

# EasyTAG: cross-platform audio tagging

David King <amigadave@amigadave.com>

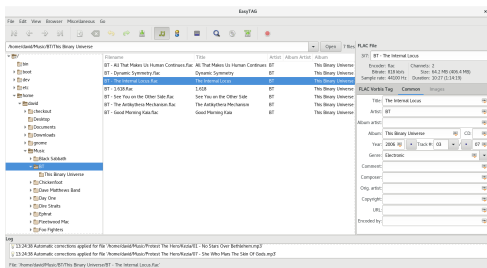
22nd May 2016 / GNOME.Asia

Licensed under CC0-1.0

[http://amigadave.com/presentations/gnome\\_asia\\_easytag\\_2016.pdf](http://amigadave.com/presentations/gnome_asia_easytag_2016.pdf)



# A simple, but old application



- First released in May 2000
- Initially used GTK+ 1 (now using 3) and written in C
- Initially only supported ID3v1.0 tags (MP3 files), now lots more
- Windows version first released in 2005
- One passionate and active maintainer until 2008



# Adding Windows support to a GTK+ application

- Start playing with MinGW packages on Fedora
- Most dependencies packaged for MinGW, only missing some tagging support
- Share the same build system for cross-compiling
- Package several dependencies, including id3lib
- Create a Windows installer using NSIS



## Taking advantage of GTK+ and GLib

- GTK+ and GLib have good portability to Windows and Mac OS X
- Building for Windows is easy from Linux using MinGW
- Building for Mac OS X requires a Mac
- Lots of helper functions for Windows filename support
- Possible to share the same build system with almost no `ifdefs`!



## Drawbacks of GTK+ and GLib

- Theming is common across all platforms
- Windows and Mac OS X support lag behind Wayland/X11 support
- Support for older versions of Windows was recently dropped
- Relatively limited feature set, so more advanced applications may have to package extra dependencies



# Stability of GTK+ across distributions

- Several LTS distributions with different release cycles
- Many different versions of GTK+ and dependencies: different default themes, differing levels of platform integration
- Theming is quite different between minor GTK+ versions
- xdg-app should help a lot here!



## Further resources

- Wiki: <https://wiki.gnome.org/Apps/EasyTAG>
- Ubuntu PPA:  
<https://launchpad.net/~amigadave/+archive/ubuntu/xdg-app>
- Mailing list: [easytag-list@gnome.org](mailto:easytag-list@gnome.org)



## Building app bundles with RPMs

- Good way to bootstrap app bundles with extra dependencies
- Relies on a modified `rpmbuild` which builds into `/self`
- Additionally, uses a special “var runtime” to provide a writable RPM database corresponding to the SDK contents

