

Gentoo & KDE

Packages, Compilation & Interaction

Marcus D. Hanwell
cryos@gentoo.org

Gentoo Linux

aKademy 2007
1 July, 2007



Outline

Introduction

- Outline

- What Is Gentoo Linux?

- Why Build From Source?

- The KDE Herd

KDE Core Applications & Libraries

- Core Applications & Libraries

- Extra KDE Applications

- Splitting Up KDE

- DBus & HAL—Conflict Between Gnome & KDE

Interaction

- Upstream KDE Developers

- Other Linux Distributions

- Distribution Applied Patches/Customisations

- How Can We Improve?

KDE 4—The Future

- Getting To Grips With CMake

- Automagic Is Bad

- Splits & Monolithic

- Split The Source Too?



What Is Gentoo Linux?

- ▶ Gentoo Linux is a source based distribution
- ▶ Large repository of ebuilds to facilitate building from source
- ▶ The ebuild repository updates from the CVS tree live
- ▶ Portage tracks dependencies, downloads source and compiles
- ▶ Recently rated 9/10 by Linux Format for the 2007.0 release
 - ▶ Large range of supported hardware architectures
 - ▶ Emphasis on providing choice through USE flags
 - ▶ Compile in only the features you want



Why Build From Source?

- ▶ Why build from source when
 - ▶ It is time consuming
 - ▶ Aggressive compiler flags cause weird bugs, `-fomg-faster`
 - ▶ Automagic build files break dependency tracking
- ▶ Each binary can be custom built
 - ▶ Compiler flags optimise to use the full power of the CPU
 - ▶ USE flags allow custom binaries to be compiled
 - ▶ Gives more power to users rather than packagers
- ▶ A large array of possible builds are possible from one source package



The KDE Herd

- ▶ The KDE herd takes care of KDE applications and libraries
- ▶ Gentoo and the KDE herd is run by volunteers
- ▶ Small team managing a large number of ebuilds
- ▶ Gentoo is desktop environment agnostic—can be tough striking the right balance
- ▶ Always on the look out for volunteers and better ways to collaborate



Core Applications & Libraries

- ▶ The core applications are kept in kde-base
- ▶ Comprise all applications and libraries part of the main release
- ▶ Available in monolithic or split ebuilds
- ▶ All are built from the main KDE release tarballs
- ▶ Maintain minimal patch sets and aim to get patches included upstream where possible
- ▶ Stay as close to upstream sources as possible



Extra KDE Applications

- ▶ Lots of other very important KDE applications

k3b killer burning application

amarok awesome music application

digikam amazing photo editing/management

kile for those \LaTeX junkies out there

- ▶ And lots more. . .



Extra KDE Applications continued...

- ▶ Present some unique challenges to effectively package
- ▶ Go into the main FHS hierarchy whereas KDE is outside it
- ▶ Can be difficult to get linking right at times
- ▶ Binary compatibility in KDE helps a lot
- ▶ Is this the optimal way to handle things?



Splitting Up KDE

- ▶ A few years ago split ebuilds were introduced to Gentoo
- ▶ Individual KDE core applications such as konsole could be compiled
- ▶ Improved granularity for dependencies—konsole instead of kdebase!
- ▶ So far we have maintained old monolithic and new split
- ▶ Many other distros only have the equivalent to split
- ▶ Should we continue supporting both?



DBus & HAL—Conflict Between Gnome & KDE

- ▶ DBus and HAL are used by Gnome and KDE
 - ▶ The API has not always been stable
 - ▶ This has led to big issues maintaining Gnome & KDE
 - ▶ Led to delayed stabilisation and forced patching
- ▶ As more is shared between DEs these types of problem are likely to become more of an issue
- ▶ Can be avoided by only using stable APIs and/or coordination
- ▶ Led to some interesting interaction between our herds!



Upstream KDE Developers

- ▶ How do we currently interact?
 - ▶ IRC
 - ▶ Mailing lists
 - ▶ Bugzilla
 - ▶ Blogs
 - ▶ Conferences
- ▶ Do we need to improve communication?
- ▶ What is the optimal way to request urgent patches or ask questions?



Other Linux Distributions

- ▶ Could we share patches more effectively?
- ▶ Shouldn't patches be pushed upstream anyway?
- ▶ Could we better coordinate security and other bug fixes?
- ▶ Does LSB makes any sense for a source distribution?
- ▶ Could we be more open?



Distribution Applied Patches/Customisations

- ▶ Traditionally Gentoo has always had a philosophy of using minimal patch sets
- ▶ We have struggled with the branding question
- ▶ More and more packages are getting branding
- ▶ KDE is following this trend
 - ▶ Gentoo is about choice
 - ▶ Branding is controlled by a USE flag
- ▶ How much should we patch and customise KDE?



How Can We Improve?

- ▶ Going forward how could we improve interactions?
- ▶ The open source world is filled with passionate people
 - ▶ Don't let it get personal
 - ▶ Disagreements should try to remain technical
 - ▶ Most of us are volunteers—time is in short supply
- ▶ Interaction can be tough to get right
 - ▶ Don't be an island



Getting To Grips With CMake

- ▶ We have a lot of experience with autotools
- ▶ Docs for CMake aren't always what they could be
- ▶ Assignment of lib dir isn't a built-in anymore
- ▶ CMake ships a lot of built in libs which is bad for security
- ▶ Hopefully all these things will improve as CMake matures
- ▶ Coloured output and progress indications are nice
- ▶ Syntax seems nicer albeit less well documented



Automagic Is Bad

If there is one thing you remember about my talk:

- ▶ Automagic detection of optional deps is bad!
 - ▶ Deterministic build systems are essential
 - ▶ Enable and disable flags should be provided
 - ▶ Automagic detection and inclusion breaks the dependency tree
 - ▶ This affects Gentoo quite badly but all binary distros have similar issues with automagic detection—unpredictable results
- ▶ Still getting to know CMake, autotools had `-with-*` and `-without-*` or `-enable-*` `-disable-*`
- ▶ One of the most difficult problems for us to fix with builds—it makes Larry grumpy!



Splits & Monolithic

We find ourselves asking the question whether we should maintain both the split ebuilds and the old style monolithic ebuilds. As far as we can tell both are popular. Traditionally we kept everything the same as upstream—one source package = one ebuild.

- ▶ Should we keep monolithic ebuilds
- ▶ Are the split ebuilds still possible?
 - ▶ Initial work has taken place in overlays
 - ▶ New build system
 - ▶ Lots of changes



Split The Source Too?

- ▶ Split ebuilds effectively build a custom source tree at unpack
- ▶ Building split ebuilds from monolithic source
 - ▶ Multiple monolithic source tarballs required
 - ▶ To install kuickshow all of kdegraphics is needed
 - ▶ Quite wasteful of resources
 - ▶ Binary distros split their packages
- ▶ Could be reduced by improving portage—ranged source
- ▶ Haven't yet conducted tests to see if this approach would work as expected



The End

- ▶ I would like to thank you all for listening
- ▶ This presentation was created in \LaTeX using the beamer class
- ▶ Gentoo is all about choice, we put a lot of work into offering as much as we can to all of our users. . .
- ▶ I welcome your input
- ▶ Questions?

