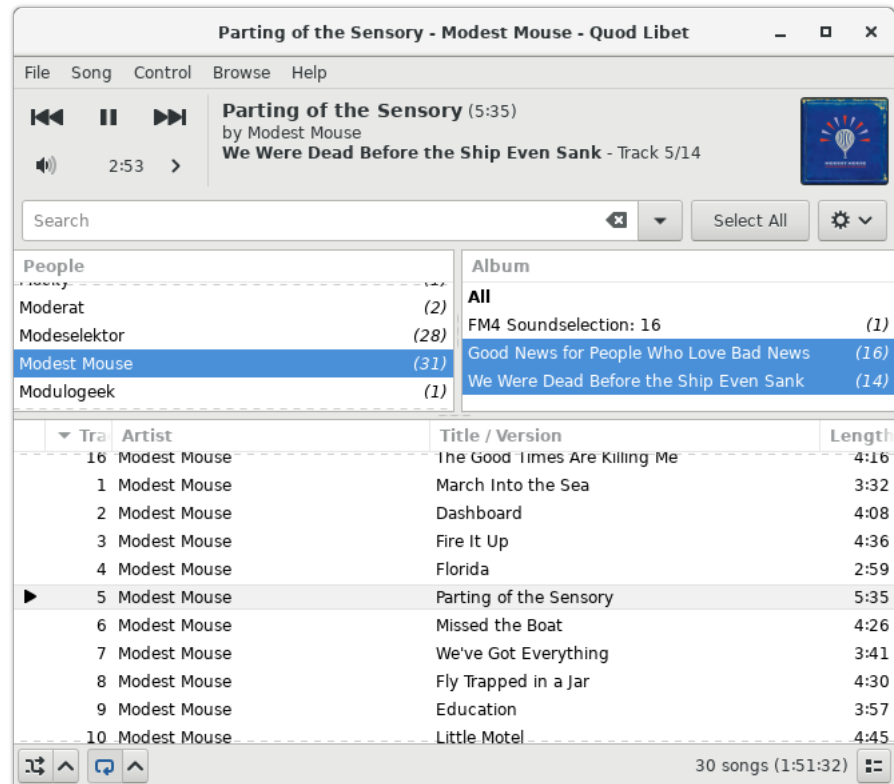
---

## Contents

---

<b>1</b>	<b>Changelog</b>	<b>7</b>
<b>2</b>	<b>Downloads</b>	<b>65</b>
<b>3</b>	<b>Features</b>	<b>69</b>
<b>4</b>	<b>Bug Tracker / Git Repo</b>	<b>73</b>
<b>5</b>	<b>User Guide</b>	<b>75</b>
<b>6</b>	<b>Packaging Guide</b>	<b>125</b>
<b>7</b>	<b>Translation Guide</b>	<b>129</b>
<b>8</b>	<b>Development Guide</b>	<b>133</b>
<b>9</b>	<b>License &amp; Contributors</b>	<b>149</b>
<b>10</b>	<b>Contact</b>	<b>151</b>





**Quod Libet** is a [GTK+](#)-based **audio player** written in [Python](#), using the [Mutagen](#) tagging library. It's designed around the idea that you know how to organize your music better than we do. It lets you make playlists based on [regular expressions](#) (don't worry, regular searches work too). It lets you display and edit any tags you want in the file, for all the file formats it supports.

Unlike some, Quod Libet will scale to libraries with tens of thousands of songs. It also supports most of the [features](#) you'd expect from a modern media player: Unicode support, advanced tag editing, Replay Gain, podcasts & Internet radio, album art support and all major audio formats - see the [screenshots](#).

**Ex Falso** is a program that uses the same **tag editing** back-end as Quod Libet, but isn't connected to an audio player. If you're perfectly happy with your favorite player and just want something that can handle tagging, Ex Falso is for you.

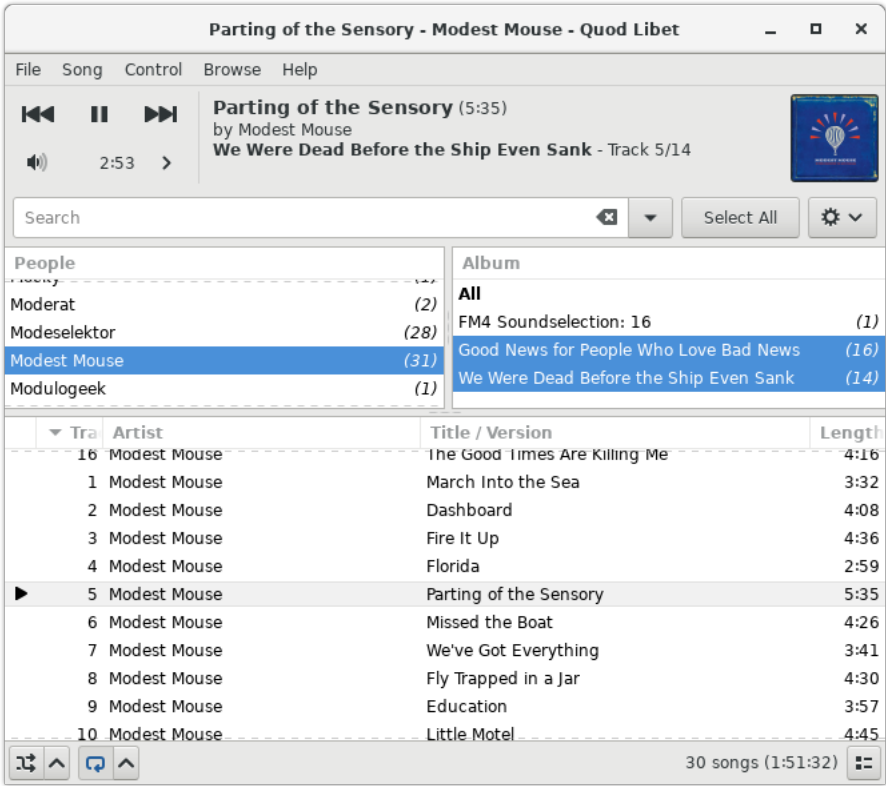


Fig. 1: The paned browser, providing a customizable hierarchy of filters for finding your music

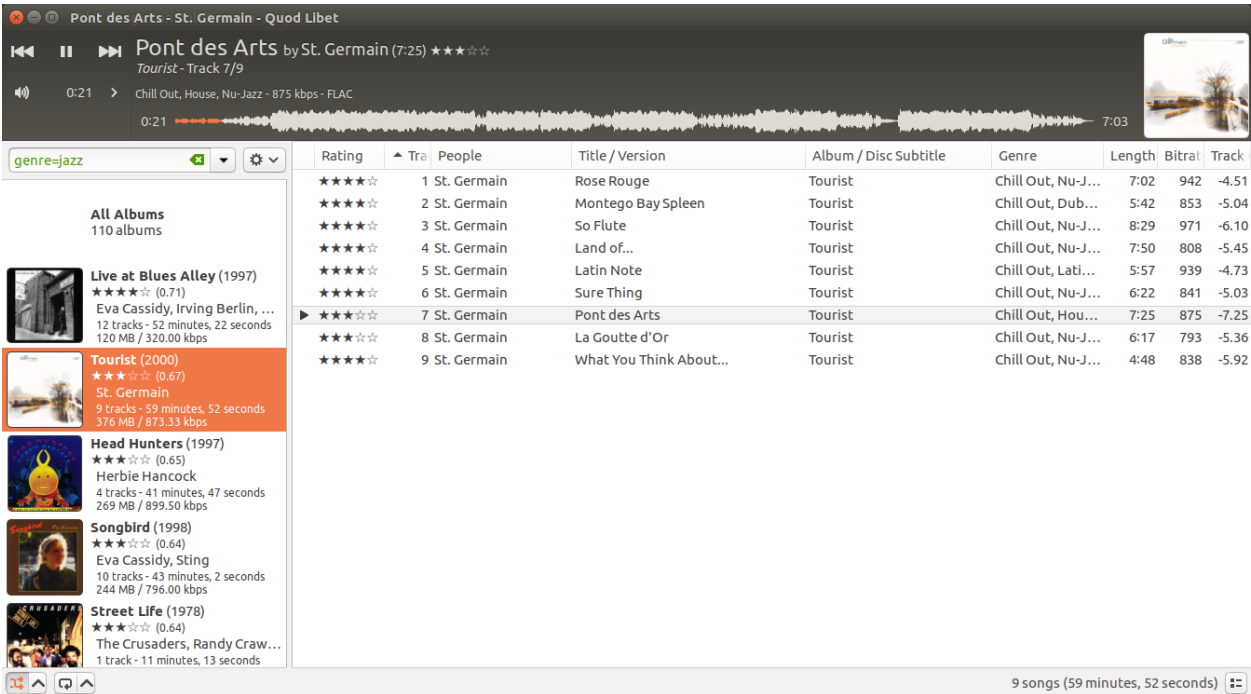



Fig. 2: The album browser, giving a visual anchor for a large library

Bad Apples - Guns N' Roses - Quod Libet


Bad Apples by Guns N' Roses (4:28) ★★★★★  
Use Your Illusion I - Track 14/16

2:37 Hard Rock - 1015 kbps - FLAC


artist=guns n' roses




**Use Your Illusion I** (1...  
★★★★☆ (0.77)  
Guns N' Roses  
16 tracks - 1 hour, 15 minu...  
518 MB / 957.62 kbps




**Use Your Illusion II** (...  
★★★★☆ (0.73)  
Guns N' Roses  
13 tracks - 1 hour, 14 minu...  
509 MB / 958.85 kbps




**Appetite for Destru...**  
★★★★☆ (0.73)  
Guns N' Roses  
12 tracks - 53 minutes, 42 ...  
382 MB / 995.25 kbps




**G N' R Lies** (1988)  
★★★☆☆ (0.63)  
Guns N' Roses  
8 tracks - 33 minutes, 10 s...  
60.8 MB / 256.00 kbps



**The Spaghetti Incid...**  
★★★☆☆ (0.58)  
Guns N' Roses  
3 tracks - 12 minutes, 48 s...  
23.5 MB / 256.00 kbps





Rating	▲ Tra	People	Title / Version	Album / Disc Subtitle	Genre	Length	Bitrat	Track
★★★★☆	2	Guns N' Roses	It's So Easy	Appetite for Destruction	Hard Rock	3:23	990	-5.47
★★★★☆	3	Guns N' Roses	Nightrain	Appetite for Destruction	Hard Rock	4:27	957	-7.90
★★★★☆	4	Guns N' Roses	Out ta Get Me	Appetite for Destruction	Hard Rock	4:22	999	-7.49
★★★★★	5	Guns N' Roses	Mr. Brownstone	Appetite for Destruction	Hard Rock	3:48	1008	-5.41
★★★★☆	6	Guns N' Roses	Paradise City	Appetite for Destruction	Hard Rock	6:47	1000	-6.50
★★★★★	7	Guns N' Roses	My Michelle	Appetite for Destruction	Hard Rock	3:40	1014	-7.03
★★★☆☆	8	Guns N' Roses	Think About You	Appetite for Destruction	Hard Rock	3:51	985	-7.15

12 songs (53 minutes, 42 seconds)

Fig. 3: The cover grid browser, giving an even more visual anchor for libraries

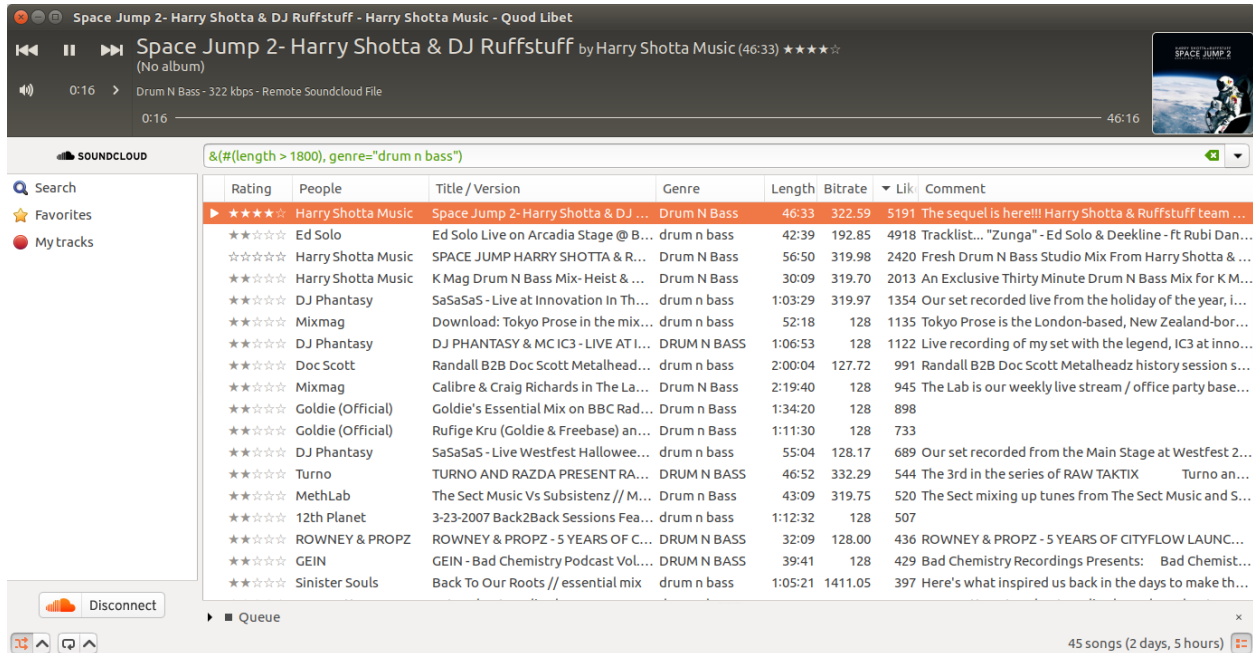


Fig. 4: The Soundcloud browser.

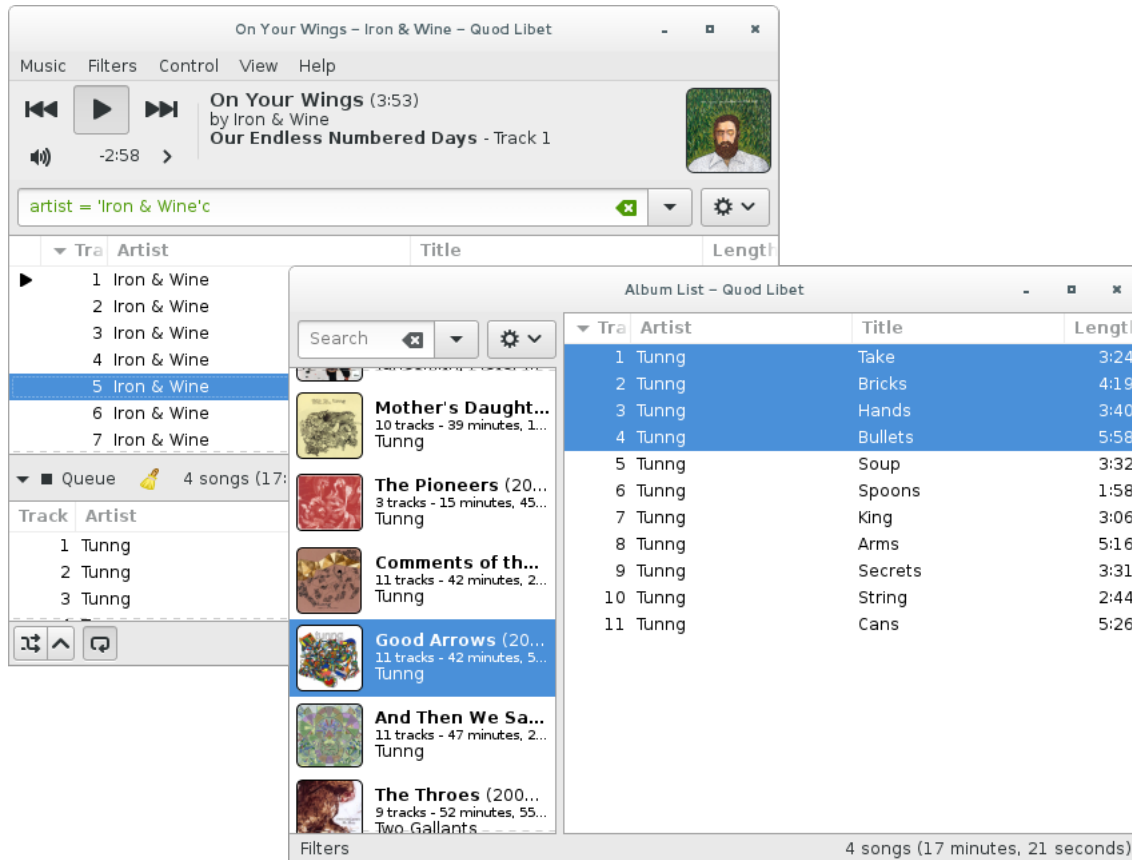


Fig. 5: Quod Libet's queue in action, and its handling of multiple browser windows



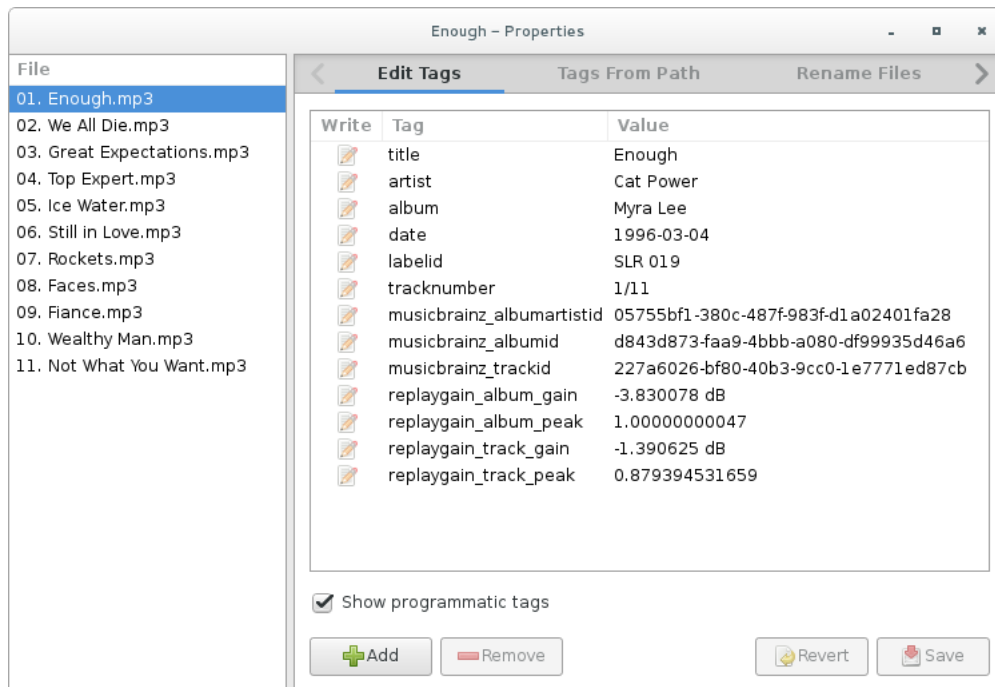


Fig. 6: The powerful and feature-rich tag editor interface, shared by Quod Libet and Ex Falso, that gives you the freedom to tag your music how you see fit

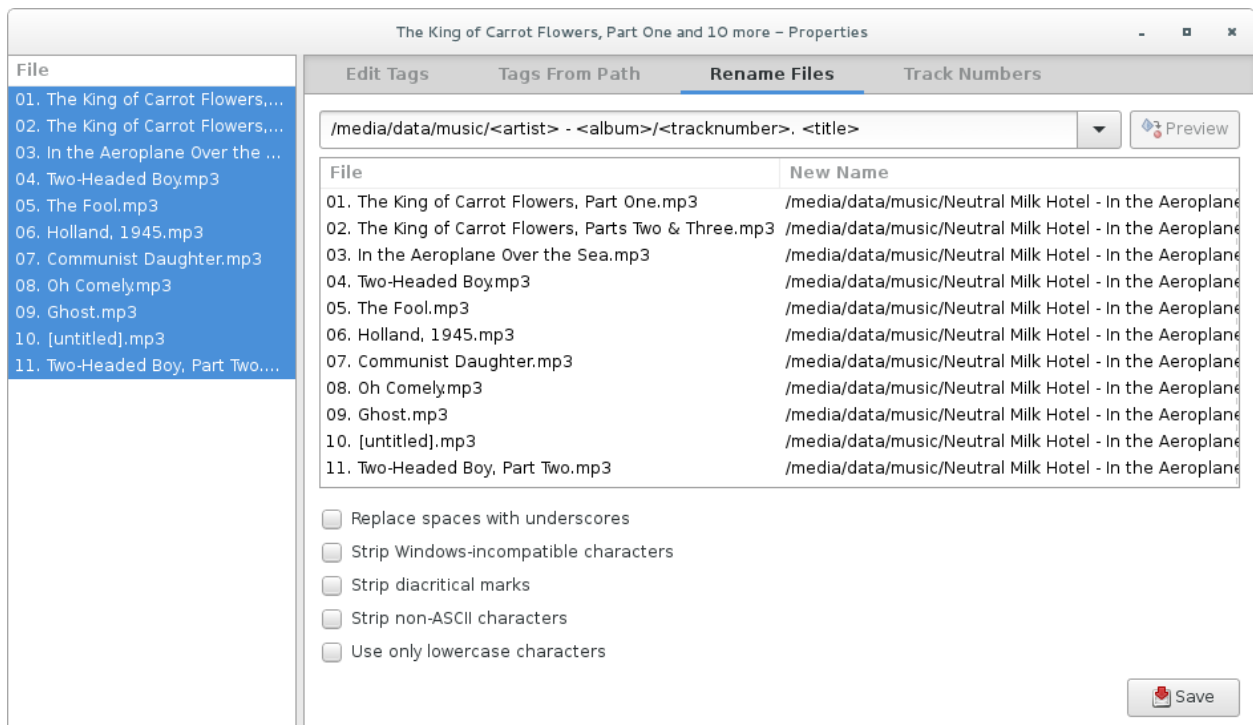


Fig. 7: Quod Libet and Ex Falso also share a powerful file-renaming system

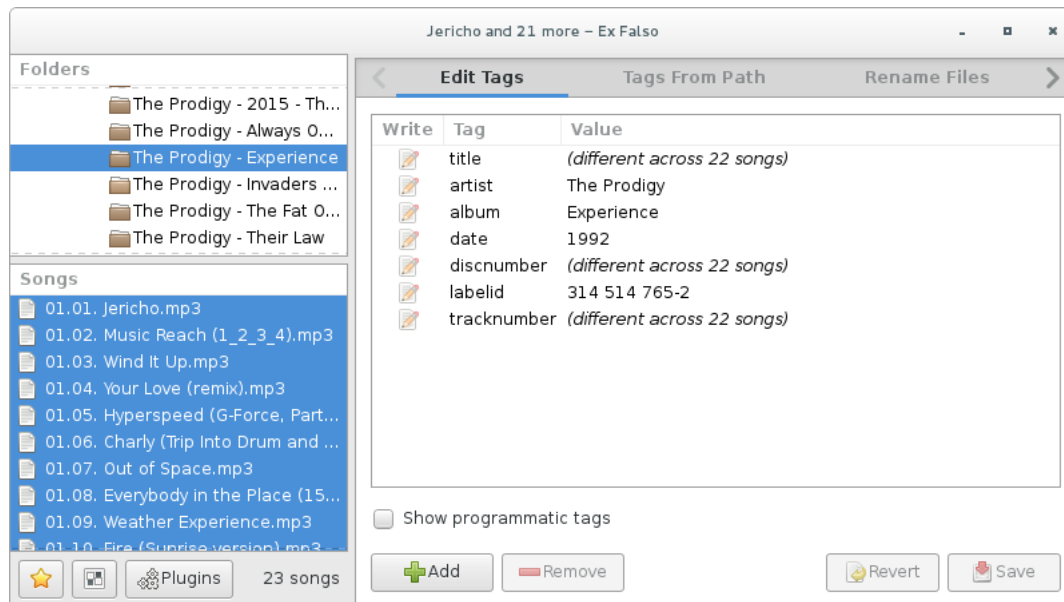


Fig. 8: Ex Falso, the standalone tagging application

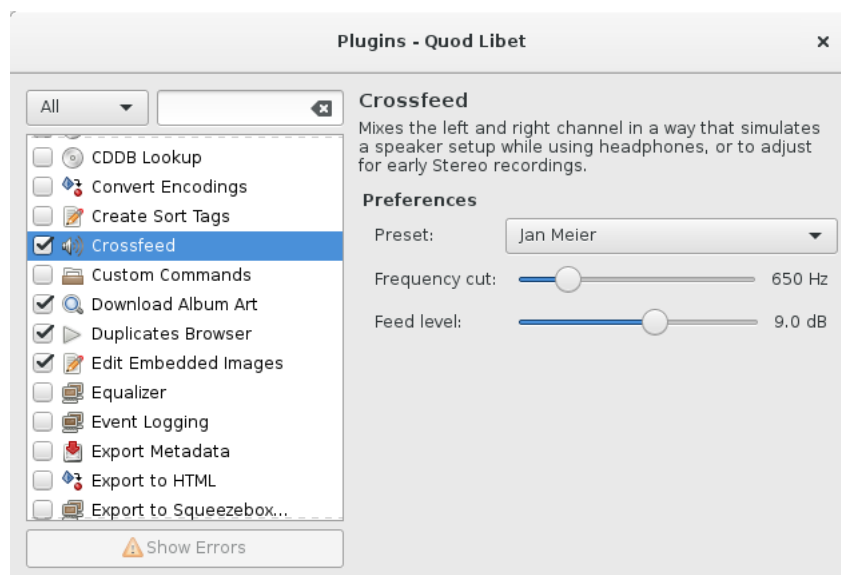


Fig. 9: Quod Libet and Ex Falso are highly extensible through many included plugins

### 1.1 4.3.0 (WIP)

#### Packaging Changes:

- Various installed files have been renamed (“exfalso” -> “io.github.quodlibet.ExFalso”) to work better with Flatpak

### 1.2 4.2.1 (2018-12-26) - DO NOT WEAR THE HAT UNDER ANY CIRCUMSTANCES

#### Translations:

- Hebrew translation update [#3027](#) (Avi Markovitz)
- German translation update [#3036](#) (Till Berger)

#### Fixes:

- Fix freezes when opening the shortcuts window with i3wm [#3055](#) (Fredrik Strupe)
- xinebe: Fix error when pausing a non-local stream [#3057](#)
- Fix mmkeys error (preventing QL to start) when mate-settings-daemon is running outside of a mate session [#3056](#)
- Fix some panels/docks not being able to match the app with the desktop file [#3029](#)
- Migrate metadata plugin fixes [#3070](#) (Fredrik Strupe)

### 1.3 4.2.0 (2018-10-31) - staffed by the living

#### Packaging Changes:

- **gettext 0.19.8** required (was 0.15)
- **intltool** no longer required

### Translations:

- Finnish translation update #2894 (Kristian Laakkonen)
- Russian translation update #2965 (Arkadiy Illarionov)
- Danish translation update #2983 (scootergrisen)
- Polish translation update #3015 (Piotr Drag)

### Codebase:

- Port lots of dbus related code from python-dbus to GDBus #2876 #2879 #2885 #2887 #2886 #2901 #2903 #2904 #2905 #2917 #2918 #2925 (Arkadiy Illarionov)
- Ported from intltool to gettext
- CI: moved from appveyor to azure-pipelines for Windows testing
- CI: Add junit test reporting #2960 (Nick Boulton)
- Various test suite fixes for NixOS #2820 (Jan Tojnar)
- Removed most Python 2 compatibility code #2957
- Add a Pipefile for pipenv #2977
- Various minor Python 3.7 compatibility fixes

### General:

- queue: Add option to keep songs after playing them #2865 (Fredrik Strupe)
- queue: Add option to not play songs from the queue #2865 (Fredrik Strupe)
- Fix non-deterministic ordering of album list and cover grid browsers #2945 (dpitch40)
- multimedia keys: add support for forward/rewind/repeat/shuffle keys #2954 (Druette Ludovic)
- tag editor: Fix a context menu crash #2968 (Eoin O'Neill)
- Remove GNOME app menu #2846
- cli: `--add-location` for adding a file/directory to the library #2970 (Fredrik Strupe)
- cli: Remove deprecated `--song-list` option (Fredrik Strupe)
- Update the big cover window on song changes #2972 (Eoin O'Neill)
- wayland: Fix the application window icon under Plasma #2974
- Various man page updates for missing commands etc #2971 (Fredrik Strupe)
- Add support for importing m3u8 playlists (Fredrik Strupe)
- tags: Don't show generic Performance role description for `~performer:roles` #2994 (zsau)
- themes: Work around redraw issues with the Breeze gtk theme #2997
- tag editor: Remember filelist scroll position on tag save #3005 (Olli Helin)
- Windows: use `SetDllDirectoryW` to prevent loading clashing DLLs not shipped by us #2817
- cover display: Fix the cover window size on multi monitor systems #2915 (Fredrik Strupe)
- session: Add an XSMP session management backend to improve (but not fix) save on shutdown behavior under XFCE #2897

- monkeysaudio: handle missing bits\_per\_sample for older format versions #2882
- Various other fixes and improvements: #2998 (Olli Helin), #2909 (zsau)

**Plugins:**

- waveformseekbar: Mouse scroll seeking #2930 (Peter Strulo)
- waveformseekbar: Add option to hide time labels #2929 (CreamyCookie)
- waveformseekbar: Fix freezes while playing streams #2987 (Olli Helin)
- animocd: Add more preset positions #2937 (Fredrik Strupe)
- New query plugin to match missing tags #2936 (Peter Strulo)
- pitch: Add spin buttons for finer control #2950 (Druette Ludovic)
- wikipedia: Unify Wikipedia plugins #2953 (Fredrik Strupe)
- equalizer: Add option to save custom presets #2995 (Olli Helin)
- mediaserver: Point out required rygel config adjustment in the plugin settings #3004 (Fredrik Strupe)
- custom commands: Fix menu order #2659 (Fredrik Strupe)
- random album: Make it Python 3 compatible #2922 (Stephan Helma)

## 1.4 4.1.0 (2018-06-03) - Trapped in a holodeck that won't take commands

**Packaging Changes:**

- No dependency changes compared to 4.0
- Various installed files have been renamed (“quodlibet” -> “io.github.quodlibet.QuodLibet”) to work better with Flatpak
- We’ve added some new icon resolutions

**Translations:**

- Update Norwegian (bokmål) translation #2833 (Åka Sikrom)
- Update German translation #2860 (Till Berger)
- Update Polish translation #2857 (Piotr Drag)
- Some Russian translation improvements #2670 (Kirill Romanov)
- Various improvements and fixes to make Quod Libet ready for Flatpak/Flathub #2842 #2851 (Felix Krull)
- Show confirmation dialog when removing songs from playlists #2667 (zsau)
- Map bare performer tags to a “Performance” role in ~people:roles #2674 (zsau)
- Add wildcard support to albumart filename preferences #2814 (zsau)
- Fix various typos #2786 (luzpaz) #2750 (Tom McCleery)
- waveformseekbar: Improve hover time indication #2690 (Eyenseo)
- Add shuffle-by-grouping plugin #2788 (archy3)
- Album List - sorting by album average playcount #2794 (Uriel Zajackovski)
- Recognize rating/playcount tags in vorbis comments #2761 (Thomas Leberbauer)

- Handle error when writing empty replaygain tag #2838 (Thomas Leberbauer)
- waveformseekbar: Clamp seek time to valid range #2729 (Peter Simonyi)
- tag editor: don't use inline separators when changing multiple tag values #2684 (Peter F. Patel-Schneider)
- Improve the lyrics file search #2567 (Pete Beardmore)
- Added advanced\_preferences config for scrollbar visibility #2697 (Meriipu)
- cli: Allow floating point arguments for volume #2661 (Jonas Platte)
- code quality: Fix raising NotImplementedError #2772 (Jakub Wilk)
- paned browser: Add shortcut Ctrl-Return to the searchbar #2745 (Felician Nemeth)
- Fix translations on systems with translations spread across multiple directories like with NixOS/Flatpak. #2819
- Fix setting the process name on Linux to "quodlibet" (4.0 regression) #2826
- Fix a case where a deadlocked QL would prevent new instances from being started #2754
- Directory browser: fix not showing Gnome bookmarks
- Various Python 3.7 compatibility fixes
- id3: always ignore TLEN frames #2758
- wayland: fix errors when showing the seek popup widget #2644
- Add cli options for setting repeat and shuffle type #2799 (Fredrik Strupe)
- Queue stop once empty: do check at end of song instead #2801 (Fredrik Strupe)
- searchbar: Don't save indeterminate search queries in the history #2871 (Fredrik Strupe)
- browsers/playlist: Make the side pane take up less space #2765 (Fredrik Strupe)
- Make songs menu key accels work across all browsers #2863 (Fredrik Strupe)
- shuffle: fix shuffle no longer working after one complete run #2864 (Fredrik Strupe)
- tag editor: Allow saving tag if present in all songs but value differ #2686 (Fredrik Strupe)
- iradio: assume http if no protocol specified #2731 (Nick Boulton)
- tag split: Allow non-word characters around tag separators #1088 #2678 (Nick Boulton)
- Various improvements to the cover source plugin system (Nick Boulton)
- gstreamer: Disable gapless for .mod files #2780
- gstreamer: Store the GStreamer registry/cache in our own cache directory to avoid conflicts with the system cache #2839
- macOS: Fix cannot re-order playlist songs with DnD #2867 (David Morris)

### Plugins:

- Add a new cover download plugin using the cover sources #2812 (Nick Boulton)
- headphonemon: fix headphone detection (4.0 regression) #2868
- plugin search: handle search for multiple words better
- importexport: pass a default value for ~#track when sorting. #2694
- equalizer: Fix scales in the preferences not showing the initial values #2722
- randomalbum: Various Python3 fixes #2721 #2726

- trayicon: hide the (useless) scrolling preferences on Windows. #2718
- Move the app/system/dependency info from the about dialog into a plugin.
- Tap BPM plugin: Handle non-numeric BPMs #2824 (Fredrik Strupe)
- plugins: Make random album work on non-album browsers again #2844 (Fredrik Strupe)
- alarm plugin: Port to Python 3 #2735 (Nick Boulton)
- Website search: support ~filename #2762 (Nick Boulton)
- Move Browse Files to core (fully) #2835 #1859 (Nick Boulton)
- qlscrobbler: Fix a potential error when upgrading from 3.9 and older #2768

**Windows:**

- Fix sys.argv not being set by exe launchers (4.0 regression) #2781
- The portable app now uses a local cache directory instead of the system one in more cases.
- Always show the scrollbars like we do on macOS #2717

## 1.5 4.0.2 (2018-01-17) - So it goes!

Bug fixes: #2723 #2721 #2722 #2726 #2717 #2694

## 1.6 4.0.1 (2018-01-13) - Water as far as the eye can see!

Translation updates by Kirill Romanov and Honza Hejzl

Bug fixes: #2677 #2672 #2671 #2680 #2687 #2669 #2699 #2698 #2704 #2703 #2683 #2706 #2705 #2710 #2718 #2719 #2713 #2668 #2715 (CreamyCookie, Nick Boulton, ...)

## 1.7 4.0.0 (2017-12-26) - Speculative fiction where everything's the same, except for one chilling difference

**Packaging Changes:**

- **Python 3.5** required (was 2.7)
  - All Python dependencies need to be switched to their Python 3 variants. In case there is a “py” in the package name it likely needs to be changed.
- **Mutagen 1.34** required (was 1.32)
- **GTK+ 3.18** required (was 3.14)
- **PyGObject 3.18** required (was 3.14)
- **GStreamer 1.8** required (was 1.4)
- **media-player-info** no longer required
- **udisks2** no longer required
- **python-futures** no longer required

- **python-faulthandler** no longer required

### Project:

- Ported from Python 2 to Python 3 [#1580](#) [#2467](#)
- Relicensed all code under “GPLv2 only” to “GPLv2 or later” [#2276](#)

### Translations:

- Update German translation [#2651](#) (Till Berger)
- Update Polish translation [#2646](#) (Piotr Drag)
- Update Norwegian (bokmål) translation [#2506](#) [#2621](#) (Åka Sikrom)
- Russian translation fixes [#2608](#) Kirill Romanov
- Update Finnish translation [#2606](#) Kristian Laakkonen

### Various:

- Allow cover image pop-up to scale up to maximum size [#2634](#) (Peter F. Patel-Schneider)
- Draw a drag handle for the pane separator with newer GTK+ [#2402](#)
- Soundcloud: Add “my tracks” category (Nick Boulton)
- Workaround Ubuntu theme bug which results in drawing artefacts with treeview separators. [#2541](#)
- Added support for custom date column timestamp formats (advanced prefs) [#2366](#) (Meriipu)
- Fix filter function (e.g. max, min) doesn’t work correctly with lastplayed [#2504](#) (Thomas Leberbauer)
- Multimeida keys: make “previous” always go to the previous song [#2494](#)
- Prefer userdir in XDG\_CONFIG\_HOME [#138](#) [#2466](#) (Sauyon Lee)
- Fix error on start under LXDE with its “org.gnome.SessionManager” re-implementation
- Improve visibility of the active state of toggle buttons in the bottom bar [#2430](#)
- Remove device support [#2415](#)
- Filesystem browser: Allow selecting multiple folders [#2399](#) (Nick Boulton)
- CLI: Allow floating point arguments for `--volume` [#2661](#) (Jonas Platte)
- Windows: SIGINT handling support
- Sentry.io error reporting now available on all platforms

### Playback:

- GStreamer: Fix gain adjustments are not applied during the first split second of a song on macOS/Windows. [#1905](#)
- GStreamer: Seeking performance improvements [#2420](#)

### Tagging:

- Add option for moving album art when renaming [#2560](#) (Pete Beardmore)
- Add DSF tagging support [#2491](#)

### Plugins:

- Add support for sidebar plugins [#152](#) (Nick Boulton)
- LyricsWindow: convert to a sidebar plugin [#2553](#) (Nick Boulton)



- Fix synchronized lyrics window not showing #1743 #2492 (elfalem)
- Add more preferences for the album cover search #2511 (Pete Beardmore)
- Waveform seekbar hoover time indication #2419 #2550 (Muges)
- New automatic seekpoint plugin (seeking based on bookmarks) #2437 (Meriipu:)

**CI:**

- Run Windows tests on appveyor #2619
- Submit coverage reports to codecov.io
- Move to circleci for Docker tests #2443
- Dockerize Windows-under-Wine tests #2444

**Various improvements, fixes and Python 3 porting fixes, thanks to:**

- Kristian Laakkonen: #2607 #2605 #2593 #2586 #2578 #2576 #2521
- Emanuele Baldino: #2622
- CreamyCookie: #2574
- Muges: #2425
- Till Berger: #2531 #2530 #2474
- Meriipu: #2486 #2449 #2616
- Fredrik Strupe: #2476

## 1.8 3.9.1 (2017-06-06) - CHECK AND MATE, FAILING BODY AND MIND

- Danish translation update #2394 (scootergrisen)
- Various bug fixes: #2409 #2364 #2406 #2401 #2410 #2414 #2387 #2411 #2386 #2400 #2404 (Nick Boulton et al.)

## 1.9 3.9.0 (2017-05-24) - If you can whistle, you can do this too

**Packaging Changes:**

- **python-zeitgeist** no longer used
- **python-feedparser** required (no longer optional)
- **python-faulthandler** required
- **GTK+ 3.14** required (was 3.10)
- **PyGObject 3.14** required (was 3.12)
- **GStreamer 1.4** required (was 1.2)
- No longer installs icons to “/usr/share/pixmaps”
- Installs more icons into “/usr/share/icons/hicolor/” theme

**Translation Updates:**

- Czech #2175 (Marek Suchánek)

- Danish #2185 (scootergrisen)
- French #2206 (Olivier Humbert)
- Czech #2209 (Honza Hejzl)
- Norwegian Bokmål #2232 #2354 (Åka Sikrom)
- French #2240 (Jean-Michel Pouré)
- German #2388 (Till Berger)
- Polish #2391 (Piotr Drag)

### General:

- Windows: Use native file choosers #2324
- operon: add “-all” option for the “tags” command. #2335
- Queue: Add checkbox to stop after queue is empty #2340 (Fredrik Strupe)
- Opt-in online crash reporting using sentry.io #2313
- Allow resizing of panes in PanedBrowser #2301 (Fredrik Strupe)
- Plugins: Add UI for plugin type filtering #2218 (Nick Boulton)
- Add accelerators for “Open Browser” Menu #2305 (Uriel Zajackowski)
- replaygain: save selected replaygain profiles to config #2279 (Didier Villevalois)
- Allow != in queries #2056 (Nick Boulton)
- Add ~#channels #1686
- songlist: make “space” trigger play/pause. See #1288
- Add --start-hidden and remove visibility restoring from the tray icon #814
- Add non-python crash reporting on the next start #1853
- mp3: include lame preset in ~encoding

### Fixes:

- Fix queue height not getting restored in some cases #2330 (Fredrik Strupe)
- macOS: Fix URL launching from labels #2306
- Windows: Fix crash when the 65001 code page is used #2333
- Windows: Fix crash with French locale in some cases. #2364
- MPRIS: Fix metadata changes not getting emitted #2359 (IBBoard)
- Tray icon: Fix rating menu #2355 (IBBoard)
- Player: Fix “previous” not working with radio streams #2198
- gstbe: increase default buffer duration. #2191
- macOS: Fix meta key for accelerators not working #2271
- Fix error in case stdout gets closed before QL #2205
- Fix icon size of app menu embedded in gnome-shell decoration #2320 #2334 (Vimalan Reddy)

### Plugins:

- Windows: Enable crossfeed plugin

- Add a plugin to export a playlist to a folder #2307 (Didier Villevalois)
- Add skip by rating plugin #2201 (Jason Heard)
- Advanced Prefs: add a configuration for the window title pattern #2272 (Didier Villevalois)
- waveformseekbar: add hidpi detection #2261 (Didier Villevalois)
- waveformseekbar: smoother drawing updates #2289 (Didier Villevalois)
- Add a tap bpm plugin #2264 (Didier Villevalois)
- Add plugin for changing the user interface language #2154
- Add external visualisations plugin #737 (Nick Boulton)
- EQ Plugin: various improvements #1913 (Nick Boulton)
- Add a plugin to toggle the menubar's visibility using "alt" #2263 #2267 (Didier Villevalois)

**Further Contributions:** #2282 (David Pérez Carmona) #2284 (Jakub Wilk) #2294 #2326 (Fredrik Strupe), #2270 #2302 #2280 #2385 (Didier Villevalois) #2308 #2314 (Uriel Zajackowski) #2331 (CreamyCookie)

**Development:**

- tests: use xvfbwrapper if available #2287
- gdist: relicense to modern style MIT
- Use docker on travis-ci #2269 #2290

## 1.10 3.8.1 (2017-01-23) - LET'S TALK ABOUT BIRDS

- GStreamer: increase default buffer duration. #2191
- Fix acoustid plugin #2192
- Fix new playlists from menu #2183 (Nick Boulton)
- mpdserver: Make it work with the M.A.L.P Android client #2179
- Waveform plugin fixes #2195 (Nick Boulton)
- Covergrid context menu fixes #2197 (Joel Ong)

**Translations:**

- Norwegian Bokmål #2194 (Åka Sikrom)
- German #2188 (Till Berger)

## 1.11 3.8.0 (2016-12-29) - Maybe it'll land somewhere cool eventually

**Packaging Changes:**

- `concurrent.futures` required (usually called python-futures, python-concurrent.futures or python2-futures in distros)
- `libgpod4` is no longer used
- Testing now requires `py.test`
- Installs a new zsh completion file

### General:

- Preferences: Add option for changing the duration display format #2021 #1727 #1967 (Nick Boulton)
- Locale-dependent number formatting #2018 (Nick Boulton)
- Fix updates across browsers on changes to ~playlists #2017 (Nick Boulton)
- Don't wake up when idle #2068 #2067
- Covergrid Browser #241 #2071 #2115 #2125 #2110 #2130 (brunob, Joel Ong, qwhex)
- Play order (shuffle / repeat) rewrite to be more modularised / powerful #2043 #2059 #2121 #2123 #2125 (Nick Boulton)
- Improvements / additions to Information window #2119 #1558 (Nick Boulton)
- Search: Fix error when query divides by 0 #2025 (faubi)
- Fix crash on tag edit abort #2081
- Library scan: ignore hidden files #2074
- Remove iPod support
- Log the filename in case something crashes #2143
- MP4: Handle empty trkn/disk #2143
- Library: support autofs mounts #2146
- Various small GTK+/Ubuntu theme related updates
- Fix crash when parsing feeds #2144 (Peter Schwede)
- Song list: ctrl+drag will now force a song copy #1952
- MP4: round bpm to nearest int. #2028 (Nick Boulton)
- Songsmenu icons improved (Nick Boulton)
- Basic zsh completion

### Translation Updates:

- Polish #2141 (Piotr Drag)
- Norwegian Bokmål #2031 #2064 (Åka Sikrom)
- Danish #2169 (scootergrisen)
- Czech #2173 (Marek Suchánek)

### Plugins:

- Discogs Cover Source #2136 (qwhex)
- Add register-date filter to lastfmsync plugin #2127 (qwhex)
- Wikipedia plugin - search instead of direct URL #2112 (urielz)
- Notification plugin: add option to mask 'Next' #2026 #2045 (Corentin Néau)
- Waveform seek bar #2046 (0x1777), related performance improvements (Nick Boulton)
- Add playlists to tray menu #2006 (Nick Boulton)
- Random album plugin fixes #2085 (draxil)
- Custom commands: minor improvements (Nick Boulton)

- Some Auto Library Update plugin fixes #1315 (Nick Boulton)
- Seek bar plugin: invert scrolling directions #2052 (Corentin Néau)

**Windows:**

- Switch to msys2 #1718
- Allow opening audio files with quodlibet.exe #1607
- Enable pitch plugin again #1534
- Windows regressions: crossfeed plugin missing (will be back in the next version)

**macOS:**

- Allow opening audio files with the bundle
- Really (really..) fix TLS #2108 #2107

**Development:**

- Tests: switch to `pytest` as the main test runner
- Tests: `setup.py` quality speedups
- Tests: All tests pass now on Python 3 under Linux and Windows
- All magic builtins gone #2044
- macOS bundle and Windows installer include everything required for running the test suite.

## 1.12 3.7.1 (2016-09-25) - And then you're doomed. Doomed to to have not ill effects, that is!

- tests: Use dbus-daemon instead of dbus-launch for creating a session bus. #2022
- Fix 100% CPU when no song column is expanded. #2030
- Fix SoundCloud login with Ubuntu 14.04 #2034
- MP4: Fix crash when saving certain bpm tags #2028 (Nick Boulton)
- Windows: Make lastfmsync plugin work #1777

## 1.13 3.7.0 (2016-08-27) - Yeah, this is like one of those scammy “name a star” sites!

**Packaging Changes:**

- **Mutagen 1.32** required
- **udisks1** support removed
- **PyGObject 3.12** required
- Add Soundcloud browser #1828 #1990 (Nick Boulton)
- Make F11 toggle fullscreen mode
- Add `--refresh` to the man page. #1914
- Add `--stop-after`. #1909

- Remove support for loading browsers from `~/ .quodlibet/browsers` #1919
- Added shortcut of <Primary>Delete for moving files to trash #1921 (Victoria Hayes)
- gstbe: always use pulsesink if pulseaudio is running. #1926
- Remove udisks1 support
- Add a “Check for Updates” dialog
- Windows: Port mmkeys support from pyhook to ctypes. Fixes accents not working when QL is running. #1168
- OSX: Enable multimedia key handling by default #1817
- Add selection tick (check) for rating(s) that are selected #1891 (Nick Boulton)
- Support composersort #1795 (Nick Boulton)
- Rework application menu #1598 (Nick Boulton)
- Add a ~#playcount equalizer play order plugin #1626 (Ryan Turner)
- Fix too large cover art border radius with Ubuntu themes
- songlist columns: handle font size changes at runtime. #1420
- Fix song list column label fade out in RTL mode
- Fix seek bar getting stuck when releasing the button outside of the widget. #1953
- Add default keyboard shortcuts for browsers/views #1540
- Restore queue state. #1605
- Add a queue toggle button to the status bar and remove the view menu
- docs: Clarified function of the queue #2004 (Bernd Wechner)

### Translations:

- Updated Polish translation #2009 (Piotr Drag)
- French translation update #1932 (Ludovic Druette)
- Fully update British English “translation” (Nick Boulton)

### Tagging:

- AIFF support #1801
- Support musicbrainz\_releasetrackid #1992
- Support musicbrainz\_releasegroupid #1896
- operon: Fix image-set when passing multiple files. #1729
- ASF: add WM/AlbumArtistSortOrder #1936
- MP4: Support saving replaygain tags #1916 (bp0)
- MP4: support replaygain\_reference\_loudness. #1928

### Plugins:

- lyricswindow: Restart WebKit when crashed #1923 (CreamyCookie)
- lyricswindow: Prevent alert windows. #1927 (CreamyCookie)
- tray icon: Improve unity detection #1999
- musicbrainz: Add option to write labelid. #1929

- musicbrainz: Write musicbrainz release track ids #1992
- Rename Force Write plugin to “update tags in files” #1938 (Nick Boulton)
- tray icon: Use App indicator when running under Enlightenment #1941 (Jakob Gahde)
- replaygain: delete tags written by bs1770gain. #1942

**Development:**

- py.test support
- Some Python 3 porting progress: 47% tests passing #1580
- OSX: build dmgs

## 1.14 3.6.2 (2016-05-24) - It seemed like there was a lesson here, but nobody was sure what it was

- Fix queue not expanding with GTK+ 3.20 #1915
- Tag editor: Fix error message for unrooted patterns #1937

## 1.15 3.6.1 (2016-04-05) - GOOD LUCK OUT THERE

- Tray icon: Don't use the app indicator for Ubuntu GNOME and KDE 4. #1904
- Tray icon: Present the window when showing the window through the app indicator menu item. #1904
- Paned browser: Fix crash with numeric tags in patterns #1903
- Paned browser: Fix missing “Unknown” entry for patterns
- OS X: Fix TLS for real
- Lyrics window: Also support webkitgtk2 3.0 (for Ubuntu 14.04)

## 1.16 3.6.0 (2016-03-24) - It is altogether fitting and proper that we should do this

**Packaging Changes:**

- **Mutagen 1.30** required
- **GTK+ 3.10** required
- **PyGObject 3.10** required
- **webkitgtk-3.0** → **webkit2gtk-4.0** (Lyrics Window plugin)
- **sphinx 1.3** required for building the documentation
- New optional plugin dependency: **libappindicator-gtk3** + **typelibs**: for the Tray Icon plugin under Ubuntu Unity and KDE Plasma
- **python-musicbrainzngs** (>= 0.5) instead of **python-musicbrainz2**
- **python-cddb** no longer needed

- **libsoup** ( $\geq 2.44$ ) + **typelib**s required
- Add a keyboard shortcut window. [#1837](#)
- Add `~language`, which shows the language name for iso639 codes
- Allow cross-device moves to trash. [#1782](#) [#1339](#) (Andrew Chadwick)
- CLI: allow backslash-escaped commas in `-enqueue-files`. [#1773](#) (Nick Boulton)
- Fix custom accels read from `~/ .quodlibet/accels` [#1726](#) [#1818](#)
- Fix determination of tag patterns in songlist [#1830](#) (Peter F. Patel-Schneider)
- Fix ratings not being stored if they are the same as the default [#1461](#) [#1846](#)
- ID3: read lyrics from USLT frame, make `~lyrics` read lyrics and form files [#1810](#) (Ivan Shapovalov)
- Make test suite run (and fail) under Python 3 [#1580](#)
- MP4: support conductor, discsubtitle, language & mood [#323](#) (Nick Boulton)
- Paned browser: Allow filters to be reset by clicking heading. [#1284](#)
- Paned browser: use sort tags [#1785](#), [#1796](#) (Peter F. Patel-Schneider)
- Patterns: Allow proper escaping in nested queries. (`<~filename=/^\/bla\/foo/|match|no-match>`)
- Player controls: use a normal button with two icons instead of a toggle button. [#1814](#)
- Playlist browser: implement scroll to playing song [#1426](#)
- Playlist browser: Make display configurable [#1780](#) (Nick Boulton)
- Playlist browser: Improve usability when creating a new playlist [#1839](#) (Nick Boulton)
- Playlist browser: Fix bug when deleting playlists [#1882](#) (Nick Boulton)
- Remove rounded cover preference and make border radius depend on theme. [#1864](#)
- Search: make “ae” match “æ” and “ss” match “ß” etc.
- Search: numeric expressions and query plugins [#1799](#)
- Song info display: show delete option to context menu. [#1869](#)
- Songlist: Highlight the current song. See [#1809](#)
- Support sort tags in song list patterns [#1783](#) (Peter F. Patel-Schneider)
- Various GTK+ 3.20 related fixes

**Translations:**

- Updated Dutch and Norwegian Bokmål translation [#1784](#) (Nathan Follens)
- Updated Polish translation [#1898](#) (Piotr Drag)

**Plugins:**

- Update icons for most plugins: more and (mostly) better chosen. [#1894](#) (Nick Boulton)
- Make songsmenu plugins only enabled if it makes sense for them [#1858](#) (Nick Boulton)
- Remove cddb plugin; it's broken
- Remove Send To plugin, in favour of Browse Folders and k3b plugins.
- New plugin: Pause on headphone unplug. [#1753](#)



- New events plugin: Shows synchronized lyrics from .lrc file with same name as the track [#1723](#)
- Add a seekbar plugin. See [#204](#)
- lyricwiki: port to WebKit2
- tray icon: support Ubuntu Unity and KDE Plasma (using libappindicator) [#1756](#)
- musicbrainz: port to musicbrainzngs. This fixes tagging of multi disc releases. [#829](#)
- Make LyricsWindow an events plugin; Add zoom level preference [#1709](#)
- Add authentication for MPDServer plugin [#1789](#)
- Custom Commands: add support for playlist name. [#1685](#) (Nick Boulton)
- Playlist Export: convert to being a playlist plugin as well as songsmenu. (Nick Boulton)

**OS X:**

- Add a simple dock menu
- TLS support (https streams..) [#1895](#)
- Add option to enable experimental mmkeys support in the advanced prefs plugin. [#1817](#)

**Wayland:**

- Fix seek bar window position (needs gtk+ 3.20)

## 1.17 3.5.3 (2016-01-16) - Uh, I GUESS that'd be good too??

- Fix crash when opening new windows under some DEs (Linux only) [#1788](#)

## 1.18 3.5.2 (2016-01-13) - This is because dates are arbitrary and friendship can be whatever we want it to be!

- Polish translation update (Piotr Drag)
- ID3: don't write albumartistsort twice [#1732](#)
- Use the stream song for --print-playing. [#1742](#)
- Fix background color of some context menus with the Ubuntu 12.04 theme
- Fix adding new tags failing in some cases [#1757](#)
- OSX: make cmd+w close windows [#1715](#)
- Fix a crash with numerics in tag pattern conditionals [#1762](#) (Nick Boulton)
- Fix tests with newer Perl (through intltool)

## 1.19 3.5.1 (2015-10-14) - HOW TO SUCCEED AT SMALLTALK

- Fix volume / mute state resetting on song change with some configurations [#1703](#)
- Fix crash when G\_FILENAME\_ENCODING is set [#1699](#)

## 1.20 3.5.0 (2015-10-07) - BETTER ANSWERS TO “HEY HOW ARE YOU?” THAN “I’M FINE”

### Packaging Changes:

- **Mutagen 1.27** required

### General:

- Add `--print-query-text` to get the current query for browsers that support it [#1634](#)
- Support conditional patterns with QL Query syntax [#1604](#) (Nick Boulton)
- Playlist content search in the playlist browser [#1593](#) (Nick Boulton)
- Disable app menu under Unity [#1599](#)
- Allow users to optionally bypass the trash even if it is available on their operating system (Eric Casteleijn) [#1573](#)
- Try to prevent fifo timeouts for slow operations [#1616](#)
- Fix border drawing with CSD/wayland
- Use float for `~#length` [#1483](#)
- Add a setting to enable/disable rating hotkeys [#1625](#) (Ryan Turner)
- Display all tags in tag list editor not just the non-default ones [#1660](#)
- Add a new `~codec` and `~encoding` tag (library reload needed) [#9](#)
- Add `~bitrate` tag including the unit
- Asymmetric search improvements e.g. ‘o’ now matches ‘ø’
- Various custom column header dialog improvements [#1660](#) (Nick Boulton)
- Prefer `txxx` replaygain over `rva2` [#1587](#)
- Support reading RG when ID3 tag key is in uppercase [#1502](#)

### Playback:

- Implement direct sink volume control (e.g. for `pulsesink`, `directsoundsink`). Changing volume will now control the PA stream volume and result in less delay [#1389](#) [#1512](#)
- Allow muting by middle clicking the volume button. Controls the pulseaudio stream mute property directly.
- Increased GStreamer pipeline buffer size to reduce CPU usage [#1687](#)
- Hide seek slider when not seekable

### OSX:

- Replace “Ctrl” with “Command” in all keyboard shortcuts [#1677](#)
- (already in 3.4.1-v2 build) HIDPI support
- (already in 3.4.1-v2 build) Support for more audio formats

### Plugins:

- Add a plugin for editing `~#playcount` and `~#skipcount`. [#1624](#) (Ryan Turner)
- Advanced preferences plugin [#1050](#) (Bruno Bergot)

- Allow to configure cover size in animosd plugin [#1049](#) (Bruno Bergot)
- Add plugin for removing TLEN frames from ID3 based files. [#1655](#)
- mpd: fix state sync with mpdroid 0.8. [#1636](#)
- Fix screensaver inhibit plugin. [#1692](#)
- qlscrobbler: fix offline mode check box. [#1688](#)
- lyrics window: use mobile wikia version

**Translations:**

- Update Dutch translation [#1618](#) (Nathan Follens)
- Updated greek translation [#1684](#) (Dimitris Papageorgiou)
- setup.py: add a new create\_po command for starting new translations

## 1.21 3.4.1 (2015-05-24) - Apparently, MY problem is a poisonous basement

**Fixes:**

- setup.py: respect --install-data [#1575](#)
- Suppress deprecation warnings with newer glib

**Regressions:**

- Fix error when invoking a plugin with many songs/albums [#1578](#)
- Fix main window sometimes not showing under Ubuntu 12.04
- Fix search not working with non-ASCII text in some cases

## 1.22 3.4.0 (2015-04-09) - She knew every of the things

**Packaging Changes:**

- The main repo moved from Mercurial (Google Code) to Git (GitHub)
- The build should now be reproducible
- **gtk-update-icon-cache** is no longer a build dependency
- **gettext** **>= 0.15** is required now at build time
- A complete **icon theme** is now required (this was also partly the case with 3.3) and an icon theme including symbolic icons is recommended. **adwaita-icon-theme** provides both for example.
- **Mutagen 1.22** required, **Mutagen 1.27** recommended
- New files installed to `/usr/share/icons/hicolor/scalable/apps/`
- **quodlibet.desktop** now contains a **MimeType** entry, which means calling **update-desktop-database** is needed after package installation.
- Improved Gnome 3.16 compatibility
- Fixes for the list tooltips in combination with GTK 3.16 scrollbars

- Include symbolic icons for gnome-shell 3.16
- Album browser: faster cover loading
- Devices: fix detection of Sansa Clip+ with some setups #1523
- Prefs: restore active tab
- Songlist: support patterns in the filter song list menu
- New shortcut `ctrl+shift+j`, like `ctrl+j` but refilters the browser always
- Make build reproducible #1524
- MP4: include codec information in `~format` (needs a library reload) #1505
- GStreamer: fix a deadlock when seeking right at a song change
- Queue: don't decide the next song too early #1496
- Song info widget: provide the full song context menu #1527
- CLI: `--run` to make QL start if it isn't already. Useful for pairing with other commands like `--play-file`. #67
- Add supported mime types to desktop file #67
- CLI: `--play-file` doesn't add songs to the library anymore #1537
- Fix QL starting twice if started in quick succession
- Tooltips: don't span multiple monitors #1554
- MPD-Server: Fix a crash when changing the port number #1559
- Fix short hang on shutdown with GStreamer plugins active #1567
- Fix setting an embedded image in case the file doesn't have tags #1531
- OSX: add a menu bar for Ex Falso
- Fifo: Fix commands failing in case QL is busy #1542
- ...

**Translations:**

- Use `msgctx` for message contexts

## 1.23 3.3.1 (2015-01-10) - Reduce, reuse, recycle, RESUBMIT

**Regressions:**

- Fix occasional errors when closing menus (with the plugin menu in Ex Falso for example) #1515
- Fix operon info #1514
- Fix operon fill error in case a tag doesn't match #1520

**Fixes:**

- Fix HiDPI DnD images when dragging multiple rows

## 1.24 3.3.0 (2014-12-31) - PARALLEL UNIVERSES. Travel there and THEN go back in time, and you can mess things up as much as you want.

### Packaging Changes:

- New optional plugin dependency: **webkitgtk-3.0 + typelibs**
- **Mutagen 1.27** recommended

### General:

- Support `--query` with all browsers that have a search entry. [#1437](#)
- Songlist: Scroll to playing song when replacing the list. [#568](#)
- Songlist: Scroll to first selected song and restore selection for it on re-sort. [#568](#)
- Consider all songs in an album for finding (embedded) album art. [#924](#)
- Support `month` (30 days) in time queries (`#(lastplayed < 1 month)`). [#706](#)
- Support playing a song that is not in the song list. [#1358](#)
- Support numeric date search e.g. `#(2004-01 < date < 2004-05)` [#1455](#)
- Playlists browser: make delete key remove the selected songs from the current playlist [#1481](#) (Nick Boulton)
- File tree: Show XDG desktop/downloads/music folders if available
- File tree: List mountpoints on linux
- Show the filter menu in secondary browser windows (filter shortcuts work there as well now)
- Add `alt+[1-9]` shortcut for notebook widgets to jump to a specific page
- Support loading ADTS/ADIF files (\*.aac). Needs mutagen 1.27.
- Search: New regex modifier "d" which makes all letters match variants with diacritic marks (accents etc.). Enabled by default for normal text searches. `Sigur Ros` will now find songs containing `Sigur Rós`. For regex and exact searches use `/Sigur Ros/d` and `"Sigur Ros"d` to enable. [#133](#)
- New `~people:real` tag which filters out "Various Artists" values (Nick Boulton) [#1034](#)
- Prefer artist over albumartist for single songs in `~people` (Nick Boulton) [#1034](#)

### Fixes:

- Update for theming changes in gtk3.15
- Fix seek slider not working with newer gtk+ and some themes [#1477](#)
- Fix playing song not restoring on start with radio/filesystem browser

### Translations:

- Russian translation update (Anton Shestakov) [#1441](#)
- Updated Greek translation (Dimitris Papageorgiou). [#1491](#)

### Tagger:

- WMA: support multiple values for producer, conductor, artist, lyricist, albumartist, composer, genre and mood (needs mutagen 1.26)

- APEv2: Support reading/writing embedded album art for APEv2 based formats (Wavpack, Musepack, Monkey's Audio)
- Allow removing and renaming from tag names which not all selected formats support.
- Allow toggling of programmatic tags in the tagging UI

### Plugins:

- Various translation related fixes (Anton Shestakov) [#1442](#) [#1445](#)
- New simple lyricwiki plugin using a WebKitGtk webview
- New Rhythmbox import plugin. [#1463](#)
- MPD server: make work again with newer MPDroid (MPDroid crashed on start)
- Trayicon: add option to quit when closing the main window instead of hiding [#619](#)
- Theme switcher: add option to enable/disable client side decorations
- ReplayGain: add option to skip albums with existing ReplayGain values (Nick Boulton) [#1471](#)
- Notifications: Make cover art display work under e19 [#1504](#)

### Operon:

- new 'edit' command for editing tags with a text editor (`VISUAL=vi operon edit song.flac`) [#1084](#)
- new 'fill' command for filling tags using parts of the file path (`operon fill --dry-run "<tracknumber>. <title>" *.flac`)

### OSX:

- Multimedia key support (Eric Le Lay)
- Global menu support / OSX integration. (Eric Le Lay)
- Various fixes / improvements

### Windows:

- Newer mutagen (1.27)
- Newer GTK+/Gstreamer (Tumagonx)
- Fix loading cover art from non-ansi paths
- Starting QL will now focus the first instance if one exists
- quodlibet.exe now passes command arguments to the running instance (`quodlibet.exe -next`) [#635](#)
- New quodlibet-cmd.exe which is the same as quodlibet.exe but can be executed in the Windows console with visible stdout [#635](#)

## 1.25 3.2.2 (2014-10-03) - ENJOY, THERE'S NO GOING BACK

### Fixes

- Fix a crash when seeking streams in some cases [#1450](#)
- Fix a crash in case Windows Explorer favourites link to a non ASCII path [#1464](#)
- Fix playback stopping when playing chained ogg streams [#1454](#)
- Fix context menus not showing sometimes with GTK+3.14.1

**Translations**

- Russian translation update (Anton Shestakov)

## 1.26 3.2.1 (2014-08-16) - BAKE HIM AWAY, TOYS

**Fixes**

- Fix Ex Falso not starting in some cases. [#1448](#)
- Album art download plugin: Fix image file extension (Nick Boulton) [#1435](#)

**Translations**

- Russian translation update (Anton Shestakov) [#1441](#)

## 1.27 3.2.0 (2014-08-01) - WHAT KIND OF GOD MADE IT SO LIONS HUG BACK TOO HARD

**Packaging Changes:**

- **Plugins got merged** into Quod Libet. This means the quodlibet-plugins tarball is gone and plugins will be installed by `setup.py install`. For distros that used to include the plugins in the main package this means all plugin related packaging code can simply be removed. For distros that offered separate packages the installation can be split by packaging `quodlibet/ext` in a separate package. Quod Libet can run without it.
- **UDisks2** is supported, in addition to UDisks1
- **Python 2.7** required instead of 2.6 (might still work, but not tested)

**Tags:**

- `~people` and `~performer` don't show roles anymore, which makes them more useful in the paned browser for example. Instead `~people:roles` and `~performer:roles` will include roles and merge roles like "Artist (Role1, Role2)". Furthermore composer, lyricist, arranger and conductor will be merged with performer roles in `~people:roles`. so "performer:role1=Foo, composer=Foo" will result in "Foo (Role1, Composition)". (qjh)
- `~#rating` in the song list is now a numeric column, `~rating` shows the stars (Jan Path) [#1381](#)

**UI:**

- HiDPI support (start with `GDK_SCALE=2`, needs cairo trunk)
- Various display fixes for GTK+ 3.13 and non-Adwaita based themes
- Seek slider width scales with song length to some extent
- Seek slider shows remaining time
- Play order plugins are now split in random/non-random and the UI was replaced by a toggle button + menu. [#1411](#)
- Removing of songs from a playlist through the context menu (Nick Boulton)
- Song list columns now remember their width/state (qjh)
- Song list columns provide an option to toggle if they expand.

- The multi sort dialog is gone, instead it's now possible to sort by multiple tags by holding down ctrl and clicking on multiple columns.

### Plugins:

- New MPD Server plugin to remote control QL, e.g. through MPDroid #1377
- New acoustid.org fingerprint tagger (basic functionality, but works)
- “Show File” merged into “Browse Folders”, it will now try to select the files if the interfaces allows it.
- Exact rating plugin (Jan Path) #1383

### Player:

- Improved GStreamer error reporting.
- Error recording is gone, since it was just annoying. #1400

### Windows:

- Fix slow startup #1389
- Windows Explorer folder context menu entry for Ex Falso

### Misc:

- Keyboard shortcuts are now documented: <https://quodlibet.readthedocs.io/en/latest/guide/shortcuts.html>

### Developers:

- Due to the inclusion of the plugins into the core, the symlink from ~/.quodlibet/plugins is no longer needed.

### Fixes:

- Fix tray icon crashing or not showing under Gnome Shell 3.12 #1429

### Packaging:

- UDisks2 supported, in addition to UDisks1
- Plugins are now included in the main tarball and will be installed by setup.py, the quodlibet-plugins tarball is gone. (Load path switched from quodlibet/plugins to quodlibet/ext for system wide plugins, loading from ~/.quodlibet/plugins is the same) #1396
- For BSDs: setup.py has a new “--mandir” to select the man page location
- Packaging guide: <https://quodlibet.readthedocs.io/en/latest/packaging.html>

## 1.28 3.1.91 [beta] (2014-07-22) - Pumps, powerheads, lights and filters!

See *final release*

## 1.29 3.1.2 (2014-06-20) - Dang it

- Fix 3.1.1 regression causing folders in the file browser to show up in reverse order #1390



## 1.30 3.1.1 (2014-04-28) - I've shown that you're dealing with an Alpha here, baby, not some weak Beta!

- Fix a crash with GTK+ 3.12 #1384
- Handle invalid flac picture blocks #1385
- Fix “setup.py install --record” #1373

## 1.31 3.1.0 (2014-04-10) - Olden times, man! NEVER LIVE THERE.

- Windows is supported again. And it should be in better shape than with 2.6 in many aspects. Embedded images work now, newer GStreamer with more codecs, operon is included etc. The file browser and EF now show the favorite folders from the Windows Explorer. The installer will now uninstall any existing installation first and as with 2.6.3 there is a portable version available.

There is still an unsolved problem regarding miss-placed context menus with multiple monitors #1319.

Thanks goes to Bakhtiar Hasmanan for providing a working PyGObject stack.

- Initial Wayland support is here (only tested under weston). This was mostly fixing weird usage of GTK+ that just happened to work under X11 and not using the screen size for calculations since there is no real screen under Wayland.
- Piotr Drag, Rüdiger Arp, Diego Beraldin and Dimitris Papageorgiou worked on improving the translations.
- Nick Boulton worked on a plugin system for playlist plugins and moved the duplication/shuffle actions to it. He also moved the rating configuration from the plugin into the core.
- Simonas Kazlauskas worked on a plugin system for cover art sources currently supporting last.fm and musicbrainz (exposed as two plugins). If active it will fetch covers in case no local cover is found. In the future we might implement the album art downloader on top of that.
- Thomas Vogt made transparency work again with GTK+3 in the OSD plugin. (fake transparency now also works again, which was the last known regression from the PyGObject port)
- operon gained new commands (image-extract, image-set, image-clear) for manipulating and extracting embedded images for all formats supporting embedded images in QL (id3, ogg, flac, wma, mp4). See the manpage [0] for examples. There is also a QL plugin which allows removing all embedded images and embed the active one. This should get better integrated into the tag editor at some point.
- Display patterns now support specifying the markup using square brackets to not need escaping in the common case. “<b><artist></b>” can now be written as “[b]<artist>[/b]” (the old way still works).
- In the radio browser the radio list now contains icecast and shoutcast2 stations in addition to shoutcast1 ones and only one additional mirror is included for each station. QL now shows ~4100 stations of ~30000 we know about. Use “Update stations” to get the new list.

Other changes:

- QL now remembers additional open browsers and reopens them on start.
- The main tool bar is better integrated with GTK+ themes.
- We use symbolic icons in many places.
- Added a simple GNOME app menu.
- ‘albumartist’ is now used for album identification.
- <shift>space enables “stop after the current song”.

- Warning before opening too many plugin windows (Nick Boulton) #1231
- New `-unqueue-all` command #1234

Fixes:

- Config gets saved atomically and handle a corrupted one #1042
- `editing:id3encoding` option was ignored #1290
- album browser: Fix sorting by rating #1352
- search: Fix results for “&(foo, !bar)” #1327
- Various crashes caused by code not being ported to PyGObject properly.

Dependencies & Packaging:

- No dependency changes compared to 3.0
- We now install `appdata.xml` files
- We now install a `dbus` service file
- `setup.py build_sphinx` builds the html user guide

## 1.32 3.0.91 [beta] (2014-02-28) - You'd have to be in space for that to work.

See *final release*

## 1.33 2.6.3 (2013-09-25) - The one that can't even go naked into space without dying!

This is a Windows only bug fix release.

**Windows:**

- Fix library saving [1230] (Sebastian Thürschmidt)

## 1.34 3.0.2 (2013-07-22) - LATER, THE OCEANS BOIL AS THE EARTH TURNS TO FIRE

**General**

- Device backend: Correctly detect `udisks1` if it isn't running [1200]
- `mmkeys`: Really make `libkeybinder` work again [1203] (Hans Scholze)
- Fix play button not starting playback if no song is active [863]
- Don't forget newly created bookmarks in some cases
- Fix “Refresh” in the file tree browser [1201]
- Fix menu separator drawing with PyGObject 3.2 (Ubuntu 12.04)
- Various fixes

**Plugins**

- Forward library events to event plugins again
- Fix bookmarks plugin, playlist export plugin, HTML export plugin
- animosd: Handle multiple monitors (Nick Boulton)

## 1.35 2.6.2 (2013-07-22) - 256 Pictures of Cool Bugs

2.6.1 skipped to keep in sync with the 3.0 branch.

**General**

- Device backend: Correctly detect udisks1 if it isn't running [1200]
- Fix play button not starting playback if no song is active [863]
- Don't forget newly created bookmarks in some cases
- Various fixes

**Plugins**

- Fix HTML export plugin
- Fix Bookmarks plugin

## 1.36 3.0.1 (2013-07-08) - *gasp*

- Fix a crasher with some PyGObject versions [1211]

## 1.37 3.0.0 (2013-06-15) - THE NEMESIS HYPOTHESIS

**Requirements & Packaging Changes**

- Python 2.6+
- GTK+ 3.2+ & GIR (instead of 2.x)
- GStreamer 1.0+ & GIR (instead of 0.10)
- PyGObject 3.2+ (3.4 highly recommended) (instead of PyGTK)
- PyGObject cairo support
- libgpod 0.5+ (instead of python-gpod)
- libkeybinder-3.0 & GIR (instead of python-keybinder)
- HAL support removed
- New *operon* script + man page
- New .ini file for registering QL as a GNOME Shell Search Provider

**Translations**

- New: Czech translation (Honza Hejzl)
- Russian translation update (Anton Shestakov)

- Lithuanian translation update [1079] (Naglis Jonaitis)
- Swedish translation update [1117] (Daniel Nyberg)
- Spanish, Basque and Galician translation update (Johám-Luís Miguéns Vila)
- Greek translation update [1175] (Dimitris Papageorgiou)

### General

- Improved rating visibility [1070] (Nick Boulton)
- File system view: DnD directories to external targets (Nick Boulton)
- Support GNOME Notification Sources
- Bayesian averaging for set (album) ratings [1085] (Nick Boulton)
- New command line tagger: operon (see *man operon*)
- Hide songs on not-mounted drives on start without library refresh [984]
- Preferences UI for managing masked mount points [984]
- Support all patterns as song list headers [507, 1121] (Nick Boulton)
- Save/restore queue position
- Documentation is now Sphinx/reST based and hosted on readthedocs.org

### Fixes

- Fix unwanted re-filtering of all open browsers if the search history changes
- Fix crash when re-adding devices while QL is running [1120]
- Remove EF directory mime type again (too many problems with file managers)

### Tagging

- APEv2: Add disc<->discnumber mapping

### Plugins

- New: Custom Commands plugin (Nick Boulton)
- New: GNOME Shell Search Provider plugin [1147]
- ReplayGain plugin: Parallel processing [807]
- CD burn plugin: Add Xfburn support [1173]

### Known Regressions

- GStreamer output device selection is no longer supported. (GStreamer 1.0 has removed the property probing interface)
- No multimedia keys support with non-GNOME DEs in some distributions due to missing packaging:
  - [https://bugzilla.redhat.com/show\\_bug.cgi?id=963476](https://bugzilla.redhat.com/show_bug.cgi?id=963476)
  - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=710254>
- Animated On-Screen Display plugin is missing transparency support
- No Windows build (work in progress)

## 1.38 2.6.0 (2013-06-15) - Someone's attacking us from... space?

2.6 is the last PyGTK/GTK+2 based release of Quod Libet / Ex Falso. It contains most of the changes that went into 3.0 and will only receive bugfix releases from here on out.

### Requirements & Packaging

- Python 2.6+
- PyGTK 2.16+

### Everything else

- See 3.0.0 NEWS with a few exceptions

## 1.39 2.9.92 [beta] (2013-06-05) - alternatetimetinelmetarhyme

### General

- Fix the main song list resetting while working with multiple browsers
- Fix various widget redraw/positioning/jumping problems
- Various fixes regarding GTK+3.6+ (seek slider, tv hints)
- Nicer about dialog
- Fix one-click ratings [1170]
- Fix various crashes with PyGObject3.2 [1172]

### Plugins

- New GNOME Shell Search Provider plugin [1147]
- Support Xfburn in the CD burn plugin [1173]
- viewlyrics plugin: Fix key handling [1171]
- Fix musicbrainz plugin [1162]
- Fix replaygain plugin in Ex Falso [1163]
- Fix fingerprint plugin [1174]

## 1.40 2.9.91 [beta] (2013-05-13) - welcome to a place where incredibly terrible things can happen to you and your friends for no reason!

### General

- Spanish, Basque and Galician translation updates (Johám-Luís Miguéns Vila)
- Tag editor: Fix context menu not showing
- Album collection: Fix crash with PyGObject 3.2
- Fix search bar text color
- Fix DnD to closed queue
- Fix hangs during unix signal handling

- Fix 100% CPU in some cases
- Fix library refresh pause/stop handling
- Some speed improvements and fixes

#### Plugins

- ReplayGain plugin now processes albums in parallel
- New dark theme option in the theme switcher plugin
- Fix GStreamer equalizer
- Fix theme switcher plugin
- Fix GStreamer mono plugin

### 1.41 2.9.82 [alpha] (2013-05-02) - One day Marty McFly got bit by a werewolf!

#### PyGObject/Gtk+3.0/Gstreamer1.0 Port - Alpha 2 release:

- Fix lyricsview plugin (Nick Boulton)
- Fix replaygain album gain/peak writing.
- Fix crash on one-click ratings in the song list.
- Fix crash when playing a song while editing its tags.
- gstbe: Increase operation timeouts (for spinning up disks etc.)

### 1.42 2.9.81 [alpha] (2013-04-27) - Because my hypothesis is: it's rad

#### PyGObject/Gtk+3.0/Gstreamer1.0 Port - Alpha 1 release

### 1.43 2.5.1 (2013-04-23) - Yes: MY COMPUTER IS A PERSON.

- Fix various widgets not showing with pygtk/pygobject trunk
- Fix QL refusing to start in some cases [1131]
- Improve web browser discovery and fix on Windows
- Fix various problems with playlists + masked files [1095]
- Reduce debug output if loading a file fails [1080]
- Plugins:
  - notify: Don't set notifications to transient [1103]
  - lastfmsync: Fix loading/saving of cache [1093]

## 1.44 2.5 (2012-12-19) - Reading on the floor: literature!

- Greek translation update (Dimitris Papageorgiou) [1064]
- Russian translation update (Anton Shestakov) [1072]
- Lithuanian translation update (Naglis Jonaitis)[1079]

## 1.45 2.4.91 [beta] (2012-11-23) - hello and thank you for installing an internet!

### News for packagers

- setup.py will install png and svg icons into hicolor and a png icon into pixmaps. It will also try to call gtk-update-icon-cache if it's in the target prefix/root (make sure the icon cache gets updated on package install)
- C extensions removed. QL is now Python only.
- PyGTK 2.16+
- Python 2.6+
- Support for libudev.so.1 ( $\geq 183$ )
- New: python-keybinder needed for multimedia keys
- New plugin directory: gstreamer
- Optional plugin dependencies:
  - Removed: python-indicate (Ubuntu sound menu), lastfmsubmitd (old last.fm plugin)
  - New: rygel (DBus UPnP MediaServer), python-zeitgeist (Zeitgeist)

### News for translators

- QUODLIBET\_TEST\_TRANS=xxx will append/prepend “xxx” to all translations so you see what is translatable and for devs to check how, long translations will affect the UI.
- “setup.py po\_stats” to see how much is translated for each po file.
- “setup.py check\_pot” to see if a file containing translatable strings is missing from POTFILES.in

### General

- Search: Handle non-ascii values for filesystem tags (~filename, ~dirname..) [227]
- New internal tags: ~originalyear ~#originalyear. [966]
- New internal tag: ~playlists (Nick Boulton)
- Shortcut: alt + left/right -> seek +/- 10 seconds [981]
- Support startup notification spec
- Support newest thumbnail spec (v0.8.0)
- Basic Unity quicklist
- New --stop switch to stop and release the audio device [1018]
- List tooltips: work with gnome shell, never shift left, support trees [778]
- New --no-plugins switch to start without plugins

- No wakeups if nothing is playing.
- Directory mimetype for Ex Falso
- Shortcut: ctrl+left/right, left/right for treeview navigation
- Some UI cleanup, less padding in the main window
- Remember window size & position for properties, info, browsers [106]
- Device selection for the gstreamer back-end
- Use lyricist for finding lyrics if available
- Click on default cover icon launches album art plugin [2] (Nick Boulton)
- Fix: Work with Compiz window placement plugin [871]
- Fix: Queue widgets not clickable in some cases
- Fix: Double-click on album will play song from queue [231]
- Fix: Filter on album in album browser now uses the album key
- New: PluginConfigMixin added to core to simplify plugin prefs (Nick Boulton)
- Fix: -status (carlo.teubner) [1045]

### Formats

- New: Monkey's Audio
- New: Ogg Opus (needs mutagen 0.21) [1012]
- New: MIDI
- Basic SPC tag parsing [282] (David Schneider)
- Add m4v to valid mp4 extensions

### Tagging

- Limit path sections to 255 chars instead of tags to 100 [915]
- ID3: Write foobar style replaygain tags. [1027]
- VORBISCOMMENT: Write totaltracks/totaldiscs [929] (Michael Ball)
- Shortcut: ctrl-s for saving changes, and configurable standard accels per locale [697] (alex.geoffrey.smith, Nick Boulton)
- Updates to tag splitting (originalartist, performer) (Nick Boulton)

### Translations

- New: Greek translation (Dimitris Papageorgiou)
- German translation update (Rüdiger Arp)
- British English translation update (Nick Boulton)
- French translation partial update (Nick Boulton)

### Plugins

- Removed: lastfmsubmitd
- Removed: Dbus mmkeys (moved to core)
- Removed: Ubuntu Sound Menu (no longer needed)



- New: UPnP AV Media Server plugin (needs rygel)
- New: ViewLyrics plugin (Vasiliy Faronov)
- New: Filter on directory [922]
- New: Zeitgeist plugin [717]
- New: Mac OS X mmkeys plugin (Martijn Pieters) [967]
- New: Telepathy/Empathy status plugin [478] (Nick Boulton)
- New GStreamer plugins: Compressor, Crossfeed, Pitch, Mono
- New: Filter on multiple tags [1014]
- New: Squeezebox Playlist Export (Nick Boulton)
- Browse Folders: Use the default file browser [983]
- Equalizer: add presets
- MPRIS: various fixes (for the GS plugin)
- Notify: Dismiss notifications after some time
- Duplicate Browser: expand/unexpand all button
- CD burn: now menu-based
- Updated: Auto Library Updater (Nick Boulton)

#### Browsers

- New: Album Collection - provides a tree-like view of albums similar to Paned.
- Playlists:
  - Delete shortcut [942] (Johannes Marbach)
  - Shuffle playlist (Nick Boulton)
  - Remove duplicates [685] (Nick Boulton)
- Album list: more filters

#### Windows

- Make Browse folder plugin work [993]
- Multimedia keys support

## 1.46 2.4.1 (2012-07-27) - Man! If I were a robot, a lot of things would be different

- Fixes:
  - Fix skipping one song during a song change [987] (This also broke the random album plugin in some cases)
  - Windows: Crash in file system view if 'My Music' folder is missing [1008]
  - Fix -quit [958]
  - Fix playing of files that don't match the file system encoding [989]
  - Workaround for mutagen ID3v2.3 update bug [mutagen 97]

- Various fixes [1013, 1002, 962]
- Plugin fixes:
  - lastfmsync crashes [957]
  - Various Duplicate browser fixes [999, 954]
  - Notification crash [975]

## 1.47 2.4 (2012-03-18) - He decides he must become... Abe Atman!

- Fixes:
  - Support xinelib 1.2 [904]
  - MP3/ID3: fix some rare crashes; prefer main embedded cover
  - Vorbiscomment:
    - \* Ignore coverart and use it as cover art source [910]
    - \* Fix deletion of metadata\_block\_picture
  - album art: update coverparadise, disable discogs (API changed)
  - squeezebox fixes (Nick Boulthbee)
  - German translation update (Rüdiger Arp)
  - Various fixes [890, 909, 899]
- Fixed regressions:
  - Python 2.5 / PyGTK 2.12 compatibility
  - Restore saved play order [891]

## 1.48 2.3.91/92 [beta] (2012-01-16) - Players only love you when they're playing

- Fixes:
  - Don't remove periods from tag values in patterns [368]
  - Don't jump to playing song on stream changes
  - Fix wrong path encoding in the exfalse file selector under Windows
  - Fix error when controlling playback during startup [810]
  - Handle invalid header patterns
  - Don't lose the radio library randomly [645]
  - Handle non utf-8 and invalid filenames in the song list [798]
  - Fix a crash when the song list changed during a gapless transition [799]
  - Tray icon doesn't appear in KDE panel [881]
  - xine backend: Fix equalizer value range
- Plugin Fixes:

- Fix “&” in notifications (xfce4-notifyd) (Anton Shestakov)
- Fix animosd config again
- Fix Amazon cover search (API change)
- General
  - Improved startup speed
  - GNOME session shutdown fixes
  - Hide all windows on shutdown
  - Handle signals during startup
  - Correctly push signals into the gtk+ mainloop (no more segfaults)
  - Cyclic saving of all libraries not just the main one
  - Rename the process to “quodlibet” or “exfalso” [736] (Nick Boulton)
  - Queue keyboard shortcut is now Ctrl+Return not just Q [747]
  - Add new songs that match the active filter to the song list
  - Focus search entry with ctrl+l [666]
  - Fix reverse sort (sort by album first)
  - Custom sort dialog [820] (Philipp Müller, Joschka Fischer)
  - Make the paned browser the default one
  - Focus the first row in all automatic list selections [835]
  - Select next song in the song list after song removal [785]
  - Speed up song removal in long lists [785]
  - Delete keyboard shortcut in the queue
  - Add menu entry to rate current playing song (Nick Boulton)
  - Make it possible to override quodlibet/exfalso icon, by placing an icon in the current icon theme [614]
  - Close buttons in all dialogs (since GNOME 3 has no close button in the window decorations for dialogs by default) [772]
  - Make main window resizable with only the queue showing [657] (Florian Demmer)
  - Make the paned browser prefs resizable
  - Search bar: No delays for any keyboard/mouse actions except typing
  - Estimate FLAC bitrate using the filesize [342] (Nick Boulton)
  - New ~#filesize tag: requires library reload (Nick Boulton)
  - Enhancements to the ways album art is detected (Nick Boulton): new tab in prefs, new option for forcing art filename [328], new option for preferring embedded image over external ones
  - Allow numeric ~#replaygain\_XXX tags (Nick Boulton)
  - album browser: Restore search on start
  - album browser: Move sort order in the preference button sub menu
  - album browser: Load all visible covers before showing the album list
  - album browser: Fewer redraws after filtering

- album browser: Add sort by genre option [340]
- radio browser: Genre filter list
- radio browser: Use the default search bar (with history)
- radio browser: Remote station list with 4000 stations
- radio browser: Properly sync the song list play icon with song changes
- radio browser: Prefill new station dialog with last URL in the clipboard
- radio browser: title falls back to organization and artist to website (for the song list)
- radio browser: Buffer process shown in status bar
- search: Stricter numeric value parsing (only allow valid units)
- search: Don't require a space between number and unit: #(added<1day)
- search: Support GB/KB/MB/B units for ~#filesize
- Gstreamer:
  - Fully support playbin1 again (QUODLIBET\_PLAYBIN1 env var)
  - Allow setting of stream buffer duration [696]
  - Sync replaygain volume change with track change [652] (use track-changed signal in newer gstreamer) this needed the removal of the 500ms queue. Can be enabled if there are problems with gapless (QUODLIBET\_GSTBE\_QUEUE env var)
  - Don't add equalizer if the plugin is disabled: No unnecessary conversions to float (flac, mp3 decoder), less CPU
  - Don't use the fluendo mp3 decoder if mad is available, less CPU
  - No video decoding/playing (mp4 files for example)
  - Properly emit song-started/ended for radio stream songs (so they get counted as auto started by the new notify plugin)
  - Add button in the prefs to print the currently used pipeline including format conversions (only in -debug mode)
  - No more jumping of the position slider during song changes
  - Better parsing of stream metadata [750]
- Translations:
  - Russian translation update (Anton Shestakov)
  - German translation update (Rüdiger Arp)
  - Italian translation update (Luca Baraldi)
- Tagging/Ex Falso:
  - Improve support for language tag, with ISO 639-2 suggestions (Nick Boulton)
  - ID3: handle TLAN [439] (Nick Boulton)
  - Ignore zero TLEN id3 frame [222]
  - Allow performer to be split out of title (Nick Boulton)
  - Ogg Theora support
  - Ex Falso about dialog

- CLI:
  - `--debug` (colored output, yay)
  - `--enqueue-files=file,file` [716] (Nick Boulton)
  - `--print-query=query` [716] (Nick Boulton)
  - `--force-previous` [441] (go to previous not depending on the current position)
- Plugins:
  - Removed old plugin import fallback code: In case loading a third party plugin fails, set the `QUODLIBET_OLDIMPORT` env var.
  - Show an error message instead of the stack trace for common plugin loading errors (import errors)
  - Improved notification plugin [588] (Felix Krull)
  - Improved scrobbler preferences with account data verification
  - Trayicon: Use custom theme icons [614] Prevent the main window from showing on startup
  - Musicbrainz: Only write sort tags that are different
  - Titlecase: New prefs switch to allow all caps in tags (Nick Boulton)
  - NEW: Website Search (Nick Boulton)
  - NEW: Inhibit Screensaver while playing (GNOME)
  - NEW: Pause while the screensaver is active (GNOME)
  - NEW: Acoustid.org fingerprint plugin (only for submitting atm)
  - NEW: Duplicates browser (Nick Boulton)
  - NEW: Mute radio advertisements (di.fm only atm)
  - NEW: Watch directories for file changes (using pyinotify) [270] (Nick Boulton, Joe Hight)
  - NEW: Theme switcher plugin
  - NEW: Squeezebox plugin (Nick Boulton)

## 1.49 2.3.2 (2011-10-17) - It doesn't matter! My beats are great!

- Fix crash in album browser [781]
- Plugins:
  - Dbus multimedia keys: Make it work with `gnome-settings-daemon 3.x`
  - Album art: Remove darktown, fix discogs.
  - MPRIS: Various fixes [776, 817, 827]
- Translation Updates:
  - Lithuanian (Naglis Jonaitis)

## 1.50 2.3.1 (2011-07-16) - YES It works in BOTH temporal directions

- Absolute path renaming on Windows [506]
- Fix dynamic song removal of songs not matching the query [713]
- Fix “-print-playing <~#rating>” [730]
- Fix search not working with an active pattern column
- Fix hang with newer GStreamer versions and sinkless pipelines
- Some minor fixes [682, 724, 704]
- Plugins:
  - Fix MPRIS not handling invalid dates (Nick Boulton)
  - Some OSD fixes (Nick Boulton)
- Translations:
  - German (Rüdiger Arp)
  - Italian (Luca Baraldi)
  - Lithuanian (Naglis Jonaitis)

## 1.51 2.3 (2011-04-01) - I THOUGHT THAT WAS PRETTY CLEAR

- Various minor bug fixes
- Some small translation updates (Anton Shestakov)
- Update of the 2.2.99 news entry

## 1.52 2.2.99 [beta] (2011-03-13) - I can imagine that one day there could be a person who would read that

- Quod Libet now needs Python >=2.5
- Drag and drop in paned browser
- Speed up adding many songs to the queue
- Smaller volume, seek controls
- Ask for playlist name on creation
- Output playing progress when using -status
- Use current icon theme icons everywhere (for DAPs etc.)
- Floating point custom tags
- Audio streaming fixes (buffering etc.) (Andreas Bombe)
- Treeview hints in paned browser
- Cover art now only uses the available space
- Support embedded covers art in WMA/Vorbis files

- Set composer, albumartist, sort tags when copying to an iPod
- Natural sorting in the song list
- Many song list speedups (sorting, filling, scrolling)
- Split up pattern results in paned browser with multi-value tags
- Only consider a song played after half has elapsed
- Undo/Redo support for all text entries
- New framework for showing running tasks, notifications in the status bar
- Text markup in the paned browser
- Restore maximized state
- Restore window position (Felix Krull)
- Make size of the queue adjustable (Florian Demmer)
- Mouse scrolling over the play button now changes songs
- Support alternate home directory using \$QUODLIBET\_USERDIR (jkohen)
- Make the default rating changeable (library reload needed)
- Drag and scroll in the song list
- Faster context menu opening
- Display playlist size (library reload needed) (Nick Boulton)
- Support queries without specifying a tag name
- All queries in the album browser use a standard operation (avg, sum etc.)
- Support ~rating, ~#bitrate in the album pattern
- Support separate song collection patterns in the paned browser
- Don't jump to a playing song if it was selected from the songlist
- Faster local cover search
- Support FreeDesktop trash spec
- Lower case option for file renaming (Nick Boulton)
- Various bug fixes, speed improvements (Jacob Lee, Johannes Rohrer, Tshepang Lekhonkhobe)
- Bug fixes:
  - Treeview hints now work with GTK+ >= 2.20
  - Search history now gets properly shared between browsers
  - Fix udev crashes
  - Paned browser leaks
  - Respect global filter in all browsers/filters
  - Don't lose tag values with differently cased tag names (APEv2)
  - Fix --set-browser (Carlo Teubner)
  - Properly handle the case where a playing song gets deleted
  - Fix redraw errors using compiz

- FSync on library save
  - Fix crash when ~/.gtk-bookmarks contains empty lines (Felix Krull)
  - Correctly identify rockboxed iPods
- Windows (Uninstall any previously installed version!):
  - Fix translation under Win 7
  - Fix cover art plugin saving
  - Add all partitions to the file selector
  - Fix various crashes with wide char user names
  - Fix icon under Win 7
  - Support multi-user installations
  - Fix freezes after opening certain folders
- New plugins:
  - Follow cursor play order plugin
  - Equalizer plugin
  - MPRIS 1.0/2.0 plugin
  - Ubuntu sound menu plugin
  - Rating reset plugin
  - Track repeat plugin (Nick Boulton)
- Plugins:
  - Go to bookmark plugin now menu based
  - Fix some album art plugin backends (Aymeric Mansoux, ..)
  - Improved “human” title casing (Nick Boulton)
  - Fix queue only plugin stopping the current song.
  - Only use allowed HTML in the notify plugin.
  - musicbrainz: allow writing sort tags (Michaël Ball)
- Translations:
  - new Latvian translation (Einars Sprugis)
  - new Basque translation (Piarres Beobide)
  - French translation updates (Nick Boulton)
  - Brazilian/Portuguese translation updates (Djavan Fagundes)
  - Russian translation updates (Anton Shestakov)

## 1.53 2.2.1 (2010-03-27) - Fewer than four out of ten people respect my promises – and possibly more!

- Fix for importing some mp3 files. [220]



- More fixes for the device backend (iPod, multi partition DAPs). [410, 412]
- Fix editing keys with multiple values. [440]
- Fix weighted playorder algorithm.
- Save songlist column patterns. [447]
- Some small fixes here and there.
- Plugin fixes:
  - Title case: Improved title casing for English text. (Nick Boulton)
  - Random Album: Algorithm improvements.
  - QLScrobbler: Fix preference pane ordering.
  - Album art: Some images weren't displayed. (Tomasz Miasko) [429]
  - Last.fm Sync, Musicbrainz: Minor fixes.
- Translations:
  - Galician, Spanish (Johán-Luís Miguéns Vila).
  - German (Rüdiger Arp).

## 1.54 2.2 (2010-02-02) - I know you are enjoying that song but a woman DIED

- Saved searches extended to Album and Paned browsers [41].
- Human sorting is now used in Album and Paned browsers [190].
- Windows is now supported (for real this time).
- foobar2000's broken TXXX:DATE now supported [220].
- Warnings are now printed for many missing dependencies.
- Fixes for device backends.
- Lyric downloading disabled until it can be fixed [273].
- Editing both key and value with multiple keys fixed (extruded) [393].
- Plugin changes:
  - AnimOSD: major update (Andreas Bombe, Christine Spang) [387].
  - MusicBrainz: major update.
  - Random Album: Changed algorithm to increase fairness.
  - QLScrobbler: Custom patterns for title and artist.
  - Last.fm Sync: new plugin to sync stats from Last.fm.
  - Notify OSD, Album Art: Minor fixes.
- Translations:
  - Galician, Spanish (Johán-Luís Miguéns Vila).
  - French (Bastien Gorissen).

## 1.55 2.1.98 [beta] (2010-01-04) - How are you going to convince people to use it?

- Christoph Reiter is now a maintainer.
- The GStreamer backend is now gapless [49].
- Win32 is once again supported [248].
- ID3 tags are removed from FLAC files upon saving [124].
- File extensions are converted to lowercase upon renaming [66].
- Thumbnails are now generated for artwork [140].
- Inline searches in the album list can now match people [239].
- Embedded album art is now supported in FLAC files [255].
- Bitrates are now reported in kbps. Library reload required [79].
- Additional ReplayGain settings (Nick Boulton) [132].
- Tag splitting setting is now order-sensitive [74].
- Paned browser now supports patterns for panes [45].
- Numeric columns have been given a few tweaks (Johan Hovold) [81].
- New ratings column options (Johan Hovold, Bastien Gorissen) [82, 83].
- Renaming when symlinks present no longer raises error (Philipp Weis) [353].
- Xine backend uses software volume control (Markus Koller) [370].
- Song positions are now saved and restored when quitting [218].
- DeviceKit-Disks (UDisks) supported for device discovery [252].
- Plugin changes:
  - New playlist export plugin [30].
  - New queue only playorder plugin [43].
  - New Python console plugin. [229]
  - Updated trayicon plugin [158].
  - Updated album art plugin (Eduardo Gonzalez) [139].
  - Updated qlscrobbler plugin (Nicholas J. Michalek) [376].
  - Updated lastfmsubmit plugin [292].
- Translations:
  - Russian (Anton Shestakov) [274].
  - Turkish (Türkerkan İnce).
  - German (Rüdiger Arp).
- Many bug fixes and performance improvements.

## 1.56 2.1 (2009-07-04) - My God, Utahraptor, that's THE PERFECT SOLUTION.

- Bug fixes:
  - Installer fixes [15, 27, 88]
  - Right-click on menu causes crash [14]
  - Removing a pane from the paned browser causes segfault [131]
  - Null bytes in tags are now stripped on load [177, 242]
  - zh\_CN translation updated [156]
- Support .oga file extension for Ogg Vorbis files [52]
- Support libre.fm scrobbling in qlscrobbler plugin
- Get Internet Radio channel listing from yp.icecast.org [18]
- Ignore errors during playback for ~#skipcount [37]
- URIs supported for `-play-file` and `-enqueue` [17]
- Many minor fixes and enhancements.

## 1.57 2.0 (2008-09-14) - Once upon a time there was a radical guy!

- Make Escape a synonym for Ctrl+W in QLTK Windows. (I#8)
- Actually fix playlist error.
- Fix Xine backend “Stop” behavior.

## 1.58 1.999 [beta] (2008-09-09) - It has been a memorable day

- Fix playlist error when loading songs.
- Unlock device when “stop” mmkey is pressed. (Javier Kohen, I#6)
- Restart song when rewinding and > 0.5s in. (Javier Kohen, I#7)
- Updated Galician and Spanish translations. (Johám-Luís Miguéns Vila)
- Make requirements consistent across all documentation.

## 1.59 1.99 [beta] (2008-09-08) - It is impossible to know if my dream came true

- New distutils-based build/test/install system.
- Multiple audio backend support.
  - Xine-based audio backend.
  - “Null” backend for Ex Falso.

- Tag Editing:
  - Tags From Path: “<~>” will eat a string but not save it.
  - Track Numbers: Allow numbering up to 999.
- Show image files in Ex Falso.
- Direct output to console and to a debugging window.
  - Functions are accessible to plugins as `print_d`, `print_e`, and `print_w`.
- `default_rating` configuration option. (Robert Muth)
- Many bug fixes and performance improvements.

## 1.60 1.0 (2007-05-05) - Yeah they just showed up one day – staring.

- Use Mutagen for ASF/WMA and MP4 support.
- Add `IsPlaying` and `GetPosition` to the D-Bus API. (Mickael Royer)
- Default “No Cover” image. (Jakub Steiner)
- Add `--unfilter` to reset browser filters.
- Sort `--enqueued` files, and add `--unqueue`.
- Basic SPC (SNES ROM audio) support.
- Paned Browser speed improvements. (Robert Muth)
- Errors when playing a song are now logged to a special `~errors` tag. It is visible from the Information screen, and can be reset.
- APEv2 tags can now override Musepack stream Replay Gain settings.
- Numerous bug fixes, especially in media device handling.
- Translation Updates:
  - Hungarian. (SZERVÁC Attila)
  - Finnish. (Jari Rahkonen)
  - Galician and Spanish. (Johám-Luís Miguéns Vila)
  - French. (Guillaume Ayoub)
  - Dutch. (Hans van Dok)
  - Japanese. (Yasushi Iwata)

## 1.61 0.24 (2006-11-19) - One wonders if our conversation today would be an appropriate epitaph.

- Media device (iPod and UMS so far) support. (Markus Koller)
- Delete removes songs from the queue. (sciyoshi)
- Per-browser window memory.
- Use Mutagen for WavPack and Musepack support.

- Keep filenames when given invalid patterns. (Markus Koller)
- Don't duplicate performers in ~performers. (Martin Bergström)
- Python 2.5 and GTK+ 2.10 compatibility.
- Fix Rename Files support on MP4 files.
- New Romanian translation, by Mugurel Tudor.
- New Slovak translation, by Lukáš Lalinský.
- Updated translations:
  - Traditional Chinese, by Hsin-lin Cheng.
  - Japanese, by Yasushi Iwata.
  - Galician and Spanish, by Johám-Luís Miguéns Vila.
  - Finnish, by Jari Rahkonen.
  - Hebrew, by Roe Haimovich.
  - Polish, by Tomasz Torcz
  - French, by Guillaume Ayoub.
  - German, by Rüdiger Arp.

## 1.62 0.23.1 (2006-08-28) - UNPOPULAR LIFE GOALS

- Multivalued tag renaming.
- Fix crash when ~/.gtk-bookmarks is not present.
- Disable 'title' completion. (hopefully temporarily)
- Parse "performer:role" tags and offer a ~performers synthetic tag.
- Updated Swedish translation, by Erik Christiansson.

## 1.63 0.23 (2006-08-14) - THE NARRATIVE OF LIFE.

- Bug Fixes:
  - Updated files no longer incorrectly appear in the paned browser.
  - Disambiguate 'filename' string for translation.
  - Hide unreadable files in Ex Falso.
  - Avoid (harmless) race condition when filling album list.
- "Select All Subfolders" menu item when browsing directories. (thanks to Alexandre Passos).
- Scan the library in the background when starting.
- Ogg FLAC and Speex files can be loaded.
- Plugin configuration IDs can be different from their names.
- Rewritten library code, many resulting UI improvements.
- Scan directories are used as File System roots.

- Replay Gain mode is chosen based on browser/play order.
- Internet Radio M3U support.
- Ex Falso runs on Win32 (thanks to Ben Zeigler).
- Song list headers can be changed via a context menu.
- True Audio (TTA) support.
- New Japanese translation, by Yasushi Iwata.
- New Traditional Chinese translation, by Hsin-lin Cheng.
- Updated Translations:
  - German, by Rüdiger Arp.
  - Polish, by Tomasz Torcz
  - French, by Guillaume Ayoub.
  - Galician and Spanish, by Johám-Luís Miguéns Vila and Javier Kohen.
  - Korean, by Byung-Hee HWANG and ChangBom Yoon.
  - Hebrew, by Roe Haimovich.
  - Portuguese, by Alexandre Passos.
  - Dutch, by Hans van Dok.
  - Hungarian, by SZERVÁC Attila.
  - Swedish, by Fredrik Olofsson.

## **1.64 0.22 (2006-07-06) - Man, forget television, books, films, short films, to a lesser extent plays and other theatre, and the remaining popular media!**

- The tray icon is now an optional plugin.
- A D-BUS interface is available. (thanks to Federico Pelloni)
- Tag editing values are autocompleted.
- Library Browser windows have more useful titles.
- New “~lyrics” synthetic tag.
- Python 2.4 is now required.
- Updated Translations:
  - Galician and Spanish, by Johám-Luís Miguéns Vila.
  - Italian, by Filippo Pappalardo.
  - Hebrew, by Roe Haimovich.
  - Finnish, by Jari Rahkonen.
  - Dutch, by Hans van Dok.
  - Polish, by Tomasz Torcz

- Portuguese, by Alexandre Passos.
- French, by Guillaume Ayoub.
- Bulgarian, by Rostislav Raykov.
- Hungarian, by SZERVÁC Attila.
- Korean, by Byung-Hee HWANG and ChangBom Yoon.

## **1.65 0.21.1 (2006-07-02) - Dude! It's not like you can't just make your own!**

- MP3s with POPM can be loaded again (Thanks, Hans van Dok)

## **1.66 0.21 (2006-06-10) - Faith, AND the possibility of weaponized kissing??**

- Bug Fixes:
  - Queue behaves correctly when randomizing two songs.
  - GStreamer error messages are properly localized.
  - Tray icon is more resilient to panel crashes.
  - “Jump...” distinguishes between identically-named albums.
  - application/ogg is recognized in audio feeds.
  - .pyo files are removed on clean.
  - util.unexpand caches the value of \$HOME.
  - Fix plugin function call ordering.
- UI Changes:
  - Improved tooltips in Preferences.
  - The Paned Browser shows song totals, and has a button to reset all selections.
  - Saving play count / rating tags can be turned off, or adjusted to a different email address.
  - The last-entered directory is used for Scan Directories configuration.
- pyvorbis is no longer required if you use Mutagen 1.3.
- Event plugins were redesigned, incompatibly.
- Test coverage data can be generated using trace.py.
- New Simplified Chinese translation by Emfox Zhou.
- New Hungarian translation by SZERVÁC Attila.
- Updated translations:
  - Finnish, by Jari Rahkonen.
  - Korean, by Byung-Hee HWANG and ChangBom Yoon.
  - Galician and Spanish, by Johám-Luís Miguéns Vila.

- Norwegian Bokmål, by Andreas Bertheussen.
- Italian, by Filippo Pappalardo.
- Polish, by Tomasz Torcz
- Lithuanian, by Jonas Slivka.
- Dutch, by Hans van Dok.

## **1.67 0.20.1 (2006-05-02) - Thanks for the eye-opener, dinosaur zombies!**

- Vorbis/FLAC tag editing works again.

## **1.68 0.20 (2006-05-01) - Feelings are boring. Kissing is awesome!**

- Bug Fixes:
  - `--play-file` will use the queue.
  - Audio Feeds remember download locations.
  - Song changes don't revert tag edits.
  - Browser song activation takes precedence over the queue.
  - Albums drag-and-drop in listed order.
  - Only reset relevant parts of Information windows on song change.
  - Deleting files not in the library removes them.
  - Non-numeric disc/track numbers sort properly.
  - Paned Browser no longer adds incorrect entries. (Debian bug )
  - Ex Falso no longer loads WAV or MOD files.
  - Allow more headers in Internet Radio and Audio Feeds.
  - New process launching method, `util.spawn`.
- UI Changes:
  - Indicator to show when songs come from the queue.
  - Rating submenu always appears in the song list.
  - Album covers hide when clicked again.
  - Select current song when jumping to it.
- New translations:
  - Norwegian Bokmål, by Andreas Bertheussen.
  - Swedish, by Erik Christiansson.
- Updated translations:
  - Polish, by Tomasz Torcz.
  - Dutch, by Hans van Dok.



- Finnish, by Jari Rahkonen.
- French, by Olivier Gambier.
- Galician and Spanish, by Johám-Luís Miguéns Vila.
- Hebrew, by Roe Haimovich.
- Portuguese, by Alexandre Passos.

## 1.69 0.19.1 (2006-04-04) - It's our secret! Our secret IDENTITY!

- Work around broken Fedora/Mandriva GNOME bindings.
- Fix global plugin directory scanning.
- Add originalartist to ~people.
- Updated Italian translation, by Filippo Pappalardo.
- Updated Korean translation, by Byung-Hee HWANG and ChangBom Yoon.

## 1.70 0.19 (2006-04-01) - i'm really thirsty you guys

- Simple X session management.
- Require Mutagen 1.0; drops Pymad.
- WAV support.
- New plugin types can be enabled/disabled.
- Album List can search and display any tags.
- “Bookmark” time offsets within songs.
- Song menu plugins require minor but incompatible updates.
- Searches, tagging patterns, and file renaming patterns can be given aliases and saved.
- Tag Editing:
  - MusicBrainz TXXX, artist/album/albumartist support.
  - Added albumartist, originalartist, originalalbum, originaldate, recordingdate.
  - Ratings, playcount Ogg Vorbis format changed.
  - COMM tags in Ex Falso are deleted properly.
- UI Changes:
  - Drops from e.g. Nautilus add to playlists/queue.
  - Clear button in Album List.
  - Horizontal scrollbar when absolutely necessary for the song list.
  - “Random” options use filtered lists.
  - Album sort is once again and forever default.
  - ‘Add to Playlist’ resorts playlist properly.
  - Enter in ‘Add a Tag’ moves from tag to value. (Debian bug )

- Standard context menu for all browsers.
- ‘Delete’ key works in Edit Tags.
- Type-ahead search works in the Album List.
- Bug fixes:
  - Double-appearances in the filesystem view.
  - FIFO misses some commands.
  - Stupid ‘refresh’ signal finally gone.
  - Error when seeking and keyboard can’t be grabbed.
- Updated translations:
  - Finnish, by Jari Rahkonen.
  - Galician and Spanish, by Johám-Luís Miguéns Vila.
  - Polish, by Tomasz Torcz.
  - Dutch, by Hans van Dok.
  - German, by Rüdiger Arp.
  - Lithuanian, by Jonas Slivka.
  - Hebrew, by Roe Haimovich.

## 1.71 0.18 (2006-03-03) - “Babies Sporting Monocles”?

- MP4 iTunes metadata can be written.
- Alt+s switches between search entry and song list.
- GStreamer 0.10 port.
- Album covers can be switched off in the Album List.
- The Album List text can be changed with a pattern.
- “Limit” in the Search view can take ratings into account.
- UI Changes:
  - Alt+Enter / Ctrl+I shows tags/information for selected songs.
  - DnD to playlists/the queue from the File System view.
  - F2 renames playlists.
  - “Add a Tag” autocompletes tag names.
  - RSS links can be dragged to the Audio Feeds sidebar.
- Bug Fixes:
  - ID3v1 tags no longer interfere with APEv2 tags.
  - Playing albums with one song no longer skips forwards.
  - ‘totaltracks’ Vorbis tags are read properly.
  - Adding songs to a playlist doesn’t unsort it.

- “Tags From Path” patterns are no longer greedy. (thanks to Decklin Foster)
- Internet Radio supports the ~people tag.
- Word-wrap in lyrics pane works properly.
- Ex Falso properly opens an initial directory. (thanks to ch.trassl)
- ‘~format’ is usable from –print-playing. ([2810])
- Plugin errors are captured in a dialog. ([2817])
- New synthetic numeric tags, ~#tracks and ~#discs. ([2828])
- Ex Falso no longer depends on GStreamer. ([2837])
- New Lithuanian translation by Jonas Slivka. ([2780])
- Updated translations:
  - Bulgarian, by Rostislav Raykov.
  - Finnish, by Jari Rahkonen.
  - Korean, by Byung-Hee HWANG and ChangBom Yoon.
  - Galician and Spanish, by Johám-Luís Miguéns Vila.
  - Hebrew, by Roe Haimovich.
  - German, by Rüdiger Arp.
  - Polish, by Tomasz Torcz.
  - Russian, by Sergey Fedoseev.
  - French, by Joshua Kwan.

## **1.72 0.17.1 (2006-01-19) - I'd like to have some nightmares, please!**

- Updated German translation, by Rüdiger Arp.
- Updated Russian translation, by Sergey Fedoseev.

## **1.73 0.17 (2006-01-18) - the grizzly icing on the prospector cake!**

- Lyrics plugin merged into Information dialog.
- “Edit Tags” now correctly removes all copies of multiple values, and displays “(missing from...)” for all missing values.
- More FIFO commands (song list and queue visibility).
- FLAC support no longer depends on pyflac.
- Load audio feeds without author information.
- ~#year, ~year internal tags.
- Numeric searches are rounded to two decimal places.
- New plugin architecture for tag editing dialogs.
- Korean translation, by ChangBom Yoon and Byung-Hee HWANG.

- Updated translations:
  - Russian, by Sergey Fedoseev.
  - Finnish, by Jari Rahkonen.
  - Portuguese, by Alexandre Passos.
  - Italian, by Filippo Pappalardo.
  - Dutch, by Hans van Dok.
  - Galician, by Johám-Luís Miguéns Vila.
  - French, by Joshua Kwan.
  - Polish, by Tomasz Torcz.

## 1.74 0.16 (2005-12-19) - would it make a difference if it never really came up often?

- Context menu plugins can make themselves insensitive.
- More command-line and FIFO options.
- Read-only M4A support. (thanks to Alexey Bobyakov)
- Wavpack support.
- Audio Feed (Podcast) support (requires <https://pythonhosted.org/feedparser/>).
- “One Song” (and repeat) play mode.
- Improved and configurable tray icon.
- New install system that is more FHS-compliant.
- ~laststarted internal tag.
- Accents are stripped when renaming to ASCII filenames.
- UI improvements:
  - Ex Falso lists are searchable in GTK 2.8.8+.
  - ^W closes transient windows.
  - More DnD support.
  - HIG-compliance for strings.
  - Double-click files in browsers to enqueue them.
  - Rename Files error dialog has a “Continue” button.
  - Ctrl-Left/Right changed to Ctrl-/,..
  - Playlist imports have a progress bar.
  - New icon that is not all black. (thanks to Tobias and Fabien)
  - Paned Browser entries have context menus.
  - Volume icons follow GTK+/GNOME theme.
- More memory and CPU optimizations.

- GStreamer error handling. (thanks to Zack Weinberg and Bastian Kleineidam)
- Musepack, MOD support migrated to Mutagen/ctypes modules.
- Updated translations:
  - Galician and Spanish, by Johám-Luís Miguéns Vila.
  - Italian, by Filippo Pappalardo.
  - Dutch, by Hans van Dok.
  - Finnish, by Jari Rahkonen.
  - Portuguese, by Alexandre Passos.
  - French, by Joshua Kwan.
  - Polish, by Tomasz Torcz.
  - German, by Rüdiger Arp.
  - Hebrew, by Roe Haimovich.

## 1.75 0.15 (2005-11-14) - Maybe I will.

- An ‘artist’ tag can be stored in the library for MODs.
- ‘All Albums’ remains on the album list after a search.
- The Play Queue displays its total time and has a clear button.
- Songs can be enqueued multiple times.
- ‘~people’ includes more people.
- Files can be added from remote URIs (e.g. HTTP).
- “Dumb” searches match any visible tags.
- Ratings are now searched with values of 0.0 to 1.0, and the number of visible notes is configurable.
- Useless columns are not displayed in Internet Radio.
- A single album cover can be refreshed in the Album List.
- Playlists have been rewritten:
  - Songs may now be in a playlist multiple times.
  - Playlists can be reordered directly, without a special window.
  - Songs can be added to playlists directly from the context menu.
  - M3U and PLS playlists (along with their songs) can be imported.
  - The interface is much more attractive.
- Drag-and-drop is generally more usable, faster, and attractive.
- Many optimizations, especially during startup.
- Updated translations:
  - Russian, by Sergey Fedoseev.
  - Galician, by Johám-Luís Miguéns Vila.

- Dutch, by Hans van Dok.
- French, by Joshua Kwan and Fabien Devaux.
- Hebrew, by Roe Haimovich.
- Finnish, by Jari Rahkonen.
- Polish, by Tomasz Torcz.
- Italian, by Filippo Pappalardo.
- New translations:
  - Spanish and Portuguese, by Johám-Luís Miguéns Vila.

## 1.76 0.14 (2005-10-22) - I'm almost certain!

- Internet radio / Shoutcast browser.
- Album List separates albums with different labelids.
- Ex Falso displays all available plugins in its menu.
- Useful ~#lastplayed/~#added/~#mtime display thanks to Lalo Martins.
- New Album List search keys and sorting options.
- New translations:
  - Galician, by Johám-Luís Miguéns Vila.
  - Italian, by Filippo Pappalardo.
- Updated translations:
  - Finnish, by Jari Rahkonen.
  - Polish, by Tomasz Torcz.
  - Dutch, by Hans van Dok.
  - Hebrew, by Roe Haimovich.
  - German, by Rüdiger Arp.
  - Russian, by Nikolai Prokoschenko.
- Many bug fixes.

## 1.77 0.13.1 (2005-09-15) - People will fall for this for sure!

- Fix playlist creation.
- Unplay when no song is playing.

## 1.78 0.13 (2005-09-11) - So, um... let's- fletcherize?

- The GStreamer backend is cleaned up, and is now the only backend. This results in lower background CPU usage and many fixes to our audio processing. Gapless playback is gone.

- A play queue was added.
- A file system browser has been added. This can view, edit, and play files outside of your library.
- The Paned Browser has a search entry.
- Search Library lets you limit the number of results.
- `~/ .quodlibet/browsers` is now scanned for custom browsers.
- Synthetic tags (“~dirname”, “~basename”, &c.) can be searched.
- Similarly, synthetic tags can be used in the Paned Browser.
- New synthetic tags, “~people” and “~playlist”.
- If the tray icon is visible, closing QL’s main window will minimize it. To actually quit, choose “Quit” from the Music menu or icon.
- Search Library and the Album List search entry have tag completion.
- Ex Falso supports plugins.
- Updated Russian translation by Nikolai Prokoschenko and Sergey Fedoseev.
- Updated French translation by Joshua Kwan.
- Updated Finnish translation by Jari Rahkonen.
- Updated Dutch translation by Hans van Dok.
- Updated Hebrew translation by Roee Haimovich.

## 1.79 0.12 (2005-07-31) - focus ENTIRELY on the sexy bits.

- New Mutagen ID3 reader/writer.
- Experimental GStreamer backend.
- Drag-and-drop to playlists.
- Weighted random playback.
- MP3 and Musepack ReplayGain support.
- Larger plugin manager window.
- Automatic mount point detection.
- Support for multiple soundcards.
- Localization enhancements.
- Translation updates:
  - Dutch, thanks to Hans van Dok.
  - Finnish, thanks to Jari Rahkonen.
  - French, thanks to Joshua Kwan.
  - German, thanks to Rüdiger Arp.
  - Hebrew, thanks to Roee Haimovich.
  - Russian, thanks to Sergey Fedoseev and Nikolai Prokoschenko.
  - Polish, thanks to Witold Kieraś.

- The usual round of interface tweaks and bug fixes.

## **1.80 0.11 (2005-05-10) - spicy burnsauce with a side of ZING!**

- Plugins (either appearing in the right-click menu, or triggered on a player event)
- Browse songs by album, with a cover display.
- “Library Browser” added to search/edit files without disturbing your playlist.
- Played songs are automatically removed from dynamic playlists.
- “Background filters” for the paned browser and search entry.
- Create/remove empty folders from within Ex Falso.
- ‘0’ to ‘4’ keys or mouse clicks can set song ratings.
- Depends on PyGTK 2.6 (as well as GTK 2.6).
- –status to print the player’s status.
- Russian translation, thanks to Sergey and Andrey Fedoseev.
- Partial French translation, thanks to Joshua Kwan.
- OSD moved to a plugin.

## **1.81 0.10.1 (2005-04-04) - What if I said I’m not really kidding?**

- The main window stays hidden when the song changes.

## **1.82 0.10 (2005-04-02) - As it turns out, my life is NOT THAT INTERESTING!**

- –seek supports +/- prefix to seek relative to the current position.
- Added Ex Falso, a tag editor based on QL (without audio playback).
- Switched MP3 genres from TIT1 to TCON.
- The library is saved automatically every 15 minutes.
- Tag by Pattern/Rename Files save patterns used.
- Adding tags with specific formats (‘date’) is less prone to error.
- Several display bugs and non-HIG windows were fixed.
- Pane-based (Rhythmbox/iTunes-style) library browser.
- Tag by Pattern/Rename Files previews can be manually edited before saving.
- Kind of browser (none, search, playlists, paned), song list sort order, and what you were browsing are remembered when you exit.
- At least one lockup-causing bug was fixed.
- Song ratings, on a 0 to 4 scale.



- Masked directories work again.
- No more dependency on Glade.
- A new icon and website.

## **1.83 0.9 (2005-03-04) - I don't want any trouble, cephalopods!**

- Major updates to the Properties dialog:
  - A new detailed 'Information' tab was added.
  - Middle click pastes current PRIMARY clipboard text in 'Edit Tags'.
  - Text in the 'Rename Files' dialog can be conditionalized.
- Non-UTF-8 filesystem encodings support (via `CHARSET/G_BROKEN_FILENAMES`).
- New numeric keys "added" and "skipcount" can be searched on.
- The `--print-playing` format string syntax has changed to match the one now used in 'Rename Files'.
- New query language enhancements (ternary relation operators, string comparisons, and Lynx-like case-sensitivity).
- The tray icon now pauses/unpauses on middle click, adjusts volume with the scroll wheel, and skips forward/backward on buttons 6/7.
- This release depends on GTK 2.6 for its new media icons.
- PMP support was removed.
- Updated German translation (thanks, Bastian!)

## **1.84 0.8.1 (2005-02-06) - Our story takes a sudden dive...**

- Fix a crash when encoding information is not available.

## **1.85 0.8 (2005-02-04) - I make jokes about whomever I please!**

- New/reloaded libraries take 20% less disk space.
- Double-clicking an album cover displays it in a larger window.
- `--shuffle`, `--repeat`, and `--volume` (`--volume-up` and `--volume-down` are deprecated and will be removed).
- Any tag name can be written to (and read from) an MP3 file.
- Playlists containing arbitrary songs can be created.
- The libmodplug Python wrapper must be downloaded separately.
- The MPC/MP+ format is supported (with a separate wrapper).
- FLAC supports ReplayGain.
- Internal changes made some things faster and others slower.
- Polish translation, thanks to Michal Nowikowski.

- German translation, thanks to Bastian Kleineidam.

## 1.86 0.7 (2004-12-18) - I'm going to ho-ho-hoard all these nuts!

- Default to proper 'alsa09' driver (rather than 'alsa').
- "Random Foo" searches are now anchored with ^...\$.
- Tag By Pattern values can be edited before being saved.
- Right-clicking on the status icon brings up a menu.
- -play, -pause, -seek-to, -query, -play-file.
- OSD (gosd) support, thanks to Iñigo Serna and Gustavo J. A. M. Carneiro. A library reload is needed to use it.
- FreeDesktop-style .folder.png files supported for album covers.
- Playlist/search UI elements can be hidden.
- Dragging playlist columns reorders them.
- Library rebuilds no longer lose play counts.
- Configurable menu accelerators when gtk-can-change-accel is set.
- Delete songs (or move them to the trash) from the player.

## 1.87 0.6 (204-12-02) - People laugh at typos in heaven?!

- Many new filtering options (top 40, not played in X days).
- Mass-set track numbers.
- Tag by Pattern can replace or add new tags.
- Maskable mount points. This lets you add files from an NFS share or portable device and not have to read them if you unmount and remount it.
- Support for sending files to PMPs from the context menu.
- -next, -previous, -play-pause, -volume-up, -volume-down.
- MOD/IT/XM/etc. support, using libmodplug and an included C extension.
- Right-clicking the status icon will pause/unpause.
- Seeking in FLACs.
- Bug fixes (including at least one crash).

## 1.88 0.5 (2004-11-18) - Everything's fine, CEPT YOU GOT NO LEGS!

- The ao audio backend is back; see the "AUDIO BACKENDS" section in the manual page for instructions on using it. This should let ALSA users use software mixing.
- ID3 APIC (embedded picture) support.
- VorbisGain (<https://sjeng.org/vorbisgain.html>) support.

- A context menu for common operations was added to the properties dialog.
- Tag values can be set from filename patterns, or vice-versa.
- Dates can be saved in MP3s now.
- `--print-playing` option, with a format string.
- More UI tweaks.
- Translation template update.
- Many bug fixes; please reload your library. You can now reload your library from the “Music” menu.

## 1.89 0.4 (2004-11-09) - The Power of Language

- Many bug fixes, primarily due to unit testing.
- Tweaks to cover detection to pick ‘frontcover’ over ‘backcover’.
- Tweaks to song display, including proper support for the ‘author’ tag.
- Remember size between invocations.
- A freedesktop.org-compatible system tray icon, using the Egg status icon code by Anders Carlsson and Sun.
- Multimedia key support, provided they’re mapped (e.g. by Acme), using the MmKeys object from Muine by Lee Willis and Jan Arne Petersen.
- UI tweaks to the main window.
- Button to link to a song’s website, or a Google search.
- Infrastructure is in place for i18n/l10n, but I’m totally new to this, so I could’ve done something horribly wrong.

## 1.90 0.3 (2004-11-01)

- Handle mono MP3s correctly.
- Crash less, especially when editing tags.
- Many smaller bug fixes.

## 1.91 0.2 (2004-10-30)

- Song properties dialog, featuring mass tag editing/addition/removal.
- Build/installation scripts.
- Interface tweaks for HIG compliance and accessibility.
- Try to save the library when `^C` is pressed.
- `~/quodlibet/current` interface to currently-playing song.
- Save current query and song on exit.
- An icon.
- FLAC support. Writing to FLAC tags could be *very* buggy, so if you value your tags, please back them up.

## 1.92 0.1 (2004-10-30)

- Initial release.

## CHAPTER 2

---

### Downloads

---

<b>Linux:</b>	<i>Flatpak</i>	
	<i>Ubuntu</i>	<i>Debian</i>
	<i>Fedora</i>	<i>openSUSE</i>
	<i>Arch Linux</i>	
<b>Windows:</b>	<i>Windows</i>	
<b>macOS:</b>	<i>macOS</i>	
<b>Development:</b>	<i>Release Tarballs</i>	

All files are signed with the following key: 0EBF 782C 5D53 F7E5 FB02 A667 46BD 761F 7A49 B0EC

---

### 2.1 Release Tarballs

Release	File	SHA256	PGP
Quod Libet 4.2.1	<a href="#">quodlibet-4.2.1.tar.gz</a>	SHA256	SIG
Quod Libet 4.1.0	<a href="#">quodlibet-4.1.0.tar.gz</a>	SHA256	SIG
Quod Libet 4.0.2	<a href="#">quodlibet-4.0.2.tar.gz</a>	SHA256	SIG

For old releases see the [full file listing](#).

### 2.2 Flatpak

[Quod Libet on Flathub](#)

## 2.3 Ubuntu

### Stable Repo (14.04+):

```
$ sudo add-apt-repository ppa:lazka/ppa
```

### Unstable PPA (16.04+):

```
$ sudo add-apt-repository ppa:lazka/dumpingplace
```

## 2.4 Debian

### Stable Repo:

- Debian Stable:

```
# deb http://lazka.github.io/ql-debian/stable/ quodlibet-stable/

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5A62D0CAB6264964
sudo apt-get update
sudo apt-get install quodlibet
```

### Unstable Repo:

- Debian Testing:

```
# deb http://lazka.github.io/ql-debian/testing/ quodlibet-unstable/

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5A62D0CAB6264964
sudo apt-get update
sudo apt-get install quodlibet
```

## 2.5 Fedora

- [Fedora Stable Repo](#)
- [Fedora Unstable Repo](#)

For stable releases check out the [official repos](#) first - they usually contain the latest release.

## 2.6 openSUSE

- [openSUSE Stable Repo](#)
- [openSUSE Unstable Repo](#)

## 2.7 Arch Linux

### Stable:

```
$ pacman -S quodlibet
```

**Unstable:** See [quodlibet-git](#) in the [AUR](#).

## 2.8 Windows

Release	File	SHA256	PGP
Quod Libet 4.2.1	<a href="#">quodlibet-4.2.1-installer.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.1.0	<a href="#">quodlibet-4.1.0-installer.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.0.2	<a href="#">quodlibet-4.0.2-installer.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>

Release	File	SHA256	PGP
Quod Libet 4.2.1 (portable)	<a href="#">quodlibet-4.2.1-portable.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.1.0 (portable)	<a href="#">quodlibet-4.1.0-portable.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.0.2 (portable)	<a href="#">quodlibet-4.0.2-portable.exe</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>

For old releases see the [full file listing](#).

## 2.9 macOS

Release	File	SHA256	PGP
Quod Libet 4.2.1	<a href="#">QuodLibet-4.2.1.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.1.0	<a href="#">QuodLibet-4.1.0.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Quod Libet 4.0.2	<a href="#">QuodLibet-4.0.2.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>

Release	File	SHA256	PGP
Ex Falso 4.2.1	<a href="#">ExFalso-4.2.1.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Ex Falso 4.1.0	<a href="#">ExFalso-4.1.0.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>
Ex Falso 4.0.2	<a href="#">ExFalso-4.0.2.dmg</a>	<a href="#">SHA256</a>	<a href="#">SIG</a>

For old releases see the [full file listing](#).





### 3.1 Audio Playback

- Multiple audio back-ends (GStreamer, xine-lib)
- Replay Gain support
- Auto-selects between ‘track’ and ‘album’ mode based on current view and play order
- Applies clipping prevention whenever available
- Configurable default (fallback) and pre-amp values to suit any audio setup
- Multimedia key support
- Real shuffle mode, that plays the whole playlist before repeating
- Weighted (by rating) random playback
- Proper ‘Previous’ support in shuffle mode
- A play queue
- Bookmarks within files (or playlists, with a plugin)

### 3.2 Editing Tags

- Full Unicode support
- Make changes to many files at once
- Make changes across all supported file formats
- Tag files based on their filenames with configurable formats
- Rename files based on their tags
- No ugly %a, %t patterns - more readable <artist>, <title> instead

- Fast track renumbering
- See full instructions at [Editing Tags](#)

### 3.3 Audio Library

- Watch directories and automatically add/remove new music
- Hide songs on removable devices that may not always be there
- Save song ratings and play counts
- Lyrics downloading and saving
- Internet Radio (“Shoutcast”) support
- Audio Feeds (“Podcast”) support

### 3.4 User Interface

- Simple user interface to Just Play Music if you want
- Useful as a small window or maximized, no feeling cramped or wasted space
- Album cover display
- Full player control from a tray icon
- Recognize and display many uncommon tags, as well as any others you want. Especially useful for classical music.

### 3.5 Library Browsing

- Simple or regular-expression based search
- Constructed playlists
- iTunes/Rhythmbox-like paned browser, but with any tags you want (Genre, Date, etc)
- Album list with cover art
- By directory, including songs not in your library

### 3.6 Python-based plugins

- Automatic tagging via [MusicBrainz](#) and CDDB
- On-screen display popups
- Last.fm/AudioScrobbler submission
- Tag character encoding conversion
- Intelligent title-casing of tags
- Find (and examine / remove) near-duplicate songs across your entire collection

- Audio fingerprinting of music
- Control Logitech Squeezebox devices.
- Scan and save Replay Gain values across multiple albums at once (using gstreamer)

## 3.7 File Format Support

- MP3, Ogg Vorbis / Speex / Opus, FLAC, Musepack, MOD/XM/IT, Wavpack, MPEG-4 AAC, WMA, MIDI, Monkey's Audio

## 3.8 UNIX-like integration

- Player control, status information, and querying of library from the command line
- Can use named pipes to control running instance.
- Now-playing is available as a fixed file



### 4.1 Bug Tracker

We use the GitHub bug tracker:

- List of existing issues: <https://github.com/quodlibet/quodlibet/issues>
- Create a new issue: <https://github.com/quodlibet/quodlibet/issues/new>

### 4.2 Git Repo

Quod Libet uses [Git](#) for source control and is hosted on [GitHub](#):

<https://github.com/quodlibet/quodlibet>

**Running from Source** (no installation is needed) :

```
$ git clone https://github.com/quodlibet/quodlibet.git
$ cd quodlibet
$ ./quodlibet/quodlibet.py
```

To get all the dependencies needed for running Quod Libet the easiest way is to use one of the unstable PPAs / repos listed on the [download page](#). They pull in or contain all the needed dependencies for the latest code and are kept up to date.



## 5.1 Overview

### 5.1.1 The Fundamentals

#### Explanation of the GUI elements

When you start up Quod Libet for the first time, and have configured your library, it should look something a little like the screenshot here, which is shown with annotations. These are all explained below.

#### 1. Player Controls

Provides buttons for seeking, changing volume and changing the currently played song.

By right-clicking, they also provide context menus with advanced options:

- The volume button lets you choose the active replaygain mode (album, track, automatic).
- The seek button can display the remaining instead of the elapsed time and provides options for interacting with bookmarks.
- The context menu of the play button has an option to pause playback after the current song.

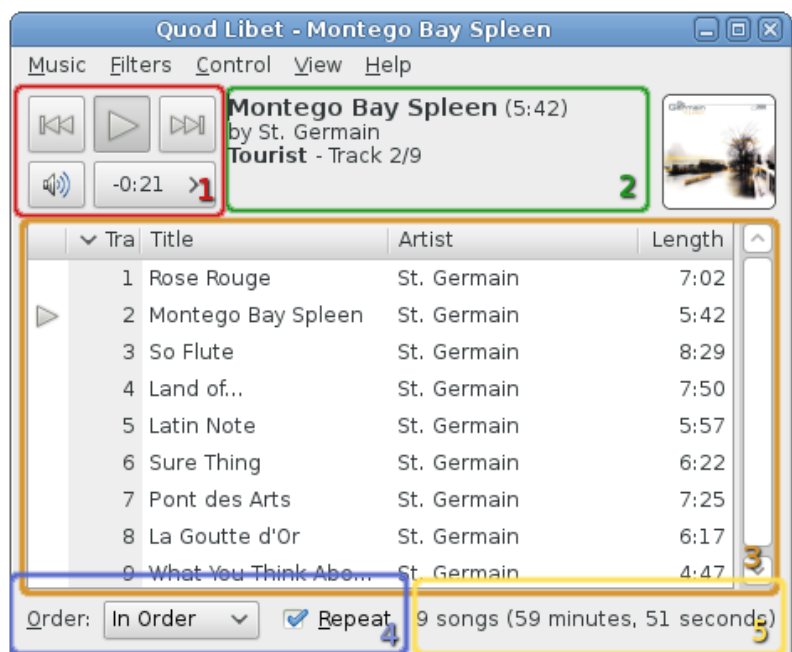


Fig. 1: A basic view of Quod Libet, and its GUI elements

## 2. Current Song Information

Displays information about the currently playing song.

Again, the context menu contains more options, like rating and tag editing for the current song.

There is also an entry called *Edit Display...* which lets you change the *tag pattern* used for display. This typically includes the people, title, album and maybe file format, bitrate etc (including *internal tags*). This also allows *text markup* for styling of your display area.

## 3. The Song List & Browser

As the name suggests, the song list is a list of matching songs presented by the current browser. Double-clicking a song in the song list will start playing it immediately.

The browser consists of a way to filter your library (songs), e.g. a simple search or by a list of albums. In the image above, no browser is active so only the current song list is shown. You can change the active browser by selecting one in the *View* menu.

## 4. Playback Order

The active playback order affects the selection of the next song or what happens if you manually switch to a new song.

Note that Play Order Plugins add to this list allowing QL new ways of ordering songs, like following your selection or adding songs to the queue instead of playing them immediately.

The *Repeat* checkbox can be used in conjunction with the playback orders, to repeat the currently selected view of songs after all song have been played or skipped.

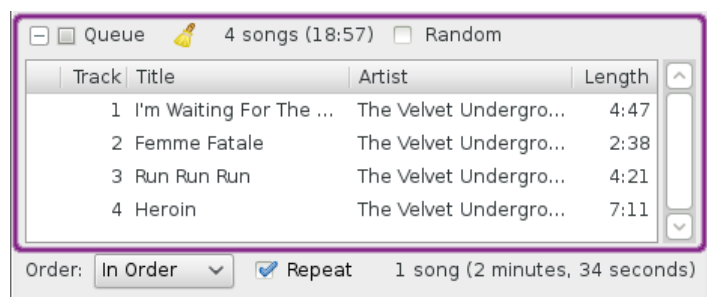
It's important to note that the playback order will not affect the queue, which has its own playback settings.

## 5. Song Information

The area in the bottom-right corner shows information about all songs in the song list *or* about your song selection if more than one song is selected.

It also shows progress information of background operations like importing music.

### 5.1.2 The Queue





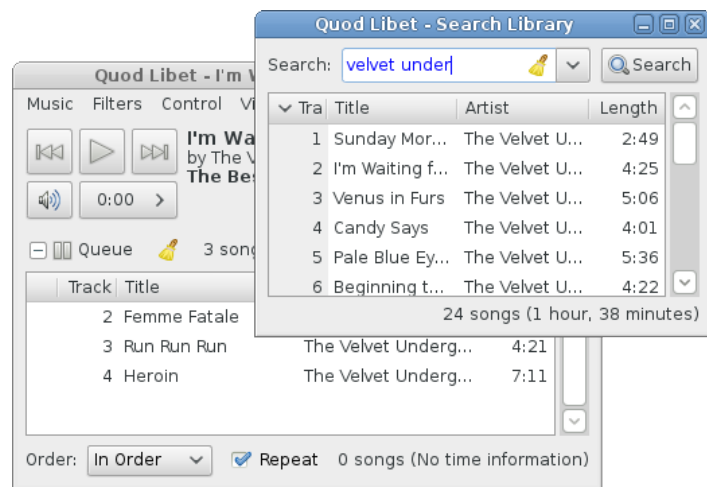
The play queue and the song list determine what's played next. Unlike the song list the queue consumes songs (from the queue not your library) as they are played. That is, once a song starts playing it disappears from the queue.

The queue however, takes precedence whenever it has songs. After it's empty, playback will continue in the main song list where it left off.

To add songs to the queue, right-click and *Add to Queue*, use the shortcut CTRL+Return or select songs in a different browser window.

For more information, see the [Queue Guide](#).

### 5.1.3 More Browsers



Besides the active browser in the main window, you can open as many different browser windows as you want by selecting one under *Browse > Open Browser* or in the tray icon plugin context menu.

In a separate browser, double-clicking a song will result in it being added to the queue rather than played immediately (as per the main browser).

See the [Browsers Guide](#) for full details on browsers and the various implementations.

Example usage:

- Manage playlists by dragging songs across browsers (eg Search browser to Playlist browser)
- Edit the tags of some songs while the ones in the main song list get played etc.
- Check out other songs by the artist currently playing without disturbing that list.

### 5.1.4 The Library

If all your music lives in one folder on your computer, adding your entire music collection in Quod Libet is easy.

From the main menu bar, select *File > Add a Folder* and browse to the top-most folder containing your music. By default Quod Libet will scan your selected folder recursively, adding all the songs found to your library automatically. Add more folders if you wish.

Depending on the size of your music collection, it may take a few minutes to perform the initial scan.

## 5.1.5 Plugins



The image above shows the plugin manager, from where you can enable / disable / configure all available plugins.

## 5.2 Tags

### 5.2.1 An Introduction to Tags

#### Overview

Quod Libet supports free-form tags for most of the common audio formats. This means you can name your tags in any way you want, but using common tag names for common purposes is advised because it helps QL to write the tags in a way that other media players can understand them and also helps QL understand certain tag values and make use of them.

Quod Libet also supports expressions using “tied tags” in several contexts - see [Tied Tags](#)

#### Sort Tags

Tags `artistsort`, `albumsort`, `albumartistsort`, `performersort` (including roles) will be used for sorting if they are present.

Lets say you have a song with an artist tag *The Beatles* and want it to be sorted as if it was named *Beatles, The*, you can add an `artistsort` tag containing *Beatles, The*.

- QL includes a basic plugin for creating such sort tags automatically.
- The musicbrainz plugin includes an option to write sort tags if found.

#### Programmatic Tags

A **programmatic tag** is one typically added by either Quod Libet / Ex Falso or another program and not designed for human consumption. Examples are `replaygain_track_gain` or `musicbrainz_albumid`. Note that to see these, you must turn on “Show programmatic tags” in the relevant preferences window.

## Internal Tags

Internal tags are tags that start with a `~` like `~people`, `~length` or `~year`. They are either not stored in files themselves, or are derived from the values in the files.

See [Internal Tags](#) for a complete list.

## Performer Roles

Quod Libet supports performer roles, which let you associate a performer in a song with a specific instrument. To use this feature, name a tag `performer:role`. Quod Libet will display the roles alongside the names in `~people:roles` and `~performers:roles`.

## Album Identification

Quod Libet uses various tags to define what songs are in the same album.

For songs to be in the same album, both `album` and `albumartist` have to be the same. In addition the first non-empty value of the following tags have to match (checked in this order): `musicbrainz_albumid`, `labelid`, `album_grouping_key`.

## Common scenarios

“I have a two disc album and each disc is shown separately.”

Make sure the album tags (and `albumsort` if present) (remove ‘CD1/2’ etc.) and `albumartist` tags are the same.

“Two albums have the same name and are merged.”

Add an `albumartist` tag for at least one of them.

## Common Questions

“Why doesn’t QL know that my albums are different ones by seeing that they don’t have the same artist?”

There are many songs that have multiple artist, so this can’t be decided on the basis of artist tags.

## Replay Gain Tags

The following (fairly standard) tags are used for volume adjustment:

- `replaygain_track_gain`
- `replaygain_track_peak`
- `replaygain_album_gain`
- `replaygain_album_peak`

See the [Replay Gain](#) section for further details.

## 5.2.2 Internal Tags

Quod Libet stores a number of tags internally, that are either not stored in files themselves, or are derived from the values in the files.

## String Tags

- `~basename`: The last component of the full path name
- `~dirname`: Everything but the last component of the file path name
- `~filename`: The full path name
- `~format`: The file format (e.g. "MPEG-4")
- `~codec`: The audio codec (e.g. "AAC LC")
- `~encoding`: Encoder name, version, settings used (e.g. "LAME 3.97.0, VBR")
- `~length`: The length of the file in H:MM:SS format
- `~mountpoint`: The component of the full path name that corresponds to the file's immediate parent mount
- `~performers`: A list of performers
- `~people`: A list of all people involved in the song
- `~rating`: A string representation of the song's rating (e.g. `1.0`). Note that in most formats these are per email address.
- `~uri`: The full URI of the song
- `~year`: The release year, derived from the `date` tag
- `~originalyear`: The original year, derived from the `originaldate` tag
- `~playlists`: Comma-separated playlist names in which the song is included
- `~filesize`: Human formatted size (e.g. *4.5 MB*)

### The `~people` Internal Tag

The internal `~people` tag combines the following tags to one: `albumartist`, `artist`, `author`, `composer`, `~performers`, `originalartist`, `lyricist`, `arranger`, `conductor` in this exact order.

For sorting, this means that all album artists come first followed by all artists and so on. For song collections / albums, the values of each included tag are sorted by frequency.

**Variants:** `~people:roles` includes roles e.g. "The Parley of Instruments (Orchestra), David Thomas (Bass)". The roles are either derived from the source tag name (`composer=Joseph Haydn` → `Joseph Haydn (Composition)`) or from the performer role (`performer:composition=Joseph Haydn` → `Joseph Haydn (Composition)`). For the latter see the `~performers` tag.

`~people:real` excludes *Various Artists*, commonly used as a placeholder for album artists on compilations, etc.

### The `~performers` Internal Tag

The internal `~performers` tag combines all the artists specified in the performer tags to a single one.

Example: `performer:vocals=Brandon Patton`, `performer:banjo=Béla Fleck`

`~performers` will then display "Brandon Patton, Béla Fleck"

**Variants:** `~performer:roles` includes the roles as well. For the above example it will display "Brandon Patton (Vocals), Béla Fleck (Banjo)"

## Song Collections / Albums

- `~length`: The length of all songs in H:MM:SS format
- `~long-length`: The length of all songs in “H hours, M minutes, S seconds” format
- `~tracks`: The real number of songs in the collection in “# track(s)” format
- `~discs`: The number of different discs in “# disc(s)” format
- `~rating`: The average rating in music notes
- `~filesize`: Total Human formatted size (e.g. *4.5 MB*)

All other tags return a list of values retrieved from all songs, without duplicates, sorted by their number of appearance.

## Numeric Tags

- `~#added`: The date the song was added to the library
- `~#bitdepth`: The bitdepth of this song
- `~#bitrate`: The bitrate of the song, in kilo bits per second
- `~#disc`: The disc number of the song (the first half of the `discnumber` tag)
- `~#channels`: The channel count
- `~#discs`: The total number of discs in this song’s collection
- `~#filesize`: The size in bytes of this song
- `~#lastplayed`: The time this song was last played through
- `~#laststarted`: The time this song was last started
- `~#length`: The length of the song, in seconds
- `~#mtime`: The time this file was last modified
- `~#playcount`: The total number of times you’ve played the song through
- `~#rating`: The rating of the song, as a number between 0 and 1.
- `~#samplerate`: The sample rate of this song
- `~#skipcount`: The total number of times you’ve skipped through the song
- `~#track`: The track number of the song (the first half of the `tracknumber` tag)
- `~#tracks`: The total number of tracks in the album
- `~#year`: The release year, derived from the `date` tag
- `~#originalyear`: The original year, derived from the `originaldate` tag

Note some numeric tags have string tag equivalents (see above) for human-readable format.

## Song Collections / Albums

- `~#tracks`: The real number of songs in the collection
- `~#discs`: The number of different discs in the collection

For all other numeric tags it is possible to define numeric functions by appending `:numeric_func` to the tag name (`~#playcount:avg` for example). All internal numeric tags use a default function in case no function is given. For user defined numeric tags the average value is returned by default.

- `avg`: Returns the average value (`~#rating`)
- `sum`: Returns the summation of all values (`~#length`, `~#playcount`, `~#skipcount`, `~#filesize`)
- `min`: Returns the smallest value (`~#year`)
- `max`: Returns the largest value (`~#added`, `~#lastplayed`, `~#laststarted`, `~#mtime`)
- **bav**: Returns the **Bayesian average value** (`~#rating`) Being most appropriate for ratings, the parameter is adjusted globally under the preferences for ratings.

### 5.2.3 Tied Tags

Tied tags are expressions to produce a readable formatted output of more than one tag. Tag values are outputted left to right, with the default output separator as `-`. To combine multiple tags start the expression with `~` and separate each tag by `~` again.

Examples:

Included Tags	Tied Tag	Result
artist, title	<code>~artist~title</code>	<i>Bob Marley - Jammin'</i>
<code>~#track</code> , <code>~dirname</code>	<code>~~#track~~dirname</code>	<i>5 - /home/user/music/Reggae</i>
date, artist, album, <code>~filesize</code>	<code>~date~artist~album~~filesize</code>	<i>2000 - Bob Marley - Songs of Freedom (Various) - 6.9 MB</i>

### Usage in Quod Libet

Tied tags can be used in tag patterns and searches as if they were normal tags:

<code>~artist~title=/AC.?DC/</code>
<code>&lt;tracknumber &lt;tracknumber&gt;. &gt;&lt;title~version&gt;</code>

### 5.2.4 Tag Patterns

Tag patterns allow more complex string representation of tags, with a notion of if-null defaulting; whilst this sounds complex (and it can be!) it's very useful where libraries have wildly varying tags.

Tag patterns can include strings (e.g. tag names) enclosed in angled brackets like this

`<artist> - <title> (<album>)`

Which might produce:

The Beatles - Drive My Car (Rubber Soul)

### Usage

Tag patterns in QL can be used to change the information displayed in various places, like the playing song area, columns in the song list and the album list in the album browser.

They can be used to group songs in the paned and the album collection browser in more complex ways. And, of course, tag renaming based on tags uses tag patterns to create the file names and folder structure.

## Conditional Tags

A simple if-then-else concept can be used in tag patterns, testing if a tag is non-empty. The syntax uses the pipe (|) character as a delimiter, in either of these formats:

- `<tag-expression|non-empty-value>` or
- `<tag-expression|non-empty-value|empty-value>`

So using the full (second) form, a Pattern of:

```
<album|Has An Album|No Album>
```

produces *Has An album* for any song with at least one album tag value, else *No Album*.

Note that these can be recursive, i.e. both *non-empty-value* and *empty-value* are themselves tag patterns, which could contain a conditional. A more useful example now:

```
<albumartist|<albumartist>|<artist>>
```

This will look for the `albumartist` tag and display that if available, else use `artist` (nearly always available).

Examples:

- `<~year|<~year>. <album>|<album>>: 2011. This is an album title`
- `<title>, by <albumartist|<albumartist>|<composer>|<composer>|<artist>>>: Liebstraum no. 3, (by Franz Liszt)`

## Conditional Tags With Comparisons

In addition to checking if a tag value is empty, the “if” expression can also contain a value comparison using the same syntax as the [search](#):

```
<sometag=test|the value was test|it was something different>
```

or more complex ones (note the needed escaping):

```
<artist=\\(Townshend, Who)|foo|bar>
```

## Text Markup

In some situations the resulting text will be displayed in the user interface like for example the album list or the area which displays information about the currently playing song. To style the resulting text you can use the following tags in combination with the tag patterns.

Tag	Result
<code>[b]..[/b]</code>	<b>Bold</b>
<code>[big]..[/big]</code>	<b>Bigger</b>
<code>[i]..[/i]</code>	<i>Italic</i>
<code>[small]..[/small]</code>	<small>Smaller</small>
<code>[tt]..[/tt]</code>	<b>Monospace</b>
<code>[u]..[/u]</code>	<u>Underline</u>
<code>[span] [/span]</code>	see below

The `span` tag can define many more text attributes like size and color: `[span size='small' color='blue']...[/span]`. See the [Pango Markup Language](#) page for a complete list of available attributes and values.

A complete example might look like this:

```
[span weight='bold' size='large']<title>[/span]<~length| (<~length>)> : [b]<~rating>[/b]
↪b<version|
[small] [b]<version>[/b] [/small]><~people|
by <~people>><album|
<album><tracknumber| : track <tracknumber>>>
```

Note also the literal newlines.

## 5.3 Searching Your Library

Pretty much every view in Quod Libet contains a search entry where you can enter search terms and save them. Quod Libet will search in artist, album, title, version, and any other visible columns for what you enter.

If that's enough for you, you can stop reading now. But what if you want something more powerful?

### 5.3.1 Combining Searches and Negation

You can combine search terms using `&` (“and”) and `|` (“or”).

If you want to listen to Electronic music but no Ambient:

```
&(electro, !ambient)
```

Or you want to get all songs by [Neutral Milk Hotel](#) including the solo performances of [Jeff Mangum](#):

```
| (mangum, neutral milk)
```

You can get all songs that don't match the search term using `!`:

```
!electro
```

Lets say you want to listen to you whole library but are not in the mood for classical music or songs by [The Smiths](#):

```
!| (classical, smiths)
```

While these searches are easy to type in, they depend on the visible columns and the active browser, also the last one might exclude some songs which happen to contain “smiths” in their album title - see below for how to perform more targeted searching.

### 5.3.2 Searching a Specific Tag

To search a specific tag, use a search like:

```
artist = delerium
album = bargainville
```



The search terms can't use quotes ("), slashes (/), hashes (#), pipes (|), ampersands (&), or bangs (!); these characters have special meanings for advanced searches. See *exact matching* or *regular expressions* for how to do searches including these characters.

In QL 3.9 onwards, you can also use != to search for things not equal:

```
artist != delerium
genre != /.+ Jazz/
```

You can also search *internal tags*, e.g.

- ~format = Ogg Vorbis
- ~dirname=Greatest Hits - search for all songs in Greatest Hits folders.

It's also possible to search in multiple tags at once:

- artist, performer = "Boa" c

### 5.3.3 Exact Matching

If you want an exact match, use quotes:

```
artist = "a girl called eddy"
```

If you need to put a " inside the quotes, you can put a \ before it:

```
version = "12\" mix"
```

Other special characters can be used without escaping:

```
title = "Concertos #1"
```

You can put a c after the last " to make the search case-sensitive:

```
artist = "BoA" c
artist = "Boa" c
artist != "Boa" c
```

### 5.3.4 Combining Tag Searches

As with free-text searches, you can combine searches using & ("and") and | ("or"); either grouping entire searches, or just the tag values. Although the examples below use simple keywords, you can also use exact matches or regular expressions:

```
artist = |(Townshend, Who)
&(artist = Lindsay Smith, album = Vat)
```

The first finds anything by **The Who** or guitarist **Pete Townshend**. The second gives the songs that match both, so you'll find songs Lindsay Smith's **Tales From The Fruitbat Vat**, but not her other albums.

You can also pick out all the songs that *don't* match the terms you give, using !:

```
genre = !Audiobook
```

is probably a good idea when playing your whole library on shuffle. Note again that in QL 3.9 onwards you can use the alternative syntax of:

```
genre != Audiobook
```

More complex searches are of course possible. For example, to select all Disco and Jazz related (`_containing_`, technically) genres, but avoiding Acid Jazz, you could use:

```
genre = &(|(Disco, Jazz), !Acid Jazz)
```

### 5.3.5 Numeric Searches

Using `#`, you can search your library using numeric values. Quod Libet keeps some internal numeric values including `track`, `disc`, `rating`, `length` etc. See [Numeric Tags](#) for full details. You can also search any other tag as long as the values have a number format like 1234 or -42.42, for example `year` or `bpm`.

For comparisons you can then use typical binary operators like `=`, `<`, `>`, `<=`, `>=` and `!=`.

- `#(skipcount > 100)` could find really unpopular songs, or
- `#(track > 50)` to figure out who makes really insane albums, or
- `#(bpm > 160)` to find really fast songs

**You can also use chained comparisons:**

- `#(10 <= track < 100)` to find all two-digit tracks.

Times like added are stored in seconds, which is pretty cumbersome to search on. Instead, you can search with semi-English, like:

- `#(added < 1 day)` for very recently added tracks

to find songs added in the last day (if you think that that's backwards, mentally add 'ago' when you read it). Quod Libet knows about seconds, minutes, hours, days, months (30 days), and years (365 days), kB (Kilobyte), MB (Megabyte), GB (Gigabyte). You can also use "HH:MM" notation, like:

- `#(2:00 < length < 3:00)` for songs between two and three minutes long.

Of course, you can combine numeric with other kinds of searches.

- `&(genre = classical, #(lastplayed > 3 days))`
- `&(artist = "Rush", #(year <= 1996))`

### 5.3.6 Playlists

You can use the `~playlists` internal tag to search by playlists. It is populated with a list of all the playlists that song appears in. This is surprisingly powerful if you're a playlist user.

- `~playlists=chilled` will return all songs included in any playlist with "chilled" in its name.
- `~playlists=|("Chilled", "Jazzy")` for all songs in either (or both) of those playlists.
- `&(#(rating>=0.75), ~playlists="")` will return all high-rated songs *not* in any playlist

### 5.3.7 Regular Expressions

Quod Libet also supports searching your library using "regular expressions", a common way of finding text for Unix applications. Regular expressions look like regular searches, except they use `/` instead of `"`, and some punctuation has special meaning. There are many good tutorials on the web, and useful online regex testers (such as [Regex Pal](#))

Some examples:

```
artist = !/\sRice/ (or in 3.9+: artist != /\sRice/)
```

or using the default tags:

```
/^portis/
```

Like with exact matches, append a *c* to make the search case-sensitive:

```
/Boa/c
```

Regex searches can also be used to escape special characters not permitted in normal searches:

```
filename = /\Music\Alternative/
```

### 5.3.8 Ignore Accents and Umlauts

Appending a *d* after searches makes it's characters match variants with accents, umlauts etc.

Both `/Sigur Ros/d` and `"Sigur Ros"d` will match songs with the artist name "Sigur Rós".

Now you can search anything!

### 5.3.9 Pluggable query expressions

More recent versions of Quod Libet allow queries to include pluggable expressions. This uses the format `@(plugin: body)` where *plugin* is the ID of the query plugin (e.g. `saved`, `python`) - see `quodlibet/ext/query/`.

From QL 3.10 onwards, the Python query plugin allows some use of external modules, notably *time* and *random*, as well as a few useful variables.

For example, here is a way of simulating an album Spotlight in an album browser:

```
@(python: Random((int(_ts / 60), a)).random() < 0.01)
```

Here `_ts` is a current timestamp, and `a` is the album data. So this generates a random number seeded on the current minute and the album key, so we randomly select 1% of our albums to look at.

### 5.3.10 Reusing queries

Complex queries can be split into simpler ones. Also, a query can be reused in other ones. This way it is easier to change and administer your searches.

In order to do so, the `Include Saved Search` query plugin (see above) must be activated. If you create a saved search named `Unrated` you can search for unrated songs from the Beatles like this:

```
&(@(saved: Unrated), Beatles)
```

For creating saved searches, use the “Edit saved searches...” item in the drop-down at the right of the query text box.

## 5.4 Library Statistics

### 5.4.1 Play Count

The internal `~#playcount` tag gets incremented after a song ends or is forced to end by the user and the time it was played was more than half of the song's duration.

In case of radio streams, which don't have a defined duration, the play count gets incremented whenever the stream is played.

Searching for all songs that where played more than 10 times:

```
#(playcount > 10)
```

### 5.4.2 Last Played Time

The `~#lastplayed` tag gets updated to the current time whenever `~#playcount` gets incremented.

Searching for all songs played less than 4 days ago:

```
#(lastplayed < 4 days)
```

### 5.4.3 Last Started Time

The `~#laststarted` tag gets updated to the current time whenever the song gets started.

Searching for all songs started less than 1 week ago:

```
#(laststarted < 1 week)
```

### 5.4.4 Skip Count

The `~#skipcount` tag gets incremented whenever the song gets forced to end by the user and the playing time was less than half of the song's duration.

Searching for songs that where skipped between 5 and 10 times:

```
#(5 <= skipcount <= 10)
```

## 5.5 Browsing Your Library

### 5.5.1 Overview

Quod Libet has different ways to browse your library, called *Browsers*, which are selectable from the *View* menu. There is always one (the *primary*) browser active.

#### Secondary Browsers

You can also browse your library in a separate window (without disturbing your current playlist) by selecting one of the options from *Browse* → *Open Browser*. You can have multiple of these open at once, and they will all react to changes to your library underneath.

## The Song List

The *Song List*, as the name implies, presents a list of all the songs that the current browser has found, or filtered for you. The columns are configurable, and can generally be any tag, or even combinations of tags from your library. For more information on tags, see [An Introduction to Tags](#).

Sorting:

- List columns sorted by “disc” and “track” are actually sorted by “album”
- All songs are sorted by the column header tag and with a special sort key. If there is something wrong with the sort order check the tags used in the sort key: “*albums*sort or *album*, *album\_grouping\_key* or *labelid* or *musicbrainz\_albumid*, *~#disc*, *~#track*, *artistsort* or *artist*, *musicbrainz\_artistid*, *title*, *~filename*” (see [An Introduction to Tags](#))

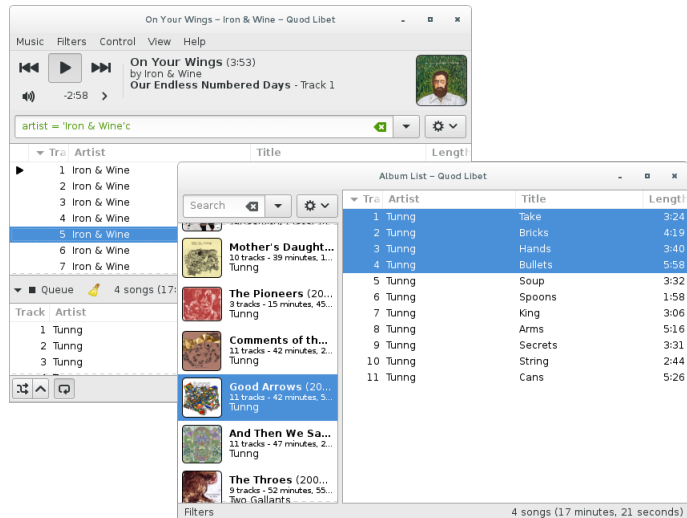


Fig. 2: Multiple Browsers - Search & secondary Album List

## Filters

**Filters allow you to remove all but a subset of songs that from a browser’s song list, typically based on a tag.**

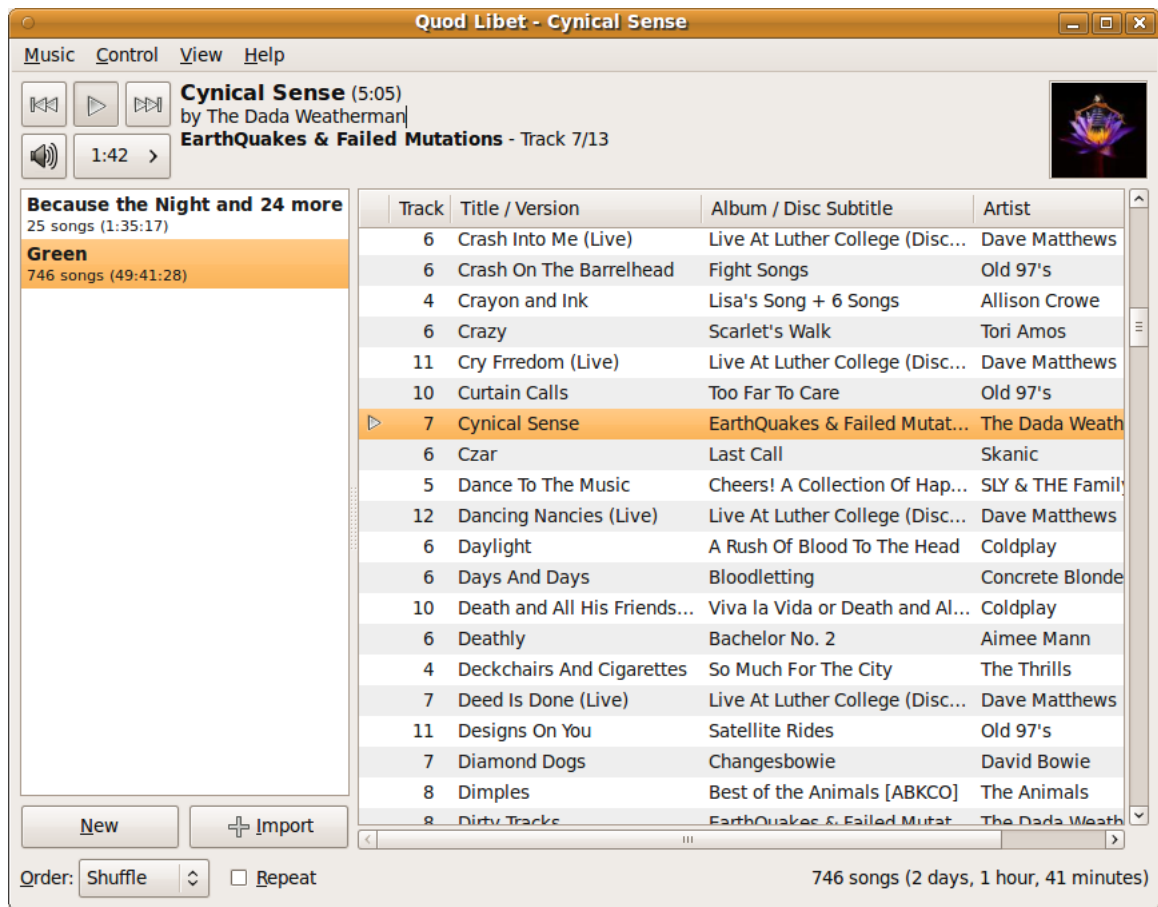
Different browsers implement these accordingly; in the Search browser, these become searches e.g. filtering by Artist might produce a search `artist='Beethoven'`

Some filters are not available on all browsers. For example, the Search Bar can filter by anything, but there’s no way to get a “top 40” in the Album List.

Note also that when using the song context menu (a.k.a. “songsmenu”), QL notices which *column* the mouse is in when you right-click on a song selection, and will offer this column as a quick filter too, if possible.

## 5.5.2 Playlists

## The Playlist Browser



Choose the *Playlists* browser by clicking on *View -> Playlists*. The usage is fairly simple - a list of songs in the right pane, and a list of playlists (with summary information) in the left.

Playlists are stored as plain text files on disk typically under `~/ .quodlibet/playlists`. The names are URL-encoded. Whilst these can be edited, it's recommended to leave it to Quod Libet, but it's good to back them up (using `svn / hg / git` works well here too).

Any file in your library can belong to any playlist or many playlists - it's up to you how you want to organize them...

## Creating playlists

There are several ways to create playlists in Quod Libet. Choose whichever suits you best:

- To create a blank playlist, select the Playlist Browser and click *New*.
- To create a new playlist with songs in it, select the songs in any other browser, right click, and select *Add to Playlist -> New Playlist*.
- To add songs to an existing playlist, either use *Add to Playlist* in the right click menu, or drag them to the playlist name on the sidebar.
- To import playlists from *pls* or *m3u/m3u8* files, use the *Import* button. In addition to creating the playlist, any files in it will be added to your library.

## Context menu support

The “songs menu”, a context menu presented when you right-click on a song (or songs), has in-built support for playlists. Just right-click on a song, and select *Add to Playlist*. This is a convenient way of adding a song to a playlist from almost *any* browser, and even better, seeing *every* playlist that song features in (they will be ticked).

## Library Changes

The Playlists Browser, like many other browsers, listens to changes in your library. This means that any changes in tags will be reflected in the song list for each. Playlist entries are indexed by file path though, so *any changes of name or directory will remove a song from its playlists* (There are, however, discussions to change this: Issue 708). Please be careful with mounted media (e.g. USB / network disks) as when the library is rescanned, these files not existing is taken as deletion, which will provoke a removal from their playlists. Please keep backups if playlists are important.

## Drag and Drop

Quod Libet has extensive drag-and-drop support in the playlists browser. You can drag and drop songs from other browsers (eg the search browser) onto an existing playlist, or songs from one playlist to another. When you drop a song into the left-hand pane in the playlist browser, but not onto a playlist, a new playlist is created, named after that song.

## Importing and Exporting Playlists

As outlined above, you can import playlists from *pls* or *m3u/m3u8* files using the *\_Import\_* button. In addition to creating the playlist, any files in it will be added to your library. You can also drag and drop an *m3u* playlist file from an external browser onto the left-hand pane in the playlist browser to import a playlist.

To export playlists to *m3u*, you first need to install the Export playlist export plugin. Once installed, you can export playlists to M3U or PLS format by right clicking on the playlist, then *Plugins -> Export Playlist*.

## Dynamic Playlists

You may wonder whether Quod Libet doesn’t have so-called “dynamic” or “smart” playlists as you may know them from other music players, that is, playlists that automatically update themselves for example to always contain all songs from a certain artist that you have in your library.

In fact, QL does have this functionality, but it is implemented via the search functionality and hence located in the search browser, not in the playlists browser.

Creating a dynamic playlist:

- Go to the search browser, (either using *View* or *Music -> Browse Library*).
- Enter what you want to search for (see [the section about searching](#) for QL’s powerful search options), for example *artist = radiohead*. Optionally, click on search to test your search and modify it until you’re happy.
- Click the arrow next to the search box on the right, to open the drop-down menu. You will see a history of some recent searches (if you have searched before), followed by *Edit saved values... .* Clicking on this, you will be presented with a dialogue box. The *Value* field has been pre-filled with your current search. If you wish, enter a name in the *Name* field (if you leave it blank, QL will name it for you). Click *Add*.

You have now created a dynamic playlist, via a saved search. To play, just go to the search browser, click on the arrow to the right of the field and select your saved search from the list.

If you want to create several of such saved searches at once, you may find it more convenient to edit a text file instead of clicking through the GUI. To do so, you can *edit the configuration text file* `~/.quodlibet/lists/queries.saved`.

### 5.5.3 Search Browser

The *Search Library* browser (which is the default one) lets you enter search terms and play all matching songs. It also supports *complex searches*. Finally, it lets you limit the number of results; right-click on the text bar and select *Limit Results*. To display your whole library, don't enter any search terms.

The browser remembers the last eight searches you made automatically. If you want to save more, you can right-click on the text entry and select *Edit Saved Values...*, which will let you name and save searches permanently. Saved searches will appear above automatically remembered ones in the menu.

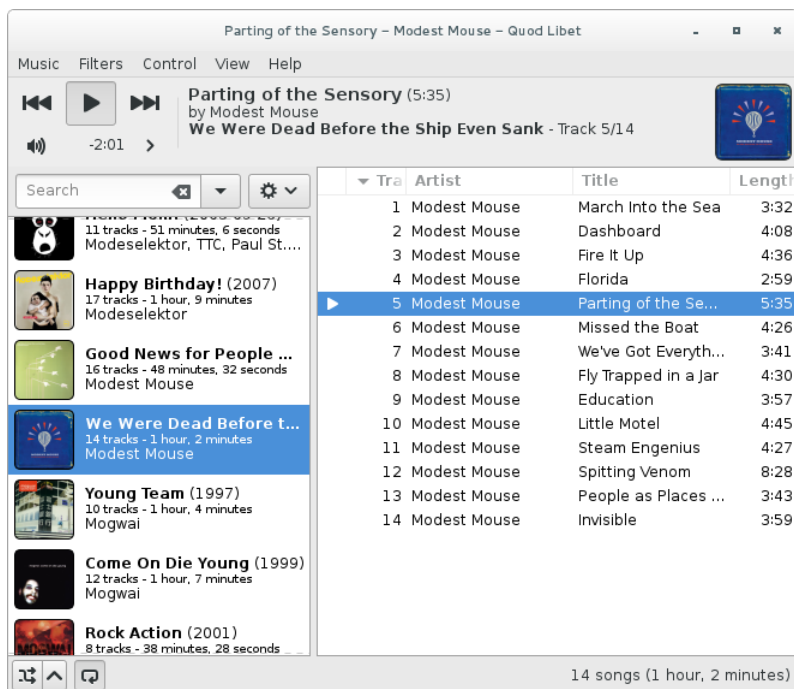
### 5.5.4 Album Browsers

#### Identifying albums

Throughout Quod Libet, albums are grouped by examining the `album` tag on individual songs. To support distinct albums with the same name, groups of songs with the same 'album' tag are further inspected for the presence of one of three tags: `album_grouping_key`, `labelid`, and `musicbrainz_albumid`, in that order. If a song has any of these, it will be used to identify the album it belongs to along with the `album` tag.

In short, if more than one album in your library has the same name, use one of the secondary tags to separate them. The MusicBrainz plugin will add `musicbrainz_albumid` tags automatically, and may be the easiest solution for adding identically-named albums to your library.

#### Album List Browser



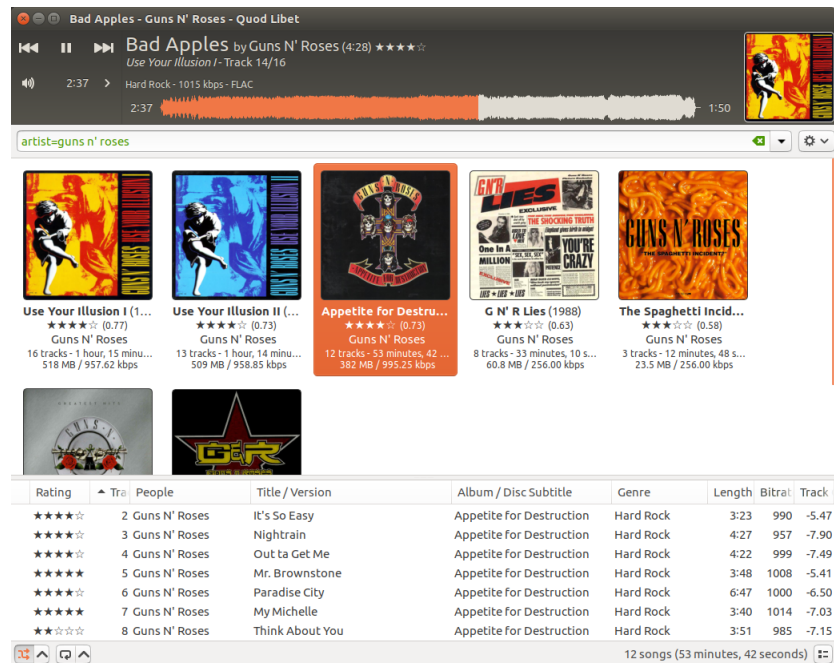


The *Album List* browser makes it easy to treat your music collection as a set of albums, presented on the left, rather than as a set of songs, via album-centric enhancements to viewing, sorting, and searching.

You can (configurably) display the album art next to each album to allow faster identification of your albums (plus *it's just prettier...*).

Extra features related to searching, sorting and presenting albums are detailed below.

## Cover Grid Browser



The cover grid is another album-like browser that places more emphasis on the album art.

It has a search similar to album list (see the section below), and configuration of its own (including a flexible per-album display area).

## Searching

After creating the list of albums using the heuristic described above, the Album List browser then computes information across all the songs in an album. While the browser uses sensible defaults as to exactly how this information is compiled, it also exposes the choice while searching. This is most useful with numeric searches. For example, to find the albums with an *average* rating of 0.6 or greater, you can search for

```
#(rating >= .6)
```

This works because the Album List averages the values of numeric tags by default. To find the albums with *any song* with a rating of .6 or greater, though, you have to add something to your search:

```
#(rating:max >= .6)
```

These tag suffixes work for any numeric search. The options are `min`, `max`, `sum`, and `avg` (the default).

For string tags, the values which get searched are created by joining all of the underlying songs' values together. The albums in an Album List also have a few tags which are computed in a particular manner. A few of the interesting ones:

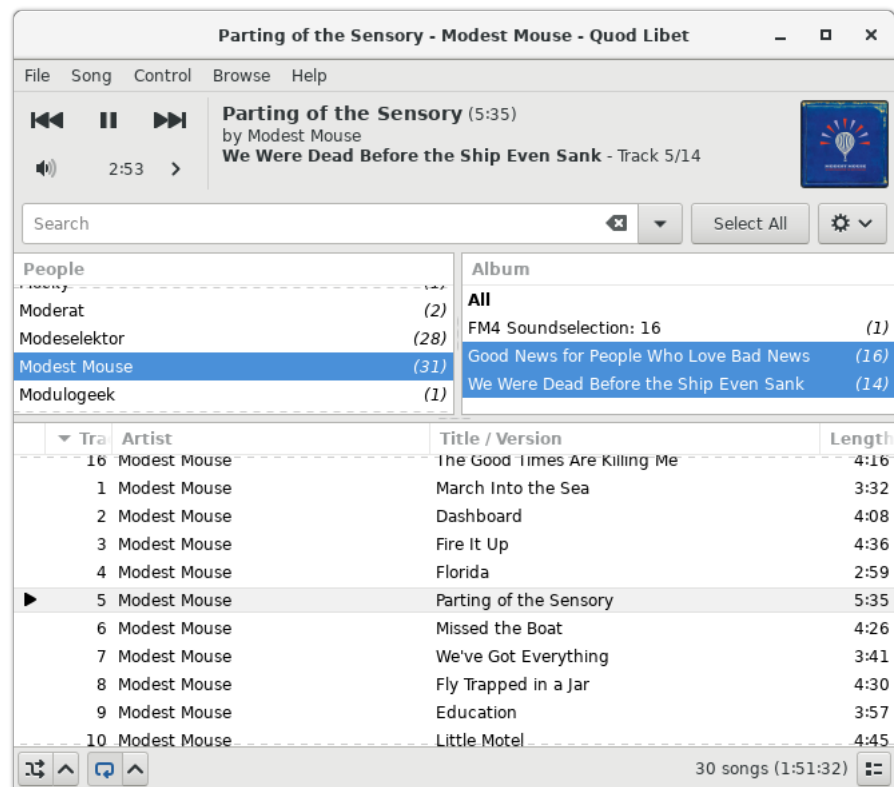
- `~#length` is computed as the sum of the length of the underlying songs.
- `~#tracks` and `~#discs` are the total number of songs and discs in an album.

Also useful, though not strictly album-only:

- `~#filesize` is the size on disk of a file in bytes (but can be formatted for humans)
- `~people` is computed from all underlying tracks, with duplicate entries removed.

## 5.5.5 Paned Browser

### Overview



The Paned Browser offers a convenient way to quickly drill down into a large music collection, by narrowing selections in several stages. Some users may find this reminiscent of [RhythmBox](#) or, to some extent, [iTunes](#).

In Quod Libet though, you can have as many panes as you want, grouped by any tags you want, for example the popular “genre/artist/album” and “artist/album”, or “artist/album/part”, or “artist/album/artist” in case you have a lot of multi-artist albums.

The songlist is presented at the bottom, and the panes, which run from left to right, are above. Clicking on an item (or items) in a pane will restrict it to just songs matching those (e.g. those artists, or dates, genres etc). This will update the counts and choices on the next pane, and the filtered results will be updated automatically in the song list.

### Pane Configuration

To change the panes, click the *Preferences* button all the way to the right of the search bar. There you can choose between some popular setups or set up custom ones using the add and remove buttons. You can change the order of the panes by dragging them to the desired place.

Unlike elsewhere, multiple values per tag for a song are split into multiple entries. Tied tags also result in multiple entries. Tag patterns take multiple values from tags and from tied tags and produce multiple entries.

Entries are sorted using sort values for tags. For tags with multiple values, each value is paired with the corresponding value from the sort tag. Quod Libet tries to do something reasonable when a value (e.g., “The Beatles”) sometimes has a sort value (e.g., “Beatles, The”) and sometimes does not or has a different sort values (e.g., “Beatles”).

## Examples

Pattern	Result
~performers	Julio Inglesias
	Frank Sinatra

Pattern	Result
<~year <~year>. <title> <title>>	1993. Summer Wind

Pattern	Result
~title~performers	Julio Inglesias
	Frank Sinatra
	Summer Wind

Pattern	Result
<~year <~performers> - <~year> <~performers>>	Julio Inglesias - 1993
	Frank Sinatra - 1993

## Using Markup

Also it's possible to change text emphasis using the [Pango markup language](#)

Pattern	Result
<~year \<b>\<i><~year>\</i>\</b> - <album> <album>>	<b>2011</b> - This is an album title

## Aggregation

On the right side of each pane you can see the number of songs of each entry. This can be configured as well by adding a pattern/tag separated by : (In case you want to use : in your pattern it has to be escaped using \:)

Pattern	Result
~year~album:(~rating>)	2011 - Album title (♪♪)

## 5.5.6 File System Browser

This browser lets you view songs based on the folder they're in. It can play and edit songs inside and outside of your song library, and adds an item to the context menu to add the selected songs to your library. If you try to add songs to a playlist or the play queue that aren't in your library, they will be added automatically to it.

---

**Note:** Since ratings and play counts are usually stored in the library rather than the song, play counts and ratings will only be saved if you add the song to your library!

---

### 5.5.7 Internet Radio Browser

This browser lets you listen to Internet radio stations (a.k.a. streaming audio or Shoutcast). It supports MP3 and Ogg Vorbis streaming, and may support other formats (AAC or RealMedia) if you have appropriate GStreamer plugins installed. *New Station* accepts either a direct URL to a stream, or a URL to a .pls file with a list of streams.

Radio stations cannot be added to playlists or the play queue. You can edit the `title`, `artist`, and `grouping` tags, but the rest are filled in by the station when you listen.

If you don't know any good stations, the *Stations...* button will let you select some. If you do know some good stations, you can add them to the Stations list.

### 5.5.8 Audio Feeds Browser

The *Audio Feeds* browser allows you to subscribe to syndicated feeds with attached audio files; these are often called “podcasts” or “blogcasts.” Feeds are automatically checked for updates every two hours, and emboldened if new entries are found.

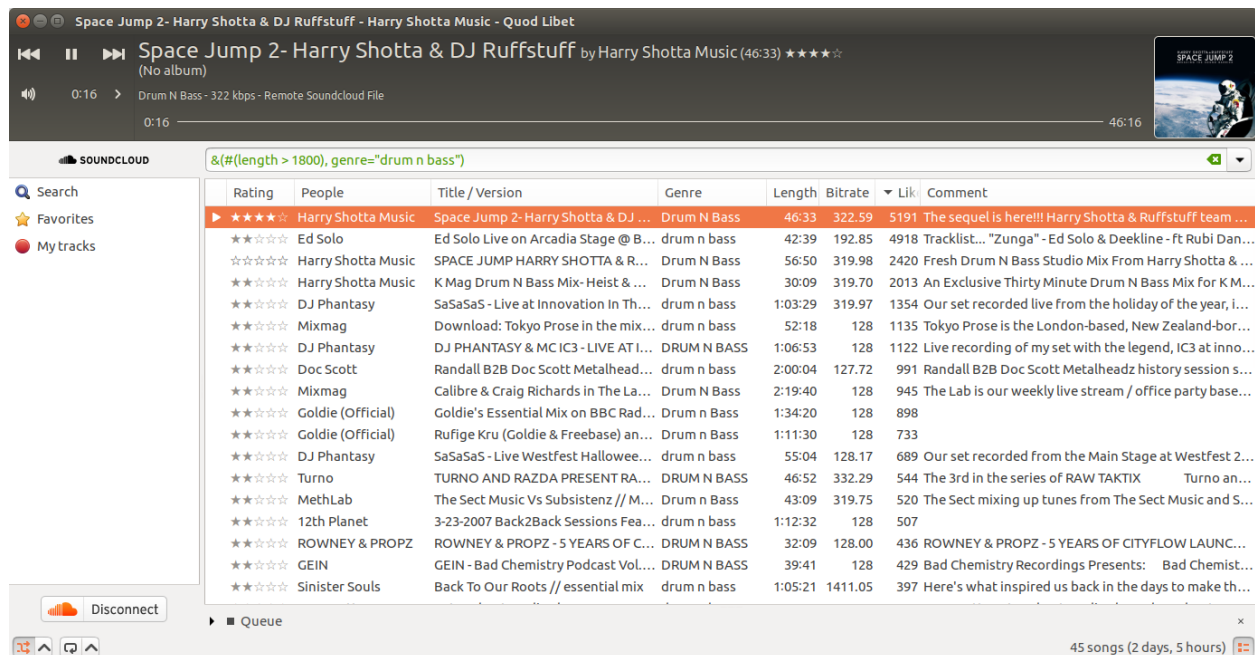
You can right-click on a file in an audio feed, and select *Download*, to download it to your hard drive. Currently this has no effect on the feed itself (so changes to the local song will not be reflected in the feed list).

### 5.5.9 Soundcloud Browser

*From v3.7*

#### Overview

This browser lets you interact with [Soundcloud](#) and play tracks from the millions on offer. Where possible it aims to keep the look and feel as familiar and integrated to the Quod Libet experience as possible.



## Connecting your account

If you have a Soundcloud account, you can also:

- access your favorites
- rate (like / unlike) songs.
- list your own tracks (QL 3.10+)

To do so, click the Soundcloud connect button at the bottom right.

This will then take you to. If your operating system is configured to process `quodlibet://` URLs with Quod Libet (see [the instructions](#) given to you there) then this process will happen automatically. If not, you can enter the code manually using the same button.

If not, you can copy the code from the web page that appears and click the QL button again to enter it.

Once you are logged, in you can log out by clicking the same button again, now a disconnect button (seen in the screenshot).

## Features

### Higher quality streams

The Soundcloud browser will use the download URL, where available, for the highest quality stream. Note this may require that you are logged in, and is usually *not* available.

### Support for Quod Libet queries

One of the more interesting features of the soundcloud browser is that it can *translate* simpler queries in standard QL syntax to something that can work across the web to a library that isn't yours.

Obviously, there are many limitations to this approach both conceptually and due to its implementation, but nonetheless queries like `&(#(length>300), title=dubstep)` get (roughly) what you might expect.

### Favorites

These are supported as `~rating` (see below), and unrating a track will unfavorite it in Soundcloud, if you are logged in.

### Tags

#### Supported tags

- `artist`, `genre`, and `title` translate as you might imagine
- `website` translates to Soundcloud's current URL for that track
- `genre` holds the genres (plural) tagged in Soundcloud (these are rarely useful but YMMV)
- `date` is the *creation* date in Soundcloud
- `~#rating` is translated to 0.0 (not a Soundcloud favourite) or 1.0 (a Soundcloud favourite)
- `~#bitrate` is the highest bitrate available when playing the track (normally 128)
- `~comments` translates to the track details in Soundcloud.
- `bpm` is the Soundcloud BPM if provided.

#### New tags

- `~#favoritings_count` and `~#likes_count` etc represent how often these tracks have been favorited / liked (note that there currently seem to be inconsistencies within Soundcloud itself as to how these are populated)
- `soundcloud_track_id` is the Soundcloud ID for the track.
- `soundcloud_user_id` is the Soundcloud ID for the user.
- `artwork_url` is the (remote) image URL for the track artwork on Soundcloud. This works well with the corresponding `Artwork URL Cover Source` plugin.

### Comments

Comments in Soundcloud have been integrated to QL as read-only bookmarks, as they are also time-specific text to do with a particular song. Unlike bookmarks they have a lot more metadata (notably the user), so this is rendered as text.

To see them, access as you normally would bookmarks (e.g. view info, right-click the time widget or edit the bookmarks).

## 5.6 The Queue

As described in the [Overview](#), the queue is used to queue up songs to be played. In recent versions of Quod Libet, some new features have been introduced to the queue, which is described below.

### 5.6.1 Ignore and Keep Songs

With [release 4.2.0](#), the queue was updated with two new checkboxes: *Ignore* and *Keep Songs*.

When the *Keep Songs* checkbox is checked, songs won't be consumed from the queue anymore when played. In this way, the queue can function as a kind of temporary playlist.

With the *Ignore* checkbox checked, the queue won't have priority anymore. This means that if a song is played from the song list, then songs from the song list will continue playing even if the queue is not empty.

If both *Keep Songs* and *Ignore* are enabled, songs will be kept in the queue while one at the same time is free to switch between playing songs from the queue and the song list. If a song in the queue is selected, songs will keep playing from the queue, while if a song in the song list is selected, then songs from the song list will be played instead.

### 5.6.2 Disable and Mode Selection

With [release 4.3.0](#), the *Ignore* and *Keep Songs* checkboxes were replaced with a *queue disable* button and a *mode selection* menu.

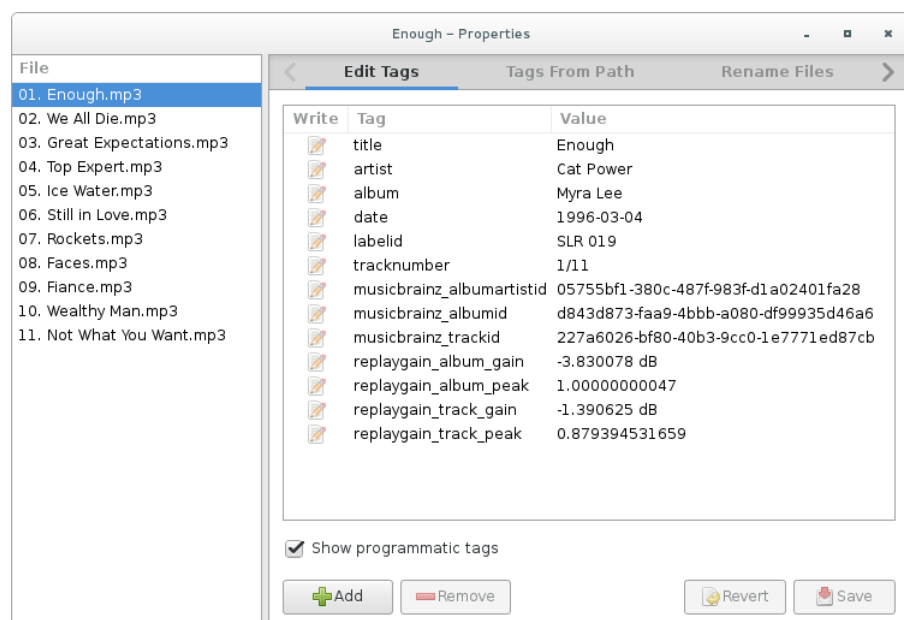
The queue can be disabled by clicking the padlock in the queue header. When disabled, the queue cannot be played from - but you can still add songs to it.

The queue also has two different modes, that can be changed in the queue preferences.

In the *Ephemeral* mode, the queue consumes songs as described in the [Overview](#). This is the default behavior.

The other mode is the *Persistent* mode. In this mode, songs will be kept in the queue even after being played. Furthermore, the queue won't have precedence over the song list unless it is explicitly being played from by first double-clicking on a song in it. This allows switching between the queue and the song list without having to clear or disable the queue.

## 5.7 Editing Tags



You can edit a song's tags by right clicking on it and selecting *Edit Tags*.

In addition to manually entering the tags, if the song's filename contains information about all or some of the tags, you can use the *Edit tags from path* tab to populate these tags automatically. Please see [editing tags from path](#) below describing the process for several songs (the process is the same).

### 5.7.1 Editing tags for several songs at once

To edit tags for several songs at once, select those songs (using Ctrl or Shift), then right click and “Edit Tags”.

Note that in the tag-editing window that opens, you have several tabs available. The default *Edit tags* tab will apply the same tags to all marked songs, so it only makes sense for tags that are common to all songs, e.g. album or genre. However, the tabs *Tags from Path* and *Track Numbers* let you edit tags that differ across songs:

#### Editing tags from path

The tab *Tags from Path* lets you batch edit tags that differ across songs, such as title, by using the filename as input. Note that you can customize the pattern that the tag editor uses to extract the tags from the filename: just imitate the pattern you see for your files, putting the relevant tag name in angular brackets.

Example:

- Your file names have a pattern like this: 01 - The Beatles - Yellow Submarine.ogg
- Edit the pattern to show: <tracknumber> - <artist> - <title>
- Note that you can omit the file extension in your pattern.
- Click on *Preview* to see how your pattern would be interpreted for each song.

The preview is shown to the right of the current value; you may have to scroll right to see it.

You can even include information from the entire path in this pattern matching:

- You have files like this ~/home/username/music/favourites/the\_beatles/yellow\_submarine/01 - Yellow Submarine.ogg
- Use pattern: <artist>/<album>/<tracknumber> - <title>
- In that case, you probably want to check the boxes for *Replace underscores with spaces* and *Title-case tags*.
- Note that QL automatically digs as far upwards in the folder hierarchy as it needs to given the pattern you put in, so you don't need to enter any (potentially complex) folder structure that is above the needed info.

You can see recent patterns you used by clicking on the drop down arrow to the right of the pattern input field. Additionally, clicking on *Edit saved values* in the drop-down that opens will let you save patterns and optionally name them. Use this for patterns that you apply frequently. If you leave the *Name* field blank for your pattern, the name will be identical to the pattern.

#### Batch edit track numbers

The *Track Numbers* tab in the tag editing window lets you batch edit track numbers ascending across the files. If your files are in the correct order, you simply check that you like the *Start from* and *Total tracks* values. If you put in any value greater than one for *Total tracks*, QL will use a tracknumber pattern *tracknumber/totaltracks*, e.g. 2/12 for *Total tracks* = 12. If you only want a single number for the track number, set *Total tracks* to zero.

If your files are not in the correct order, for example because they are sorted alphabetically, you can drag and drop them into the desired order in the *File* field inside the *Track numbers* tab of the tag editing window before (optionally) clicking preview and then save.



## 5.7.2 Rename Files Based on Tags

QL also lets you rename the files of songs based on tags, either for one song or for several songs. Edit patterns the same way you would for *Edit tags from path* (see above). This feature even lets you move them to a different directory; for more info see the [renaming files guide](#).

## 5.7.3 Splitting Tags

If a tag contains a value that can be regarded as multiple tag values, it is often possible to split the tag. This can be done by right-clicking on the tag and then selecting the appropriate split in the `Split Tag` menu.

There are in general two types of tag splitting possible: splitting on a single character and *subtag* splitting (extracting values in enclosures). The separating characters for both can be configured in the *Tags* tab in the preferences.

Splitting on a single character - like `,` or `&` - will split a tag into multiple tags of the same type, but with different values. An example of this can be an artist tag with the value `Foo, Bar`, that can be split into two separate tags *artist* = `Foo` and *artist* = `Bar`.

With *subtag* splitting, the end of the tag value must contain a value enclosed in some pair of characters - like `()` or `[]`. Depending on the type of tag, the enclosed value can then be extracted and put in a new tag. An example of this can be an album tag with the value `Album (CD 1)`, which can be split into *album* = `Album` and *discnumber* = `1`. The enclosure can also contain multiple values separated by a single-character separator as explained above, like *artist* = `Foo (Bar, Baz)`, which can be split into *artist* = `Foo`, *performer* = `Bar` and *performer* = `Baz`.

## 5.8 Renaming files

### 5.8.1 Basic Syntax

Quod Libet allows you to rename files based on their tags. In some cases you may wish to alter the filename depending on whether some tags are present or missing, in addition to their values. A common pattern might be:

```
<tracknumber>.<title~version>
```

You can use a `|` to only insert text when a tag is present:

```
<tracknumber|<tracknumber>.><title~version>
```

You can also specify literal text to use if the tag is missing by adding another `|`:

```
<album|<album>|No Album> - <title>
```

A reasonable use of `albumartist` would be:

```
<albumartist|<albumartist>|<artist|<artist>|No artist>>
```

..which uses the first of the following: `Albumartist`, `Artist` or “No artist”.

You can of course also move files across your filesystem to another directory by mixing path elements and `<tag>` syntax:

```
/home/*username*/Music/<artist>/<album>/...
```

## 5.8.2 Simple Renames

Like tagging by filename, renaming by tags uses tag names enclosed by `< . . >` to substitute values. To rename songs as their artist followed by their title, use `<artist> - <title>` (The file extension, `.ogg`, `.mpc`, and so on, is automatically added). Other common patterns include

- `<tracknumber>. <title>`
- `<tracknumber>. <artist> - <title>`
- `~/music/<artist> - <album>/<tracknumber>. <title>`
- `~/music/<artist>/<album>/<tracknumber>. <title>`

You can also use tied tags to rename, e.g. `<artist~title>`.

## 5.8.3 Creating Directories

Note that if you use `/` (a directory separator) in your filename, you “must” start the pattern with a `/` (or a `~/`, which expands to your home directory). To see why, consider what would happen if you tried to rename the same file twice with `<artist>/<title>`. The first time it would go under `Artist/Title.ogg`, the second time, `Artist/Artist/Title.ogg`. When you specify the full path, this can’t happen.

If you don’t use a `/` in the pattern, the file gets put in the same directory.

## 5.8.4 Conditional Renaming

Consider the `<tracknumber>. <title>` pattern.

When the file is missing a track number, you get a filename that starts with `..`, which isn’t good. So Quod Libet lets you use “conditional renaming” to avoid that.

To use conditional text, after the tag name (but inside the `< . . >`) put a `|` (a pipe). Then after the pipe, place all the text you want, including other tag names inside `< . . >`. That text will only be added when that tag isn’t empty.

To avoid the original problem, only display the track number, period, and space when the track number tag exists:

```
<tracknumber|<tracknumber>. ><title>.
```

Quod Libet also lets you change the text if a tag “doesn’t” exist: Use a second pipe. `<tracknumber|<tracknumber>|00>. <title>` will use the track number at the start of the filename if it exists, or `00` if it doesn’t.

## 5.8.5 Go crazy with conditions / More examples

So you basically want to remember that it goes `<condition|<conditional tag>|<else tag>>` You can however even put conditions inside each other. Here’s an example that I just created, and so far it seems to work:

I first had:

```
/mnt/musik/<genre|<genre>/><artist|<artist>|Unknown>/<album|<album>/><tracknumber|
↪<tracknumber> - ><title>
```

Let’s dissect this:

- `/mnt/musik: My basic music partition`

- `<genre|<genre>/>`: If there is a tag “genre”, put the song into that folder (creating the folder if necessary). If there is no tag genre, skip this level in the folder hierarchy (note that the trailing slash of `<genre>/` is inside the `<>` that delineate the conditional “block”.
- `<artist>|<artist>|Unknown>/`: If there’s a tag artist, put everything into that folder, else put into a folder called “Unknown”. Not that the trailing slash is outside the `<>` that delineate the conditional block, since we always want that folder level.
- `<album|<album>/>`: Album folder as needed, else skip
- `<tracknumber|<tracknumber> - >`: Prepend tracknumber if it exists
- `<title>`: Duh.

However, for songs that don’t have a genre tag, I wanted to use a tag I use called “language” and sort into that folder instead. But I have some songs that have a genre tag and also a language tag, and those songs should only go into the genre folder; the language folder should be ignored.

Turns out QL can do this just fine, by expanding the `<genre>` conditional block in the expression above to `<genre|<genre>/|<language|<language>/>>`.

Basically, the pipe after the second `<genre>/` introduces what should be done if the first condition isn’t met (i.e. no genre tag), but here instead of putting straightforward text or another label we then introduce a second conditional block, `<language|<language>/>>`, which puts in a language tag folder, if the song has a tag “language”.

The full expression now looks like this:

```
/mnt/musik/<genre|<genre>/|<language|<language>/>><artist|<artist>|Unknown>/<album|
↪<album>/><tracknumber|<tracknumber> - ><title>
```

## 5.9 Playback

### 5.9.1 Audio Backends

Quod Libet currently supports GStreamer and Xine as an audio backend. The default backend can be changed in `~/.quodlibet/config` by setting the backend option (`gstbe = GStreamer`, `xinebe = Xine`, `nullbe = no backend`). Make sure Quod Libet isn’t running while you edit the file.

#### GStreamer Backend

##### Custom Pipelines

It’s possible to attach a custom GStreamer pipeline to the player backend under *File* → *Preferences* → *Playback* → *Output Pipeline*. The pipeline syntax is equivalent to what is used in the `gst-launch` utility. See `man gst-launch` for further information and examples.

In case the custom pipeline doesn’t contain an audio sink, Quod Libet will add a default one for you.

##### Debugging Pipelines

In case you are interested in which GStreamer elements and audio formats are used in the current pipeline, start Quod Libet in debug mode (`quodlibet --debug`), go to *File* → *Preferences* → *Playback* and press the *Print Pipeline* button. It will print the whole pipeline used for the current active song to *stdout*.

For debugging GStreamer related issues see the official GStreamer docs: [Running and debugging GStreamer Applications](#)

## Gapless Playback

Gstreamer supports gapless playback for all common formats except MP3. See the following bug report for more information: [https://bugzilla.gnome.org/show\\_bug.cgi?id=620323](https://bugzilla.gnome.org/show_bug.cgi?id=620323)

## Selecting an Output Device

If you want QL to output to a different device you have to pass the device option to the sink by setting a custom pipeline. In case of pulseaudio you can get a list of available devices by executing:

```
#!/usr/bin/env python2
import gi
gi.require_version("Gst", "1.0")
from gi.repository import Gst

Gst.init(None)
dm = Gst.DeviceMonitor()
dm.start()
for device in dm.get_devices():
    if device.get_device_class() == "Audio/Sink":
        props = device.get_properties()
        element = device.create_element(None)
        type_name = element.get_factory().get_name()
        device_name = element.props.device
        print "%s device=%r" % (type_name, device_name)
dm.stop()
```

which should give you something like:

```
pulsesink device='alsa_output.pci-0000_00_1b.0.analog-stereo'
```

which you can use as is, as a custom pipeline.

## Xine Backend

The Xine backend needs either xine-lib 1.1.x or xine-lib 1.2.x. Since most distributions make QL only depend on GStreamer, you might have to install xine-lib manually (*libxine1*, *libxine2* in Debian/Ubuntu).

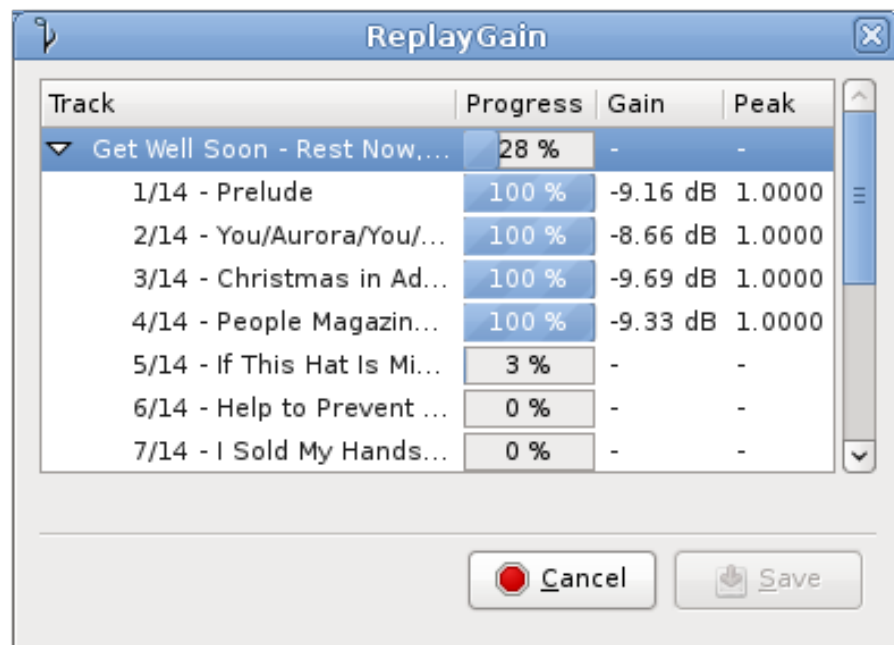
To enable the backend, set the `backend` option in the `config` file to `"xinebe"` while QL isn't running.

## 5.9.2 Replay Gain

**Replay Gain** is a common standard for adjusting the loudness of songs/albums to the same level.

Quod Libet provides tools for analyzing songs, saving the calculated values in the song's metadata and adjusting the volume.

## Audio Analysis



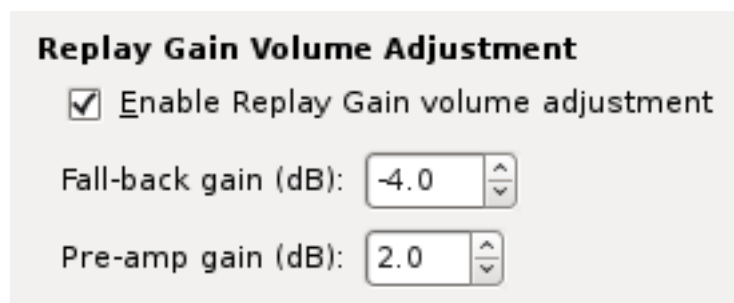
For calculating replay gain values QL provides a replaygain plugin. Simply select some songs, or some albums and activate the replaygain plugin in the plugin submenu. The replaygain plugin has to go through every bit of song data so this can take some time.

After the analyzing part is finished you can save the calculated values. They will be written into the song's metadata.

The replay gain plugin calculates a peak and a gain value for each song and for each album. The *gain value* specifies how much the volume of the song has to be adjusted so it equals the replay gain standard loudness level. The *peak value* holds the value of the loudest (or highest amplitude in fact) part in the song/album. This is needed because the range where the song volume can be increased is limited and pushing the volume over it would lead to sound distortion, so called “clipping”. The calculated gain adjustments are relative to 89 dB.

To find out more about how the calculation is done, read the [detailed summary](#) over at the hydrogenaudio's knowledgebase. QL uses GStreamer for this operation, so check out [rganalysis docs](#) for more info.

## Setting it up



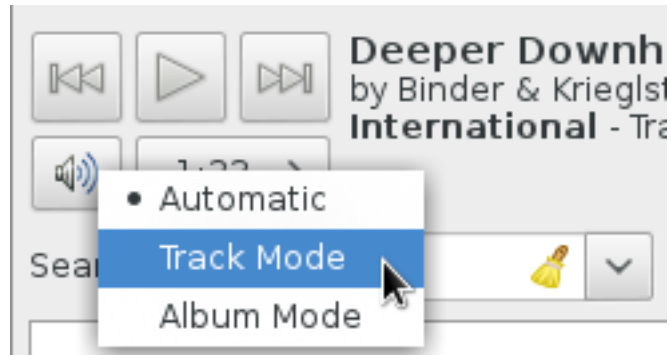
Under *Music > Preferences > Player* you can activate replay gain adjustment.

There are two configuration options available:

*Fall-back gain* is the volume adjustment that gets used for songs with no replay gain tags (which have not been analyzed). Most audio files are rather high volume these days and the difference between an 89db song and an unadjusted song can be quite high (e.g. >9dB). So to avoid a sudden volume jump, try setting this value to the average gain adjustment in your library.

*Pre-amp gain* simply gets added to each song's gain value, i.e. it adjusts the default loudness level (89 db). The problem is though, that if it is too high, Quod Libet can't increase the volume for some songs because they would exceed the volume limit and cause clipping.

### Applying volume adjustment



In Quod Libet the gain type is tied to the active browser and play order. The “Media Devices” and “Playlist” browsers will only use track gain. In all other browsers the “In Order” play order will switch on the album gain adjustment (it will fall back to track gain if there is no album gain). In “Shuffle” mode only the track gain gets applied.

It is possible to set the active gain type manually by right clicking on the volume slider button and choosing the type in the menu. The chosen gain type will then be active until Quod Libet is closed.

### Querying existing Replay Gain values

You can perform simple or complex searches (see [SearchingGuide](#)) using RG data. Here are a few useful ones:

Find *all files without track Replay Gain*:

```
replaygain_track_gain=""
```

Find *really loud albums* (best in AlbumListBrowser):

```
#(tracks>5, replaygain_album_gain<=-11)
```

## 5.10 Keyboard Shortcuts

The following lists describe various keyboard shortcuts and actions for their corresponding interface element. Many more can be seen in context menus accessible by right clicking on a specific interface element or by pressing Shift-F10 while the element is focused.

Primary stands for the Control key on Linux and Windows and the Command / key on OS X.

Alt stands for the Option / key on OS X.

### 5.10.1 Main Window

- Alt + Left - Seek backwards by 10 seconds
- Alt + Right - Seek forward by 10 seconds
- Primary + L - Focus the search entry

### 5.10.2 Browsers

- Primary + Shift + J - Reset filters and jump to the playing song.

### 5.10.3 Queue and Playlist Browser

- Primary + Left Click on song(s), then drag and drop - Add a copy of the songs to the playlist / queue.

### 5.10.4 Song Lists

- Primary + I - Open the information window for the selected song(s)
- Alt + Return - Open the tag editor for the selected song(s)
- Primary + Return - Queue the selected song(s)
- Primary + F - Show the inline search entry
- Primary + Left Click on a column header - Add the column to the list of columns to sort by.

### 5.10.5 Tree Views

- Left or Primary + Left - Collapses the element or select the parent element
- Right or Primary + Right - Expands the element

### 5.10.6 Text Entries

- Primary + Z - Undo the last change
- Primary + Shift + Z - Redo the last undone change

### 5.10.7 Paned Browser

- Primary + Home - Select all songs in all panes
- Primary + Return - When the search entry is in focus: enqueue all songs in the song list

## 5.11 Interacting with Quod Libet

### 5.11.1 In Python

Quod Libet supports *plugins written in Python*. These plugins can interact with songs more directly than the other interfaces, changing or reading metadata using dict-like `!AudioFile` objects. The plugin interface has not yet stabilized, although we do not expect it to change drastically in the future.

### 5.11.2 The Unix Way

#### Querying the player

Quod Libet writes information about the current song to `~/.quodlibet/current`. The file is in key=value form. Key values are in UTF-8, except for the `~filename` key, which has unknown encoding (but will point to the file being played when interpreted as a sequence of bytes).

There is a `--print-playing` option which can use the same syntax as the `RenamingFiles` interface to print information about the current song: `quodlibet --print-playing '<artist> - <tracknumber~title>'`

`quodlibet --status` provides player state information. The first word will be either *playing*, *paused*, or *not-running*, followed by

- the selected View,
- volume setting (0.0 - 1.0)
- Playback order & repeat settings
- Current song progress (0.0 - 1.0)

#### Controlling the player

Quod Libet understands a number of command line arguments to control a running player. For a full list, see the man page.

- `--next`, `--previous`, `--play`, and `--pause` should be self-explanatory; `--play-pause` toggles pause on and off.
- `--volume n` sets the volume to anywhere between 0 (muted) or 100 (full volume)
- `--seek <time>` seeks within the current song.
- `--query <search text>` searches in your library (if the current browser supports it).
- `--play-file <filename>` plays that file or directory.

Quod Libet can also be controlled via a `FIFO`, `~/.quodlibet/control`. To see how the command-line arguments map to FIFO commands, refer to `process_arguments()` in <https://github.com/quodlibet/quodlibet/blob/master/quodlibet/quodlibet/cli.py>; as a simple example:

```
# Sets volume to 50%
echo volume 50 > ~/.quodlibet/control
```



### 5.11.3 Integration with third party tools

#### Quod Libet in Conky

**Conky** is a lightweight system monitor for X. It includes builtin objects for many popular music players, but not quodlibet (yet). That doesn't mean you can't use conky with quodlibet. After installing conky, add the following to your `~/.conkyrc` file:

```
${if_existing /<path to your home directory>/.quodlibet/current}
${exec quodlibet --print-playing "<artist>"}
${scroll 50 ${exec quodlibet --print-playing "<title~album>"} }
${endif}
```

will display the current artist on one line with a scrolling display of song title and album on the next line. Conky will only attempt to display this information if quodlibet is playing.

#### eSpeak: Speech Synthesizer

You can use **eSpeak** to hear the current playing title.

```
quodlibet --print-playing "<~~people~title>" | espeak -s 120 -v $(quodlibet --print-
→playing "<language|<language>|en>")
```

In this example Quod Libet will use the value of the language tag to tell eSpeak which language/voice to use for the specific title (Use `espeak --voices`` to get a list of all available languages).

You can also lower the volume during speaking:

```
VOL=$(echo $(quodlibet --status | head -n1 | cut -d\ -f3)*100 | bc)
quodlibet --volume=$(echo $VOL/3 | bc)
quodlibet --print-playing "<~~people~title>" | espeak -s 120 -v $(quodlibet --print-
→playing "<language|<language>|en>")
quodlibet --volume=$(echo $VOL/1 | bc)
```

## 5.12 Configuration Files

QL stores some of its configuration as plain text files, and in some cases it may be convenient to edit these files directly, or to synchronize them across different computers/ home directories.

Like many Linux applications, QL stores user configuration in a hidden directory, in this case `~/.quodlibet`. (Note that this may not always be the case, see [the FAQ](#) for details). Feel free to explore; but maybe make a backup of the directory first.

### 5.12.1 Saved values for search, tagging and renaming patterns

For *searching your library*, *editing tags from a filename pattern*, and *renaming files based on tags*, you can save the (search or pattern) values you enter for later use. See the relevant section for how to do it using the GUI.

The patterns you create using “saved values” are in fact stored in simple text files:

- `~/.quodlibet/lists/queries.saved`: Search patterns
- `~/.quodlibet/lists/tagpatterns.saved`: Patterns to tag files based on filename
- `~/.quodlibet/lists/renamepatterns.saved`: Patterns to rename files based on tags

You'll see that the format is very easy: Each saved pattern consists of two lines, the first line contains the QL pattern, the second line its name. The name may be identical to the pattern. The next saved "pattern – name" pair follows immediately on the next two lines.

Here's an example of what `~/ .quodlibet/lists/queries.saved` might look like:

```
artist = schubert
All by Schubert
artist = radiohead
All by Radiohead
&(genre = classical, #(lastplayed > 3 days))
&(genre = classical, #(lastplayed > 3 days))
~format = ogg
All ogg files
```

Or an example of a `~/ .quodlibet/lists/renamepatterns.saved`:

```
~/music/<artist>/<album>/<tracknumber|<tracknumber>. ><title~version>
Music from an album
~/music/misc/<artist> - <title>
Stray song
```

Just edit these files or synchronize them across computers or home directories (for different users) as needed.

## 5.13 Command Manuals

### 5.13.1 quodlibet

#### audio library manager and player

##### Manual section 1

#### SYNOPSIS

**quodlibet** [ **-print-playing** | *control* ]  
**exfalso** [ *directory* ]

#### DESCRIPTION

Quod Libet is a music management program. It provides several different ways to view your audio library, as well as support for Internet radio and audio feeds. It has extremely flexible metadata tag editing and searching capabilities.

This manual page is only a short reference for Quod Libet. Complete documentation is available at <https://quodlibet.readthedocs.io/en/latest/guide/index.html>.

#### OPTIONS

- add-location=<location>** Add a file or directory to the library
- debug** Print debugging information
- enqueue=<filename|query>** Enqueue a filename or query results

---

**--enqueue-files=<filename[,filename..]>** Enqueue comma-separated files  
**--filter <tag=value>** Filter on a tag value  
**--focus** Focus the running player  
**--force-previous** Jump to previous song  
**--help** Display brief usage information  
**--hide-window** Hide main window  
**--list-browsers** List available browsers  
**--next** Jump to next song  
**--no-plugins** Start without plugins  
**--open-browser=BrowserName** Open a new browser  
**--pause** Pause playback  
**--play** Start playback  
**--play-file=filename** Play a file  
**--play-pause** Toggle play/pause mode  
**--previous** Jump to previous song if near the beginning, otherwise restart  
**--print-playing** Print out information about the currently playing song. You may provide in a string like the kind described in the RENAMING FILES section below.  
**--print-playlist** Print the current playlist  
**--print-query=<query>** Print filenames of results of query  
**--print-query-text** Print the active text query  
**--print-queue** Print the contents of the queue  
**--query=search-string** Search your audio library  
**--queue=<on|off|t>** Show or hide the queue  
**--quit** Exit Quod Libet  
**--random=tag** Filter on a random value  
**--rating=<[+|-]0.0..1.0>** Set rating of playing song  
**--rating-down** Decrease rating of playing song by one star  
**--rating-up** Increase rating of playing song by one star  
**--refresh** Refresh and rescan library  
**--repeat=<off|on|t>** Turn repeat off, on, or toggle  
**--repeat-type=<current|all|one|off>** Repeat the currently playing song, the current list, stop after one song, or turn repeat off  
**--run** Start Quod Libet if it isn't running  
**--seek=<[+|-][HH:]MM:SS>** Seek within the playing song  
**--set-browser=BrowserName** Set the current browser  
**--show-window** Show main window  
**--shuffle=<off|on|t>** Turn shuffle off, on, or toggle

**--shuffle-type=<random|weighted|off>** Set the shuffle type to be random, to prefer higher rated songs, or turn shuffle off

**--start-hidden** Don't show any windows on start

**--start-playing** Begin playing immediately

**--status** Print playing status

**--stop** Stop playback

**--stop-after=<0|1|t>** Stop after the playing song

**--toggle-window** Toggle main window visibility

**--unfilter** Remove active browser filters

**--unqueue=<filename|query>** Unqueue a file or query

**--version** Display version and copyright

**--volume=<[+|-]0..100>** Set the volume

**--volume-down** Turn down the volume

**--volume-up** Turn up the volume

## ALBUM COVERS

Album covers should be put in the same directory as the songs they apply to, and have “folder”, “front”, or “cover” in their filenames. If you want to store multiple albums in the same directory but keep distinct cover images, the name of the appropriate image file must contain the labelid tag value, e.g. COCX-32760 cover.jpg.

## TIED TAGS

Many places in Quod Libet allow you to use “tied tags”. Tied tags are two tag names joined together with a “~” like “title~version” or “album~part”. Tied tags result in “nice” displays even when one of the tags is missing; for example, “title~version” will result in Title - Version when a version tag is present, but only Title when one isn't. You can tie any number of tags together.

## SEARCH SYNTAX

All of Quod Libet's search boxes support advanced searches of the following forms:

tag = value  
tag = !value  
tag = “value”  
tag = /value/  
tag = &(value1, value2)  
tag = l(value1, value2)  
!tag = value  
l(tag1 = value1, tag2 = value2)  
&(tag1 = value1, tag2 = value2)  
#(numeric tag < value)

`#(numerictag = value)`  
 `#(numerictag > value)`

The ‘c’ postfix on strings or regular expressions makes the search case-sensitive. Numeric values may be given as integers, floating-point numbers, MM:SS format, or simple English, e.g. “3 days”, “2 hours”.

See <https://quodlibet.readthedocs.io/en/latest/guide/searching.html>.

All internal tags begin with a ~ character. Non-numeric internal tags are ~base- name, ~dirname, ~filename, ~format, ~length, ~people, and ~rating. Numeric internal tags are ~#added, ~#bitrate, ~#disc, ~#lastplayed, ~#laststarted, ~#length, ~#mtime, ~#playcount, ~#skipcount, and ~#track.

See [https://quodlibet.readthedocs.io/en/latest/guide/tags/internal\\_tags.html](https://quodlibet.readthedocs.io/en/latest/guide/tags/internal_tags.html).

## RENAMING FILES

Quod Libet allows you to rename files based on their tags. In some cases you may wish to alter the filename depending on whether some tags are present or missing, in addition to their values. A common pattern might be

```
<tracknumber>. <title~version>
```

You can use a ‘|’ to only text when a tag is present:

```
<tracknumber|<tracknumber>. ><title~version>
```

You can also specify literal text to use if the tag is missing by adding another ‘|’:

```
<album|<album>|No Album> - <title>
```

See [https://quodlibet.readthedocs.io/en/latest/guide/renaming\\_files.html](https://quodlibet.readthedocs.io/en/latest/guide/renaming_files.html).

## AUDIO BACKENDS

Quod Libet uses GStreamer for audio playback. It tries to read your GConf GStreamer configuration, but if that fails it falls back to osssink. You can change the pipeline option in `~/.quodlibet/config` to use a different sink, or pass options to the sink. For example, you might use `esdsink` or `alsasink device=hw:1`.

See <https://quodlibet.readthedocs.io/en/latest/guide/playback/backends.html>.

## FILES

**~/.quodlibet/songs** A pickled Python dict of cached metadata. Deleting this file will remove all songs from your library.

**~/.quodlibet/config** Quod Libet’s configuration file. This file is overwritten when Quod Libet exits.

**~/.quodlibet/current** A “key=value” file containing information about the currently playing song.

**~/.quodlibet/control** A FIFO connected to the most-recently-started instance of the program. `–next`, `–previous`, etc., use this to control the player.

**~/.quodlibet/plugins/** Put plugins here.

**~/.quodlibet/browsers/** Put custom library browsers here.

See <https://quodlibet.readthedocs.io/en/latest/guide/interacting.html>.

### BUGS

See <https://github.com/quodlibet/quodlibet/issues> for a list of all currently open bugs and feature requests.

### AUTHORS

Joe Wreschnig and Michael Urman are the primary authors of Quod Libet.

### SEE ALSO

<https://quodlibet.readthedocs.io/en/latest/guide/>,  
<https://quodlibet.readthedocs.io/en/latest/guide/faq.html>,  
`regex(7)`, `gst-launch(1)`

## 5.13.2 exfalse

### audio tag editor

Manual section 1

### SYNOPSIS

**exfalse** [ *directory* ]

### DESCRIPTION

Ex Falso displays and edits audio metadata tags. Supported formats include MP3, Ogg Vorbis, FLAC, Musepack (MPC), WavPack, and MOD/XM/IT.

This manual page is only a short reference for Ex Falso. Complete documentation is available at <https://quodlibet.readthedocs.io/en/latest/guide/>.

### OPTIONS

Ex Falso may be given a directory to open on the command line.

### TIED TAGS

Many places in Ex Falso allow you to use “tied tags”. Tied tags are two tag names joined together with a “~” like “title~version” or “album~part”. Tied tags result in “nice” displays even when one of the tags is missing; for example, “title~version” will result in Title - Version when a version tag is present, but only Title when one isn’t. You can tie any number of tags together.

## RENAMING FILES

Ex Falso allows you to rename files based on their tags. In some cases you may wish to alter the filename depending on whether some tags are present or missing, in addition to their values. A common pattern might be

```
<tracknumber>. <title~version>
```

You can use a ‘!’ to only text when a tag is present:

```
<tracknumber|<tracknumber>. ><title~version>
```

You can also specify literal text to use if the tag is missing by adding another ‘!’:

```
<album|<album>|No Album> - <title>
```

See [https://quodlibet.readthedocs.io/en/latest/guide/renaming\\_files.html](https://quodlibet.readthedocs.io/en/latest/guide/renaming_files.html).

## BUGS

See <https://github.com/quodlibet/quodlibet/issues> for a list of all currently open bugs and feature requests.

## AUTHORS

Joe Wreschnig and Michael Urman are the primary authors of Ex Falso.

## SEE ALSO

<https://quodlibet.readthedocs.io/en/latest/guide/>,  
<https://quodlibet.readthedocs.io/en/latest/guide/faq.html>

### 5.13.3 operon

#### command line music tagger

Manual section 1

## SYNOPSIS

**operon** [*–version*] [*–help*] [*–v* | *–verbose*] *<command>* [*<argument>...*]

**operon help** *<command>*

## OPTIONS

<b>–h, --help</b>	Display help and exit
<b>--version</b>	Print the program version
<b>–v, --verbose</b>	Verbose mode

## COMMAND-OVERVIEW

### Edit Tags

*add* Add a tag value  
*remove* Remove a tag value  
*set* Set a tag and remove existing values  
*clear* Remove tags  
*copy* Copy tags from one file to another  
*edit* Edit tags in a text editor  
*fill* Fill tags based on the file path

### Show file metadata

*list* List tags  
*info* List file information  
*print* Print tags based on the given pattern

### Edit Embedded Images

*image-extract* Extract embedded images  
*image-set* Set embedded image  
*image-clear* Remove embedded images

### Miscellaneous

*tags* List all common tags  
*help* Display help information

## EDIT TAGS

### add

Add a new tag `<tag>` with the value `<value>` to all files.

operon add [-h] `<tag>` `<value>` `<file>`...

**-h, --help**                      Display help and exit

**Example:** operon add artist 'The Beatles' song1.ogg song2.ogg

### remove

Remove all values from the tag `<tag>` that match either `<value>` or the regular expression `<pattern>` from all files.

operon remove [-h] [--dry-run] `<tag>` (-e `<pattern>` | `<value>`) `<file>`...



- h, --help** Display help and exit
- dry-run** Print the results without changing any files
- e, --regexp <regexp>** Remove all tag values that match the given regular expression

**Example:** operon remove artist 'The Beatles' song.ogg

## set

Replace all values of the tag `<tag>` by `<value>` in all files.

operon set [-h] [--dry-run] <tag> <value> <file>...

- h, --help** Display help and exit
- dry-run** Print the results without changing any files

**Example:** operon set artist 'The Beatles' song.ogg

## clear

Remove all tags that match `<tag>` or the regular expression `<pattern>` from all files. If `--all` is specified, all known tags will be removed.

operon clear [-h] [--dry-run] (-a | -e <pattern> | <tag>) <file>...

- h, --help** Display help and exit
- dry-run** Print the results without changing any files
- a, --all** Remove all tags
- e, --regexp <regexp>** Remove all tags that match the given regular expression

**Example:** operon clear -a song.ogg

operon clear -e 'musicbrainz\_.\*' song.ogg

operon clear date song.ogg

## copy

Copy all tags from the file `<source>` to `<dest>`. All tags in `<dest>` will be preserved. In case the destination format doesn't support setting a tag from source, no tags will be copied. To ignore tags that aren't supported by the destination format pass `--ignore-errors`.

operon copy [-h] [--dry-run] [--ignore-errors] <source> <dest>

- h, --help** Display help and exit
- dry-run** Print the results without changing any files
- ignore-errors** Skip tags which the target file does not support

**Example:** operon copy song.flac song.ogg

## edit

Shows all tags in a text editor and will apply any changes made to the text to the tags. *operon* will use the editor specified in the VISUAL or EDITOR environment variables and if those are not set fall back to 'nano'.

`operon edit [-h] [--dry-run] <file>`

<b>-h, --help</b>	Display help and exit
<b>--dry-run</b>	Print the results without changing any files

**Example:** `VISUAL=vi operon edit song.flac`

## fill

Fill tags based on file paths and a given pattern.

`operon fill [-h] [--dry-run] <pattern> <file>...`

<b>-h, --help</b>	show this help message and exit
<b>--dry-run</b>	show changes, don't apply them

**Example:** `operon fill --dry-run "<tracknumber>. <title>" "01. Was Ist Ist.flac"`

## SHOW FILE METADATA

### list

Lists all tags, values and a description of each tag in a table.

`operon list [-h] [-a] [-t] [-c <c1>,<c2>...] <file>`

<b>-h, --help</b>	Display help and exit
<b>-a, --all</b>	Also list programmatic tags
<b>-t, --terse</b>	Output is terse and suitable for script processing

**-c, --columns <name>,...** Defines which columns should be printed and in which order

**Example:** `operon list -a song.flac`

`operon list -t -c tag,value song.ogg`

### info

Lists non-tag metadata like length, size and format.

`operon info [-h] [-t] [-c <c1>,<c2>...] <file>`

<b>-h, --help</b>	Display help and exit
<b>-t, --terse</b>	Output is terse and suitable for script processing

**-c, --columns <name>,...** Defines which columns should be printed and in which order

**Example:** `operon info a.ogg`

## print

Prints information per file built from tag values. The pattern can be customized by passing a pattern string (See `quodlibet(1)` for the pattern format)

`operon print [-h] [-p <pattern>] <file>...`

- h, --help**                Display help and exit
- p, --pattern <pattern>**   Use a custom pattern

**Example:** `operon print -p "<album> - <artist>" a.ogg`

## EDIT EMBEDDED IMAGES

### image-extract

Extract all embedded images to the current working directory or the specified destination directory.

`operon image-extract [-h] [--dry-run] [--primary] [-d <destination>] <file>...`

- h, --help**                Display help and exit
- dry-run**                Print the found images and resulting file paths but don't save them
- primary**                Only extract the primary images for each file
- d, --destination <destination>**   Save all images to the specified destination

**Example:** `operon image-extract asong.mp3 anotherone.ogg`

### image-set

Set the provided image as primary embedded image and remove all other embedded images.

`operon image-set <image-file> <file>...`

- h, --help**                Display help and exit

**Example:** `operon image-set cover.jpg song.mp3`

### image-clear

Remove all embedded images from all specified files.

`operon image-clear <file>...`

- h, --help**                Display help and exit

**Example:** `operon image-clear song.mp3`

## MISCELLANEOUS

### tags

List all common tags

`operon tags [-h] [-a] [-t] [-c <c1>,<c2>...]`

<b>-h, --help</b>	Display help and exit
<b>-a, --all</b>	Also list programmatic tags
<b>-t, --terse</b>	Output is terse and suitable for script processing

**-c, --columns <name>,...** Defines which columns should be printed and in which order

**Example:** operon tags -a

operon tags -t -c tag

### help

operon help [<command>]

**Example:** operon help list

### SEE ALSO

regex(7)  
exfalse(1)  
quodlibet(1)

## 5.14 Frequently Asked Questions

### 5.14.1 Why don't all my songs appear in the song list when searching for them?

Do you have a global filter in use? Check the *Browsers* tab in *Preferences*.

### 5.14.2 Where does Quod Libet store all its metadata?

The short answer was originally: in `~/ .quodlibet`. For newer versions of QL it's more complex:

- On Windows, it will be in your user's AppData folder under Quod Libet (except portable builds)
- On OS X, it will be in `~/ .quodlibet`.
- On Linux / Unix systems,
  - if the `QUODLIBET_USERDIR` environment variable is set, this will be used
  - else, `$XDG_CONFIG_HOME/config` will be used, if it exists
  - else `~/ .quodlibet` will be used still.

Under there you'll find all sorts of things, separate from the audio file tags themselves, e.g.

- `songs` - the pickled songs database.
- `config` - the master Quod Libet configuration file - edit with care
- `playlists/` - a directory for all playlists
- `lists/` - a directory for saved searches and so on
- `stations/stations_all` - the Internet radio stations lists

### 5.14.3 Whenever I type a space, Quod Libet pauses

Users of some keyboard layouts, including the popular French Alternative, may hit this bug. In these layouts, the spacebar sends a non-breaking space character, which GTK+ interprets as `<control>space`. This is a [known bug in GTK](#). You can work around it by changing your keyboard layout to send a regular space.

### 5.14.4 How do I add custom / unusual tags to the columns in the song list?

1. Refer to *editing tags* if you need to add any custom tags.
2. Right-click the song list header bar and select *Customize Headers* from the context menu (or click *Preferences* from the main menu and select *Song List*)
3. In the *Others* field, click *Edit*, then *Add*, and enter the custom tag name, remembering that they are case-sensitive.

Lesser-known (but useful) tags here might include `~#playcount`, `~#skipcount`, `~#bitrate` or `~playlists`.

### 5.14.5 How do I use a different soundcard with Quod Libet?

See the chapter on configuring the AudioBackends in the user's guide.

### 5.14.6 Why does Quod Libet sort my songs out of order?

Music metadata, like music, comes in many languages, and sorting multi-language text is hard to do. It depends on your language as well as the text being sorted, and often is still not well-defined. [Unicode Technical Standard #10](#) outlines an algorithm to sort multi-language text, but even then it needs ordinal data for each character for each language. We don't know of any Python implementations of it, and any implementation we use would have to be fast since we compare thousands of strings when sorting.

### 5.14.7 I have two albums with the same name which are merged in the Album List

Tag them with different `albumartist` tags. You can also use `musicbrainz_albumid` tags, which several other taggers and our "MusicBrainz Lookup" plugin can write.

### 5.14.8 I have two discs of the same album, and they don't get merged in the Album List

Make sure they have the same name (i.e. without "(disc x)" on the end). If they are still not merged, they have different `albumartist`, `labelid` or `musicbrainz_albumid` tags. If they have different label ID tags, delete the incorrect one. If they have different MusicBrainz album ID tags, add a `labelid` tag that is the same for both albums.

### 5.14.9 Where does Quod Libet store ratings and playcounts?

By default these are in the internal database - and remember this is indexed by filename, so renames will lose data. For this and other reasons, some users prefer to save them *in the tags themselves* (under an email address), using the most appropriate tag for that format (e.g. ID3's `popm` tag for ID3). See the configuration under *Preferences* -> *Tags* -> *Save ratings and play\_counts in tags*.

Note that some caching is used, so changes aren't written all the time. Try the *Update Tags In Files* plugins to control this explicitly.

### 5.14.10 Can I show more than 0 to 4 notes when rating songs?

Close Quod Libet; in `~/.quodlibet/config` find the `ratings = 4` line. Change it to `ratings =` however many ratings you want. It's best if the value divides 100 evenly; multiples of 2 and 5 are good. You will need to use the ratings right-click menu to set ratings above 4.

### 5.14.11 How can I hide incomplete albums from the Album View?

One way is to enter `#(tracks > 5)` into the search box above the album list - this will only show albums with greater than 5 tracks.

### 5.14.12 How can I list my tracks based on their ratings?

Right-click somewhere on the headers bar (below the search bar), select "Track Headers" from the menu and add "Ratings". Now if you click "Ratings" on the headers bar your tracks will be sorted based on their ratings.

### 5.14.13 How is album art handled?

There are many ways users like to keep their album art, and Quod Libet supports graphics (primarily *.jpg* but *.gif* and *.png* also) in these ways:

- Files in the *album* directory with fixed names eg `folder.jpg`, `cover.jpg`, `front.png`
- A file containing the `labelid` (eg *COCX-32760 cover.jpg*)
- Files of certain other names linked to a given album in a shared directory: `<musicbrainz_albumid>.ext` or `<artist> - <title>.ext`
- Sub-folders of certain names (`covers/` or `<labelid>/`) with compatible images in them.
- Embedded cover art in the file itself (incomplete support in some formats).

There are fuzzy-matching algorithms to try to determine the most specific match if multiple of the above exist.

If you're adding new album art, the *Album Art downloader* plugin allows you to do so easily and is compatible with the above.

### 5.14.14 Why do songs disappear from my playlists?

This is due to the way the library works, and that playlists entries are based on filename. One of several things might have happened, before a re-scan of the library (on start-up or otherwise)

- The songs have been renamed, moved, or their directory moved. Note this includes using *Rename Files* from the tag editor.
- A removable (mounted) media device - USB disk, network share, internet folder or whatever is/was no longer available (at the time of refresh).

Note if you're using the Auto Library Update this will happen immediately (There are ideas to improve this: Issue 961).

### 5.14.15 Can QL read my ID3 tags encoded in euc-kr / cp1251 / windows-1252 etc?

You can define a custom list of encodings to check. UTF-8 is always tried first, and Latin-1 is always tried last. To make your own list, close QL, open up `~/ .quodlibet/config`, and find the `id3encoding` option. You can enter any valid encodings here, separated by spaces, and they will be tried in order. If you have files already imported into your library with incorrect tags, you'll need to reload them.

Quod Libet saves ID3 tags in UTF-8 or UTF-16.

### 5.14.16 What does the name mean?

*Quodlibet* or *Quod libet* is Latin for “whatever you please” or “whatever you want”, which is the kind of attitude we want to convey with QL: you control how you fiddle with your music. A *quodlibet* is also a type of musical composition, an improvisation by several players or vocalists at once, which is a pretty accurate description of QL's development.

*Ex falso quodlibet*, or “from a falsehood, whatever you please” is one of the properties of material implication (*if/then*) in classical logics; in standard notation it can be written as  $A \rightarrow B$ .

Finally, the initial directory imported into Subversion was named *ql*, because I was experimenting with a syntax for a `_query _language`.

### 5.14.17 Where do the release names come from?

Daily Dinosaur Comics at the time of the release.

### 5.14.18 I like <my favorite player>, so I won't use Quod Libet!

Okay. We think Quod Libet beats other players in the areas where it counts (where exactly it does count is undecided; ‘tag editing’, ‘massive libraries’, and ‘regex searching’ have all been cited); we didn't like the other players. If you do, continue using them. You still might want to check out Ex Falso, since while there's an awful lot of media players out there, there are far fewer choices for tag editors. You could also *help us make Quod Libet better*.

### 5.14.19 Changing the volume in Quod Libet changes the master volume!

Since 3.5 Quod Libet will control the PulseAudio stream volume directly (same as the application slider in `pavucontrol`) which might have an effect on the master volume and vice versa. To restore the old behavior disable `flat-volumes` mode in PulseAudio. See `man pulse-daemon.conf` for more information.





This page is directed at distributions, packagers and developers. Please *contact us* if there is anything unclear / out of date / missing. For license & copyright information see *License & Contributors*

### 6.1 Existing Packaging

The following distributions package Quod Libet:

- Arch Linux: <https://www.archlinux.org/packages/extra/any/quodlibet/>
- Debian: <https://packages.debian.org/source/sid/quodlibet>
- Fedora: <https://admin.fedoraproject.org/pkgdb/package/rpms/quodlibet/>

The Ubuntu PPA / unstable repo builds are automated by the following scripts:

<https://github.com/quodlibet/ppa-scripts>

### 6.2 Non-Optional Runtime Dependencies

The following software is needed to start Ex Falso or Quod Libet.

- **Python** ( $\geq 3.5$ )
- **PyGObject** including **cairo support** ( $\geq 3.18$ )
- **pycairo** ( $\geq 1.8$ )
- **mutagen** ( $\geq 1.34$ )
- **GTK+** ( $\geq 3.18$ )
- **libsoup** ( $\geq 2.52$ )
- On OS X only: **PyObjC**

- **feedparser**

For icons a complete **icon theme** is needed, preferably with symbolic icons. For example **adwaita-icon-theme**.

For playback support in Quod Libet one of the following two is needed:

### 6.2.1 GStreamer

**Required:**

- **GStreamer** ( $\geq 1.8$ ) + **typelibs**
- **GStreamer Plugins Base**: Vorbis, Alsa, ...

**Optional but recommended:**

- **GStreamer Plugins Good**: Pulseaudio, FLAC, Jack, ...
- **GStreamer Plugins Ugly**: MP3 (mad), ...
- **GStreamer Plugins Bad**: MP3 (mpg123), MP4, Opus, ...
- **GStreamer libav/ffmpeg**: WMA, ...

### 6.2.2 Xine

- **xine-lib** 1.1 or 1.2 (the shared library, no Python bindings)

## 6.3 Optional Runtime Dependencies

**dbus-python:**

- Enables the DBus interface
- Multimedia key support under GNOME

**libkeybinder-3.0 + typelib:**

- Multimedia key support under non Gnome setups

**libgtksourceview-3 + typelib:**

- Undo/Redo support for multiline text fields

**libmodplug1:**

- For MOD support

## 6.4 Plugin Dependencies

All plugin dependencies are optional and will only prevent the corresponding plugin from loading.

**notification-daemon (or any other implementation of the dbus spec):**

- For the notification plugin

**python-musicbrainzngs ( $\geq 0.5$ ):**

- For the musicbrainz plugin

**GStreamer Plugins Good:**

- For the replaygain plugin

**GStreamer Plugins Bad:**

- For the acoustid plugin

**python-dbus:**

- “Browse Folders”
- Screensaver plugins
- uPnP server
- Gnome search provider
- gajim status updater
- MPRIS
- ...

**rygel:**

- The uPnP media server

**pynotify:**

- For the auto library update plugin

**webkit2gtk (== 4.0) + typelib:**

- For the Lyrics Window plugin

**libappindicator-gtk3 + typelib:**

- For the Tray Icon plugin under Ubuntu Unity and KDE Plasma

## 6.5 Build Dependencies

- **Python**  $\geq 3.5$  (stdlib only)
- **gettext**  $\geq 0.19.8$  for translations.
- (optional) **sphinx**  $\geq 1.3$

For user documentation `setup.py build_sphinx` can be used to create the HTML user guide and put it in the build directory in the `sphinx` subdirectory. This is not part of the default build process and requires **sphinx**.

## 6.6 Testing Dependencies

- The build dependencies
- **pytest**
- **pyflakes**
- **pycodestyle**
- **polib**



### 7.1 How-to

If you're fluent in a language other than English, and have some spare time, you can help us translate Quod Libet. Translation is a continuous process on an active projects like this - as developers add new features or modify existing text used, new translations will be needed. Many of the translations files are missing a lot of strings, so please try the trunk version in your language.

Of course, there might be mistakes in the English, too! Please let us know if you find them.

#### 7.1.1 Translation Software

To translate, you'll want to have gettext and git installed:

```
apt-get install gettext git
```

For translating itself you need a PO editor like [Poedit](#):

```
apt-get install poedit
```

#### 7.1.2 Translation Process

Get the QL code:

```
$ git clone https://github.com/quodlibet/quodlibet.git quodlibet
$ cd quodlibet/quodlibet
```

To translate the last release, update to the stable branch:

```
$ git branch -a # to get the list of branches
$ git checkout quodlibet-X.X # for example: quodlibet-3.3
```

To translate current trunk, update to the default branch:

```
$ git checkout master
```

You can find the translation file for your chosen language in:

```
./po/<lang>.po
```

In case there's not already a translation for your language, create one:

```
$ ./setup.py create_po --lang=<mylang>
```

Update translations so all new strings that were added since the last translation update get included:

```
$ ./setup.py update_po --lang=<mylang>
```

Now start translating...

```
$ poedit ./po/<lang>.po
```

Test the translation by generating MO files and running Quod Libet and Ex Falso. `build_mo` will create a 'build' directory including the processed translations and if 'build' is present QL/EF will use these translations instead of the global ones.

```
$ ./setup.py build_mo --lang=<mylang>
$ ./quodlibet.py
$ ./exfalso.py
```

If your active system language is not the one you are translating, you can run them like:

```
$ LANG=<mylang> ./quodlibet.py
```

Finally run our unit tests to make sure the translation will not cause programming errors. If it says something else, there's a problem with the translation.

```
$ ./setup.py test --to-run PO.<mylang>
```

And send us the .po file you made:

- Create a pull request.
- Or create a [new issue](#) linking to your updated .po file. If you don't have a place for making the file accessible create a [gist](#) with the content of the .po file and include the gist URL in the issue description.

## 7.2 FAQ

### 7.2.1 What do these things in strings mean?

- This is `%s,an %(foobat)s example,for {translators}, {0}`: see [String Formatting](#)

### 7.2.2 What does `msgctxt` in the .po files mean?

It stands for message context and (optionally) allows for different translations of the same translatable text appearing in different contexts. For example the word "Search" could be a title of a window or the label of a button. The former meaning "a search" and the latter "to search".

### 7.2.3 Why are some strings not translatable?

In case of strings like “Apply” and “Ok” that show up on buttons, these come from GTK and are not part of QL. But there is a chance that we have forgotten to mark a string translatable, so please contact us.

If anything else is not translatable you can start QL with the following variable set:

```
QUODLIBET_TEST_TRANS=xx ./quodlibet.py
```

which will append and prepend “xx” to all translatable strings.

### 7.2.4 What does *titlecase?* mean?

There’s a special string *titlecase?* which should be translated as anything if your language does not use *title casing* (eg *This Is Title Casing*) for labels. If it is left untranslated, title-casing will be used.

### 7.2.5 String Formatting

Some strings include replacement tokens like %s or {foobar}. These mark places where text gets replaced at runtime, so they should be carried over to the translation without changing their content.

Some examples showing the strings to translate and the resulting strings where the tokens have been replaced:

- Hello %s -> Hello Lou
- The number %d -> The number 42
- Hello %(name)s -> Hello Lou
- Hello {name} -> Hello Lou
- Hello {0} -> Hello Lou
- Hello {0.name} -> Hello Lou
- Hello {} -> Hello Lou

In the case of Hello %(name)s, a possible German translation and text displayed to the user would be Hallo %(name)s and Hallo Lou.

### 7.2.6 Plural-Forms

The first thing you need is a Plural-Forms line for your language. The GNU gettext manual has a chapter on plural forms with examples for many languages. This should go after the “Content-Transfer-Encoding” line.

The Plural-Forms line tells gettext how many plural forms there are in the language and how to use them. For example:

```
msgid "artist"
msgid_plural "artists"
msgstr[0] "artist"
msgstr[1] "artists"
```

The English plural expression, “n != 1” means to use msgstr[0] if the count is 1, otherwise use msgstr[1]. If your language has 3 plural forms, you’ll need msgstr[0], msgstr[1], and msgstr[2], and so on.

Sometimes (usually, even) the English strings will be the same. For example, %d selected doesn’t change whether it stands for *1 selected* or *99 selected*. If it does in your language, you should translate them differently. There are

further difficulties for the many languages that have gender agreement and an unspecified noun in the phrase, but these are often translated with brackets (eg in French: *1 sélectionné(e)*, *99 sélectionné(e)s* perhaps)

### 7.2.7 Fuzzy translations

A translation marked *fuzzy* is (usually) one that has been matched to a similar previous translation, often by *gettext* itself. Note that fuzzy translations are not treated as accurate translations so will not be used.

**Common reasons for strings being marked as fuzzy include:**

- A contributor corrects a typo in the source (English) text
- A developer changes the **Mnemonic Label** - This is the underscore you see in many translation strings.
- The English has changed, but not much
- sometimes it *just happens*...

For example:

```
#: ../quodlibet/browsers/albums.py:425
#, fuzzy
msgid "Sort _by:"
msgstr "Ordina per data"
```

Here, in the Italian *.po* file, you can see this message has been matched, presumably used from a “Sort by date” translation previously entered. This explains why this string was missing in the Italian build.

As a translator please make sure there are no translations left marked as fuzzy. In *poedit*, you can click the cloud (!) icon, or in a text editor you should simply remove the *fuzzy* string above the *msgid*.



### 8.1 Overview

#### 8.1.1 Building and Installing Quod Libet

While Quod Libet uses `distutils/setup.py` for building and installing (`./setup.py build install` etc.) we don't recommend it, as it doesn't provide a way to uninstall the application again and it might not do the right thing by default depending on your distribution/operating system.

Instead we recommend running Quod Libet directly from the git checkout for development/experiments and use one of our unstable repositories for everyday use.

See *[Creating a Development Environment](#)* on how to proceed.

#### 8.1.2 Testing Changes

To make sure that your changes don't break any existing feature you should run the test suite by executing:

```
./setup.py test
```

For more details and options regarding testing, code quality testing and test coverage see *[Testing](#)*.

### 8.2 Creating a Development Environment

This will show you how to get the latest and freshest Quod Libet running on your machine and allow you to change it as you wish. The main task here is to install all the software which Quod Libet uses and depends on.

### 8.2.1 Linux

The easiest and recommended way to get all the dependencies needed for the development version is to install one of our unstable repositories. By doing so all the needed dependencies are automatically installed along the way. See the [download section](#) for a list of available repositories.

In case your distribution is not supported you have to find/install the dependencies yourself. See the [Packaging Guide](#) for a list of dependencies.

Now clone the git repository and start Quod Libet:

```
$ git clone https://github.com/quodlibet/quodlibet.git
$ ./quodlibet/quodlibet/quodlibet.py
```

### 8.2.2 OS X

On OS X all the needed dependencies are included in the provided bundle itself. Download the latest bundle, which is guaranteed to work with current git master: [QuodLibet-latest.dmg](#). It contains a “run” script which passes all arguments to the included Python with the right environment set up.

```
$ git clone https://github.com/quodlibet/quodlibet.git
$ ./QuodLibet.app/Contents/MacOS/run <path_to_git_repo>/quodlibet/quodlibet.py
```

If you want to build a bundle yourself or change/add dependencies check out the “[osx\\_bundle](#)” directory in the git repo for further instructions.

### 8.2.3 Windows

On Windows we use the [msys2](#) environment for development.

Check out the [win\\_installer](#) directory in the git repo for further instructions.

## 8.3 Testing

Quod Libet uses the CPython unittest framework for testing and [pytest](#) as a test runner. All testing related code can be found under `quodlibet/quodlibet/tests`.

To run the full tests suite simply execute:

```
./setup.py test
```

For checking the code coverage of the test suite run:

```
./setup.py coverage
```

### 8.3.1 Selecting a Specific Test

To only test a subset of the test suite, pass a comma separated list of test class names to `setup.py` via the `--to-run` option. For example:

```
./setup.py test --to-run=TMP3File,TAPICType
```

Similarly the coverage report can also be generated for a subset of tests:

```
./setup.py coverage --to-run=TMP3File,TAPICType
```

Selecting by class name can take a long time because it needs to import all tests first. To speed things up you can just use `pytest` directly:

```
py.test tests/test_formats_mp3.py
py.test tests/test_formats*
py.test tests/test_formats_mp3.py::TMP3File
```

To just run code quality tests:

```
py.test tests/quality
```

Some helpful `py.test` options are `-s` for not hiding stdout and `-x` for stopping on the first error. For more information check out <https://docs.pytest.org/en/latest/usage.html>

### 8.3.2 Abort on First Error

By passing `-x` to `setup.py test` the test suite will abort once it sees the first error instead of printing a summary of errors at the end:

```
./setup.py test -x
```

## 8.4 How to contribute

### 8.4.1 Testing

One of the most helpful things both regular users and developers can do is to test others' code. The easiest way to do this is to run the development version of Quod Libet. Development versions are kept stable, and the developers generally run the latest code to play their own music, so this is a safe and helpful way to contribute.

Please keep in mind that Quod Libet is not forward compatible, meaning that if you use a newer version, reverting to an older version could lead to errors and data loss. So always backup your *config files* if you plan to downgrade at a later point.

You can find development packages in the *download section*.

### 8.4.2 Filing bug reports

#### Useful Links

- [List current bugs](#)
- [Add a new bug](#)

**Writing a good bug report** It helps the developers to format bugs in a standard way, with a short summary as the Issue title, ideally:

1. **Steps to reproduce** (how the bug can be demonstrated again)
2. **Expected output** (what *should* happen)
3. **Actual output** (what *did* happen)

Also: the more logs, system details, and insight about the library / files the better the chance of a speedy resolution.

For more general tips see “[How to Report Bugs Effectively](#)”.

**Look through existing issues** Quod Libet is a mature project (in its second decade!), and there have been a *lot* of features and bugs discussed over the years. It’s probable that what you’re thinking has been discussed at some point, so please search through existing open (and to a lesser extent closed) issues before creating a new one. This reduces noise and saves the maintainers time.

**One bug per ticket** Please do not create an item (ticket) in the issue tracker which contains reports of multiple unrelated issues. Even if you are reporting several very minor bugs, each one deserves its own issue. This allows each issue to receive independent discussion and analysis, and to be closed separately.

**Viewing Debug information** If the bug you have found does not raise an exception, the debug window won’t appear and the dump won’t be generated. In this case, run `quodlibet --debug` from the command line using the command `quodlibet --debug`. It will show additional information that might be useful.

**Testing the latest code** Some problems are fixed in the development branch which aren’t yet fixed in the current release. If you can, try to reproduce your bug against a recent checkout before filing.

### 8.4.3 Filing enhancement requests

The most important component of an enhancement is the *why*. State what it is Quod Libet doesn’t do for you, and give as much information about why you think adding a feature which accomplished this would be a good thing. If you have an idea as to how a feature might be implemented, suggestions are welcome, but be sure to explain why you want a feature before explaining how you envision it being implemented. Not only does this make your feature more likely to be supported, it allows others to enhance, generalize, and refine your ideas.

As with bugs, please check for existing feature requests first and refrain from submitting multiple feature requests in the same issue. If you have related ideas, file them separately and mention the issue numbers of previous ideas.

### 8.4.4 Translation

Help us *translate Quod Libet into other languages*. You don’t need to know how to program in Python to do it.

### 8.4.5 Getting started as a developer on Quod Libet

On an long-standing project it can be daunting helping out for the first time. The [newcomer-friendly](#) tag has been added to (some) issues to indicate where the existing developers feel there is an opportunity to help out without too much background (or risk). See the discussion around this in [Issue 2516](#)

The other area perfect for newcomers is in the rich *plugins ecosystem* - at time of writing QL has >80 plugins. These require less knowledge of Python, GTK+ and the QL architecture / codebase.

It’s best to examine existing (and past) PRs, keep an eye on the mailing list, and especially the Github issues list. Reading the unit / integration tests is usually instructive too. IRC can be a good place for more immediate questions.

### 8.4.6 Submitting changes

Patches are always welcome, and should be in the form of a pull request or by attaching a patch to the issue. Please work on existing issues where possible (there are a lot), or at the very least make sure there is an accompanying issue for your PR.

If you follow the [Coding Guidelines](#) it will be much easier to get your changes included quickly.

## 8.5 Coding Guidelines

### 8.5.1 Getting started

Before you can start making changes to Quod Libet you have to set up a development environment. See “*Creating a Development Environment*” for further details.

### 8.5.2 Source Overview

browsers/*	Things in the View menu
ext/*	Extensions to QL / EF (i.e. the plugins)
formats/*	File format support
library/*	Library management code
plugins/*	Base classes and structural enabling plugins
operon/*	Operon, the QL CLI tool
qltk/*	GTK+ widget subclasses/extensions
util/*	General utility functions and setup code

If you want to get a full overview of QL’s code, good places to start are `browsers/_base.py`, `formats/_audio.py`, and `library/libraries.py`.

### 8.5.3 Code Guidelines

We try to keep Quod Libet’s code in pretty good shape; when submitting a patch, it’s much easier to get it included quickly if you run through this checklist of common-sense code quality items. Make sure your patch:

- Passes existing tests. You can test this by executing `./setup.py test`
- Is commented.
- Adds your name to the copyright header of every file you touch. This helps you get credit and helps us keep track of authorship.

### 8.5.4 General Guidelines

We prefer Python to C. We prefer `ctypes` to compiled C wrappers. A good way to get a new feature applied is if you include tests for it. Stock strings and string reuse are awesome, but don’t make the interface awkward just to avoid a new string.

### 8.5.5 Unit Tests

Quod Libet comes with a lot of tests, which helps us control regression. To run them, run `./setup.py test`. Your patch can’t break any unit tests, and if you change tests in a non-obvious way (e.g. a patch that removes an entry point and also removes a test for it is obvious) you should explain why.

It’s possible, indeed encouraged, that a changeset was for no other purpose than to *improve* the testing / test coverage, as there have been plenty of bugs that have slipped through. As usual, any fix associated with a confirmed bug should include tests that would have found the original bug, where possible.

## 8.5.6 Printing Text

All terminal output should go through the `print_`, `print_w`, `print_e`, or `print_d` functions. These will handle Unicode recoding. They also let us capture all output for debugging purposes.

## 8.5.7 Translations

All text that could be visible to users (with debugging mode disabled) should be marked translatable.

You can do this by simply using the `_` function which is globally available (through `__builtin__`):

```
print_w(_("This is translatable"))
```

To handle plural forms use `ngettext`:

```
text = ngettext("%d second", "%d seconds", time) % time
```

It is good practice to add a comment for translators if the translation depends on the context:

```
# Translators: As in "by Artist Name"  
text = _("by %s") % tag
```

## 8.6 Plugin Development

A Quod Libet / Ex Falso Plugin is a simple python module or package which contains sub-classes of special plugin classes and is placed anywhere in `~/.config/quodlibet/plugins` so it can be found by QL. These classes can provide special methods that get used by QL or can take action on certain events.

At the moment the following plugin classes exist:

### 8.6.1 Plugin Types

#### Event Plugins

- Are instantiated as long as they are enabled or QL is running. They get notified when songs start, get added etc.
- Can also (3.10+) listen to selection changes in the main songlist.
- Example: Last.fm scrobbler that watches what you play and sends the information to last.fm.

#### Tag Editing Plugins

- Can extend many parts of the tag editor.
- Example: Title case your tags.

#### GStreamer Plugins

- Can inject elements into the GStreamer pipeline and configure them on the fly.
- Example: Tempo adjustment to play music/audio books faster.

#### Play Order: Shuffle Plugins

- Can decide which song to play next or what happens if you select one.
- Example: weighted shuffle, which prefers higher rated songs.

**Play Order: Repeat Plugins:**

- How to repeat the songs
- Example: repeat each song a set number of times

**Songs Menu Plugins**

- Can be accessed through the play list context menu and get passed all selected songs.
- Example: Album art search, to search album art for all selected songs.

**Playlist Plugins**

- Similar to Songs Menu plugin, and in fact derived on the same base class
- Can be accessed through the playlist context menu in the *Playlist Browser*
- Example: remove duplicate songs from a playlist.

**Cover Source Plugins**

- Fetch covers from external or local resources

## 8.6.2 Creating a new Plugin

1. Create a file `myplugin.py` and place it under `~/.config/quodlibet/plugins` (create the folder if needed). Alternatively (better), if you are running from source, put it in `quodlibet/ext` under a directory according to its plugin type.
2. Write the following into the file:

```
from quodlibet.plugins.events import EventPlugin

class MyPlugin(EventPlugin):
    PLUGIN_ID = "myplugin"
    PLUGIN_NAME = "My Plugin"
```

3. Restart Quod Libet
4. In Quod Libet open `Music Plugins` and search the list for “My Plugin”

**Tips:**

- The best way to find out what is possible is to read the documentation of the [plugin base classes](#).
- The easiest way to get started creating a new plugin is to look for [existing plugins](#) that do something similar to what you want.

## 8.7 Useful Development Tools

### 8.7.1 Memory Profiling

**GObject Instance Count Leak Check**

Requires a development (only available in debug mode) version of glib. Jhbuild recommended.

```
jhbuild shell
GOBJECT_DEBUG=instance-count GTK_DEBUG=interactive ./quodlibet.py
```

- In the GTK+ Inspector switch to the “Statistics” tab
- Sort by “Cumulative” and press the “Next” multimedia key to quickly switch songs.
- If something in the “Cumulative” column steadily increases we have a leak.

## 8.7.2 Performance Profiling

### cProfile

- <https://docs.python.org/2/library/profile.html>
- bundled with python

```
python -m cProfile -s [sort_order] quodlibet.py > cprof.txt
```

where `sort_order` can one of the following: calls, cumulative, file, line, module, name, nfl, pcalls, stdname, time

Example output:

```
885311 function calls (866204 primitive calls) in 12.110 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1      0.002      0.002   12.112   12.112 quodlibet.py:11(<module>)
      1      0.007      0.007   12.026   12.026 quodlibet.py:25(main)
19392/13067    0.151      0.000    4.342      0.000 __init__.py:639(__get__)
      1      0.003      0.003    4.232    4.232 quodlibetwindow.py:121(__init__)
      1      0.000      0.000    4.029    4.029 quodlibetwindow.py:549(select_browser)
      1      0.002      0.002    4.022    4.022 albums.py:346(__init__)
...
...
```

### SnakeViz

- <https://jiffyclub.github.io/snakeviz/>
- `pip install snakeviz`

```
python -m cProfile -o prof.out quodlibet.py
snakeviz prof.out
```

### Run Snake

- <http://www.vrplumber.com/programming/runsnakerun/>
- package `runsnakerun` in debian Wheezy

```
python -m cProfile -o prof.out quodlibet.py
runsnake prof.out
```

Example: <https://www.google.at/search?q=runsnakerun&tbm=isch>



## Gprof2Dot

- <https://github.com/jrfonseca/gprof2dot>

```
python -m cProfile -o output.pstats ./quodlibet.py
gprof2dot.py -f pstats output.pstats | dot -Tpng -o output.png
```

## Line Profiler

- [https://github.com/rkern/line\\_profiler](https://github.com/rkern/line_profiler)

```
# wrap all functions of interest with the @profile decorator
./kernprof.py -l ./quodlibet.py # creates quodlibet.py.lprof
python line_profiler.py quodlibet.py.lprof > prof.txt
```

Example output:

```
Timer unit: 1e-06 s

File: test.py
Function: a at line 2
Total time: 0.001528 s
```

Line #	Hits	Time	Per Hit	% Time	Line Contents
2					@profile
3					def a():
4	1	134	134.0	8.8	print "hello"
5	1	12	12.0	0.8	b = []
6	101	628	6.2	41.1	for i in xrange(100):
7	100	696	7.0	45.5	b.append(i)
8	1	58	58.0	3.8	print "world"

## strace

- <https://linux.die.net/man/1/strace>

```
strace -c ./quodlibet.py
```

Example output:

time	seconds	usecs/call	calls	errors	syscall
81.64	0.013274	9	1444	178	read
7.04	0.001144	0	4041	3259	open
3.81	0.000619	11	54		getdents64
2.80	0.000456	0	1004		fstat64
1.84	0.000299	0	2221	1688	stat64
...					
...					

## IOProfiler

- <https://code.google.com/archive/p/ioapps/wikis/ioprofiler.wiki>

- strace GUI
- Shows read/write to files (how many reads/writes per file, which parts of the files were affected)

```
strace -q -al -s0 -f -tttT -oOUT_FILE -e trace=file,desc,process,socket ./quodlibet.py
ioreplay -c -f OUT_FILE -o OUT_FILE.bin
ioprofiler.py
# open OUT_FILE.bin
```

Example: <https://code.google.com/archive/p/ioapps/wikis/IOProfilerScreenshots.wiki>

## 8.8 Frequently Asked Questions

### 8.8.1 Any plans to use Python 3?

Older releases of QL use Python 2. As of *release 4.0.0*, QL is Python 3 only.

#### Why don't you use SQLite for the song database?

Although the song data Quod Libet stores would benefit from a relational database, it does not have a predefined schema, and opts to let users define their own storage keys. This means relational databases based on SQL, which require predefined schemata, cannot be used directly.

#### What about <my favourite NoSQL DB> then?

This gets asked fairly often. MongoDB, CouchDB etc are indeed a closer match to the existing setup, but there is *significant* work porting to any of these, and each comes with a compatibility / maintenance cost. There has to be a genuine case for the benefits outweighing the migration cost.

#### Any environment variables I should know about?

**QUODLIBET\_TEST\_TRANS** When set to a string will enclose all translatable strings with that string. This is useful for testing how the layout of the user interface behaves with longer text as can occur with translations and to see if all visible text is correctly marked as translatable.

```
QUODLIBET_TEST_TRANS=XXX
```

**QUODLIBET\_DEBUG** When in the environment gives the same result as if `--debug` was passed.

**QUODLIBET\_BACKEND** Can be set to the audio backend, overriding the value present in the main config file. Useful for quickly testing a different audio backend.

```
QUODLIBET_BACKEND=xinebe ./quodlibet.py
```

**QUODLIBET\_USERDIR** Can be set to a (potentially not existing) directory which will be used as the main config directory. Useful to test Quod Libet with a fresh config, test the initial user experience, or to try out things without them affecting your main library.

```
QUODLIBET_USERDIR=foo ./quodlibet.py
```

## 8.9 Maintainer Guidelines

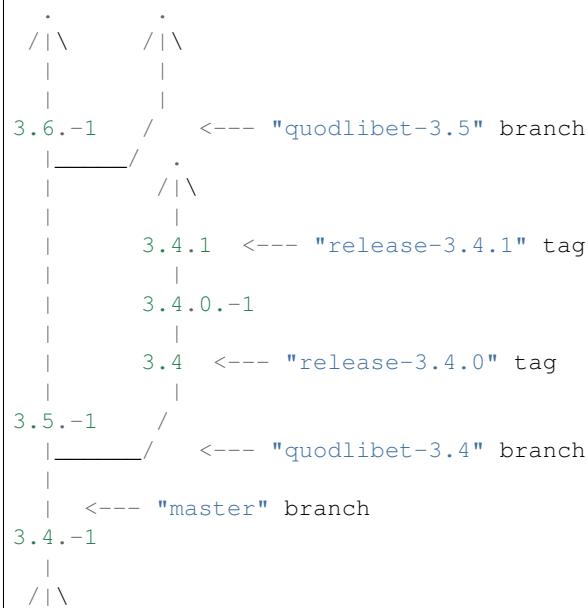
### 8.9.1 Downstream Bug Trackers

Some bug reports never make it to us so check these once in a while.

- [Fedora](#)
- [Debian](#)
- [Arch Linux](#)
- [Ubuntu](#)
- [Gentoo](#)

### 8.9.2 Tags & Branches

At the point where no more functionality will be added to a release, a new branch gets created. All bug fixes pushed to the master branch should be cherry-picked to the respective stable branches and vice versa.



### 8.9.3 Release Checklist

New Stable branch:

- `git checkout -b quodlibet-x.y`
- `git commit -m "new stable branch"`
- `git push`
- `git checkout master`
- Update version to (X, Y + 1, -1)
- `git commit -m "version bump"`

New Stable release:

- `git checkout quodlibet-x.y`
- Cherry pick stuff from master
- Update NEWS
- `git commit -m "update NEWS"`
- `setup.py distcheck`
- Update version to (X, Y, Z) in `const.py`
- Update version to (X, Y, Z) in `appdata.xml.in`
- `git commit -m "release prep"`
- `git tag release-x.y.z`
- `git push origin release-x.y.z`
- Update version to (X, Y, Z, -1)
- `git commit -m "version bump"`
- Cherry pick NEWS commit onto master
- Create Windows builds / tarballs / macOS dmgs
- Create checksums / signature, attach everything to the github tag
- Update `release_db/make.py`; run `./release_db/update.sh`
- Update stable PPAs (ubuntu/debian/OBS)
- Update the flathub repo
- Write release mail

## 8.10 Tag Formats & Spec Deviations

### 8.10.1 Deviations

Quod Libet deviates from several of the standards it claims to support. We'll document them here, but most of the time we won't change them; we had a good reason for violating them in the first place.

### 8.10.2 General Deviations

Since we translate to and from many formats, sometimes we're forced to accept the lowest common denominator among tag formats. Don't worry; if it's too low we'll figure out some way around it.

- Dates are restricted to YYYY or YYYY-MM-DD formats. Blame ID3.
- `=` and `~` are not allowed in tag names, and tag names must also be pure ASCII.
- Tag names are always case-insensitive.
- Newlines are not allowed in tag values. If you use a newline in a tag value, it will instead be treated as two values for the tag.

### 8.10.3 ID3

#### UTF-8 and ID3 Tags

If Quod Libet finds data in an MP3 claiming to be “latin 1”, it won’t trust it. It’ll try UTF-8 and some other encodings before it gives up and actually considers it Latin 1. On the other hand, when it writes tags, it saves purely ASCII values as UTF-8 and any non-ASCII values as UTF-16. This means it won’t try to “guess” when it reloads them, and other programs shouldn’t either.

Using a format like Ogg Vorbis, FLAC, or Musepack, the tag encoding is known and Unicode is perfectly supported.

#### Genre and TCON

The ID3 standard is ambiguous in its specification for the TCON frame. If you try to use a numeric genre in the form of 02 or (23) (for example) it will get translated into a text genre. This means there’s no way to have a genre actually start with (xx) for MP3s.

#### QuodLibet::TXXX Frames

ID3 doesn’t let you have frames named whatever you want like Ogg Vorbis and APEv2 do. However, it does let you create “TXXX” frames which are text data with an associated “description”. Quod Libet uses these frames with a *‘QuodLibet’::<tagname> description* to store tags that don’t have ID3 counterparts. For example, labelid is stored as *‘QuodLibet’::labelid*.

#### RVA2 / ReplayGain

Quod Libet implements ReplayGain using MP3’s RVA2 frame. Unfortunately there is no standard on how to read RVA2 frames to support RG properly. If the description string is “album” the gain is treated as the album/audiophile value. Any other value is read as the track/radio value, but an actual description of “track” will preempt other values.

I’ve passed this information along to the GStreamer guys and their RVA2 support should match this, once it’s completed.

foobar2000-style TXXX:replaygain\_\* tags are also supported, but migrated to the proper RVA2 format on save.

#### COMM Frame Language codes

Language codes in COMM frames with empty descriptors are replaced by x00x00x00 on save. These tend to contain garbage rather than valid language codes anyway, and empty descriptors are usually a sign of comments migrated ID3v1 or other formats that do not support language markers.

#### Legacy Stuff

Multiframe Hack:

Since versions of ID3 prior to 2.4 did not support multiple values in a single text frame, we stored multiple text frames of the same type with one value for each if you tried to save more than one value per frame. This was strictly a violation of the ID3 spec, but we never encountered an ID3 reader that had trouble reading the tags saved this way (and still haven’t).

Now that we use Mutagen, we store multiple values in the standard ID3v2.4 format. Old tags are migrated when you edit them.

QuodLibet:: COMM Frames:

Quod Libet used to use COMM frames instead of TXXX frames for its extended tag names. It will still load old COMM tags, but clears them when you save the file again.

### 8.10.4 APEv2

#### Naming Conflicts

Since we turn APEv2's `Subtitle` tag into `version`, you can't edit a tag named `subtitle` in MPC files. Similar problems exist for `Track`, `Catalog`, `Year`, and `Record location` tags.

### 8.10.5 VorbisComment

This is a list of Ogg Vorbis tags Quod Libet uses that require special handling. They are presented here in the hopes that other applications will adopt them.

#### rating

The `rating` tag has a subkey of an email address, and is formatted as e.g. `rating:quodlibet@sacredchao.net`. The email is used as a unique identifier to allow multiple users to share the same files (it need not actually be an email address, but having it as such ensures that it's unique across users). It represents how much a user likes a song. Its value is a string representation of a floating point number between 0.0 and 1.0, inclusive. This format is chosen so the application may decide what precision it offers to the user, and how this information is presented. If no value is present, the rating should be assumed to be 0.5.

Example: `rating:quodlibet@sacredchao.net=0.67`

#### playcount

The `playcount` tag has a subkey of an email address, and is formatted as e.g. `playcount:quodlibet@sacredchao.net`. It stores how many times the user has played the song all the way through, as a numeric string. If no value is present, it should be assumed to be 0.

Example: `playcount:quodlibet@sacredchao.net=3`

#### website

The `website` tag stores an absolute IRI (Internationalized Resource Identifier), as per RFC 3987. The intent is that it hold an IRI suitable for opening in a standard web browser (e.g. `http`, `https`, or `ftp` scheme). As it is an IRI, and not a URI, it should be stored in unescaped form. If an application needs a URI, it should follow the procedure in RFC 3987 section 3.1 to convert a valid IRI to a valid URI. As per the Vorbis comment specification, the tag must be a UTF-8 representation of the Unicode string.

This tag may occur any number of times in a file.

### Performer Roles

This is similar to the ID3v2 IPLS (involved person list) frame. Quod Libet displays these tags by putting the role after the name, parenthesized (e.g. `Béla Fleck (banjo)`), but in other supporting programs the role can be associated with the name in any understandable way, or simply ignored and treated like an ordinary performer tag.

## 8.11 Flatpak

To share the config between the distro package/development environment and the flatpak symlink the config directory:

```
ln -s ~/.config/quodlibet ~/.var/app/io.github.quodlibet.QuodLibet/config
```

Our official flatpak is hosted on Flathub:

<https://github.com/flathub/io.github.quodlibet.QuodLibet>





---

## License & Contributors

---

### 9.1 License

Quod Libet and all associated modules are free software distributed entirely under the terms of the [GNU GPL version 2 or later](#). Alternate licensing terms such as MIT, BSD or LGPL may be available for certain files or modules; see the files for details.

#### 9.1.1 Authors

Alex Geoffrey Smith, Alexandre Passos, Alexey Bobyakov, Anders Carlsson, Andreas Bombe, Andrew Chadwick, Anton Shestakov, Ari Pollak, Arkadiy Illarionov, Aymeric Mansoux, Bastian Kleineidam, Bastien Gorissen, Ben Zeigler, Benjamin Boutier, Bernd Wechner, Bruno Bergot, Carlo Teubner, Christine Spang, Christoph Reiter, Corentin Néau, CreamyCookie@github, David Kågedal, David Morris, David Pérez Carmona, David Schneider, Decklin Foster, Didier Villevalois, Eduardo Gonzalez, Emanuele Baldino, Eoin O'Neill, Eric Casteleijn, Eric Le Lay, Erich Schubert, Eyenseo@github, Federico Pelloni, Felicián Németh, Felix Krull, Florian Demmer, Fredrik Strupe, Guillaume Chazarain, Hans Scholze, IBBoard@github, Iñigo Serna, Jacob Lee, Jakob Gahde, Jakub Wilk, Jan Arne Petersen, Jan Path, Jason Heard, Javier Kohen, Joe Higton, Joe Wreschnig, Johan Hovold, Johannes Marbach, Johannes Rohrer, Jonas Platte, Joschka Fischer, Josh Lee, Joshua Homan, Joshua Kwan, Kristian Laakkonen, Lalo Martins, Lee Willis, Ludovic Druette, Lukáš Lalinský, Markus Koller, Martijn Pieters, Martin Bergström, Meriipu@github, Michael Uрман, Michaël Ball, Mickael Royer, Muges@github, Nicholas J. Michalek, Nick Boulton, Niklas Janlert, Nikolai Prokoschenko, Olli Helin, Pete Beardmore, Peter F. Patel-Schneider, Peter Simonyi, Peter Strulo, Phidica@github, Philipp Müller, Philipp Weis, Phoenix Dailey, Quincy John Hamilton, Remi Vanicat, Robert Muth, Ruud van Asseldonk, Ryan Turner, Sauyon Lee, Sebastian Thürschmidt, Simon Larsen, Simonas Kazlauskas, Steven Robertson, Thomas Leberbauer, Thomas Vogt, Till Berger, Tobias Wolf, Tomasz Miasko, Tomasz Torcz, Tshepang Lekhonkhobe, Türkerkan İnce, Uriel Zajaczkowski, Vasiliy Faronov, Victoria Hayes, Vimalan Reddy, Zack Weinberg, a-vrma@github, dpitch40@github, luk1337@github, luzpaz@github, sphh@github, zsau@github

#### 9.1.2 Translators

Alexandre Passos (pt), Andreas Bertheussen (nb), Anton Shestakov (ru), Avi Markovitz (he), Bastian Kleineidam (de), Bastien Gorissen (fr), Byung-Hee HWANG (ko), ChangBom Yoon (ko), Daniel Nyberg (sv), Dimitris Papageorgiou

(el), Djavan Fagundes (pt), Einārs Sprūģis (lv), Eirik Haatveit (nb), Emfox Zhou (zh\_CN), Erik Christiansson (sv), Fabien Devaux (fr), Filippo Pappalardo (it), Guillaume Ayoub (fr), Hans van Dok (nl), Honza Hejzl (cs\_CZ), Hsin-lin Cheng (zh\_TW), Jari Rahkonen (fi), Javier Kohen (es), Jean-Michel Pouré (fr), Joe Wreschnig (en\_CA), Johám-Luís Miguéns Vila (es, gl, gl\_ES, eu, pt), Jonas Slivka (lt), Joshua Kwan (fr), Kirill Romanov (ru), Kristian Laakkonen (fi), Luca Baraldi (it), Ludovic Druette (fr), Lukáš Lalinský (sk), Marek Suchánek (cs), Mathieu Morey (fr), Michal Nowikowski (pl), Mugurel Tudor (ro), Mykola Lynnyk (uk), Naglis Jonaitis (lt), Nathan Follens (nl), Nick Boulton (fr, en\_GB), Olivier Gambier (fr), Olivier Humbert (fr), Piarres Beobide (eu), Piotr Drag (pl), Roe Haimovich (he), Rüdiger Arp (de), SZERVÁC Attila (hu), Till Berger (de), Tomasz Torcz (pl), Tükerkan İnce (tr), Witold Kieraś (pl), Yasushi Iwata (ja), scootergrisen (da), wvxwxvw@github (ru), Åka Sikrom (nb), Δημητρη Παπαγεωργίου (el), (ru), ‘Cthulhu’ (uk), (ru), “zbrox” (bg), (ru)

### 9.1.3 Artists

Fabien Devaux, Jakub Steiner, Tobias

## CHAPTER 10

---

### Contact

---

It's easy to find the email addresses of Quod Libet's authors and maintainers, but please don't email us directly unless asked to, or if it concerns us specifically. If you contact just us, it means no one else can help you, and we can be really bad about replying to email.

If you have a question or want to report a bug please [open a new issue](#) on the GitHub issue tracker.

If you want to address the wider Quod Libet community consider sending a mail to our [quod-libet-devel Google group](#).

Quod Libet has an IRC channel on [OFTC](#) , [#quodlibet](#). It is a good place to ask for help with installation and configuration, or discuss development (assuming people there are paying attention).

There is also now a [Quod Libet Gitter chatroom](#) too, for people that prefer that.