

## The saga of official binary packages for Gentoo Linux

Andreas K. Hüttel (dilfridge) & Sam James (sam)



FOSDEM 2026, Brussels, 1 February 2026

## about us & outline

Andreas K. Hüttel (dilfridge@g.o)

- Gentoo developer since 2010
- background in physics
- Gentoo Council member
- toolchain, release engineering, binhost, base system, perl, libreoffice, PR, ...

Sam James (sam@g.o)

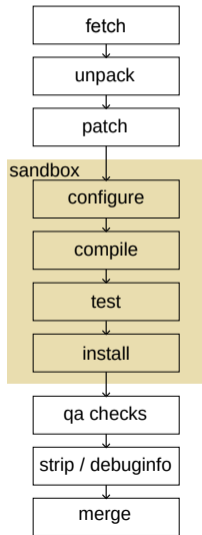
- Gentoo developer since 2020
- background in maths / CS
- Gentoo Council member
- toolchain, binhost, base system, QA, security, python, LLVM, maths, ...

hello there! — backstory — current implementation — future improvements



## “emerging” a package from source

- package manager: Portage
- the main command is “emerge”
- fetch, unpack, patch sources
- configure, compile, install to temporary directory
- QA checks
- strip binaries, separate out debug information
- merge files into main system
- not *that* much different from other package builders (of course, this omits a lot of details)
- dependencies provided by the main system



# xpak — old-style Gentoo binary packages

- compressed tar archive
- custom trailer for metadata
- advantages:
  - single file
  - tar can unpack it
- disadvantages:
  - custom binary format
  - relies on obscure tar behaviour
  - requires dedicated tools
  - no gpg signature integration
  - no specification versioning, not easily extensible

## SYNOPSIS

binpkg (tbz2)

```
|<-xpak_offset->|  
<tar>|<  xpak  >|<xpak_offset>"STOP"
```

xpak

```
"XPAKPACK"<index_len><data_len><index><data>"XPAKSTOP"
```

index

```
|<-----index_len----->|  
|<index1><index2><index3><index4><...>|
```

indexN

```
|<-name_len->|  
<name_len>|<  name  >|<data_offset><data_len>
```



## no global rebuilds, ever!

- in Gentoo, compilation takes place on a user system, CPU cycles are limited  
→ no global rebuilds (“mass rebuilds”) of all packages, ever!
- need to make sure in other ways that all installed packages are consistent
- examples: library upgrade changes soversion, Perl update changes module directories, ...

```

huettel@kona /usr/lib64/perl5/vendor_perl $ ls 5.42/
Algorithm  Business  Digest    Git        JSON       MailTools.pod  MRO        Pegex      Rex        Spreadsheet  Tie        XXX.pod
aliased.pm  Canary    Dist      Git.pm     JSON.pm    Math           namespace  Pegex.pm   Rex.pm     StackTrace  Time       YAML
Alien      Capture  Encode    Gtk3.pm    Lab        MCE           NaturalDocs  Pegex.pod  Role       Statistics   TimeDate.pm  YAML.pm
alienfile.pm  Carp     Error     Hash       Lexical    MCE.pm        Net         Perl       RPC        String      Try         YAML.pod
App        Class    Error.pm  Hook       Lingua     MCE.pod       Number      PerlIO     Safe       Sub         Type
Archive    Clone    Eval      HTML       List       Method        OLE         Pod        Scope     Syntax      Types
Astro      Config  Exception HTTP       Locale     MIME          oo.pm       POD2       SGMLS     syntax.pm   URI
Authen     Convert  Expect.pm Image      Log        Mixin        Package     PPI        sgm1spl-specs  Sys       URI.pm
auto       CPAN     Exporter  inc        LWP        Module        PAR         PPI.pm    SGMLS.pm  Task       USB
B          Crypt   ExtUtils  Inline     lwpcook.pod  Moo          Parallel    PPIx      Shell     Term       Version
Biber     Data    FFI       Inline.pm  LWP.pm     Moo.pm        Params      Readonly.pm  Software  Test       WWW
Biber.pm  Date    File      Inline.pod lwptut.pod  Moose       Parse       Ref        Sort      Text       x86_64-linux
Browser   DateTime  Gentoo   IO         Mail       MooseX       Path        Regexp     Specio    Throwable  XML
Bundle    Devel   Getopt    IPC        MailTools.pm  Mozilla      PDL         Reply     Specio.pm  Throwable.pm  XXX.pm

```



## dangers of dependency automagic

- installed package set on the builder can differ from the package target environment
- example: OpenEXR installed on builder, but not on target
- example: Different ICU versions on builder and target
- build systems can create “automagic dependencies” on libraries (and other dependencies) found at build time
- autoconf is actually a *good* example, since the magic can always be overridden there

```
checking for jpeg_enable_lossless... yes
checking for jpeg_simple_lossless... no
-----
checking for libjxl >= 0.7.0 libjxl_threads... yes
-----
checking for lcms2 >= 2.0.0... yes
checking for lcms2/lcms2.h... no
-----
checking for OpenEXR >= 1.0.6... yes
-----
checking for pangocairo >= 1.28.1... yes
checking for pango >= 1.28.1... yes
-----
checking for libpng >= 1.0.0... yes
-----
checking for librsvg-2.0 >= 2.9.0... yes
checking for cairo-svg... yes
-----
checking for libtiff-4 >= 4.0.0... yes
-----
checking for libuhdr >= 1.3.0... no
-----
checking for libwebp >= 0.4.1... yes
checking for libwebpmux >= 0.5.0 libwebpdemux >= 0.5.0... yes
checking if WMF package is complete... no
checking for libxml-2.0 >= 2.0.0... yes
```



## old-style Gentoo helpers (deprecated)

### revdep-rebuild

- find binaries with broken dynamic linking
- identify affected packages
- rebuild affected packages

### perl-cleaner

- find modules installed in `vendor_perl` of previous Perl 5 versions
- find binaries linking to `libperl.so`
- identify affected packages
- rebuild affected packages

revdep-rebuild, perl-cleaner try to fix what is already broken :(

### preserved libraries mechanism

- on removal of dynamic libraries, check if anything still dynamically links to them
- keep such libraries, re-assign them to their reverse dependencies
- record affected packages in `@preserved-rebuild` package set

### emerge @preserved-rebuild

- rebuild packages depending on preserved libraries
- since headers etc. are gone / updated, package will afterwards use newer library version or stop using library



## subslot rebuild mechanism

- package depends with specific operator “:=” on library
- library declares “subslot”
- “any subslot fulfils the dependency, but if it changes we need a rebuild”
- examples for subslot values:
  - library soversion  
(e.g. readline, SLOT=0/8)
  - library major version  
(e.g. ICU, SLOT=0/77)
  - package major version in general  
(e.g. Perl, SLOT=0/5.42)
  - ...

```
inherit cmake optfeature xdg

DESCRIPTION="Modern music player and library organizer based on Clementine and Qt"
HOMEPAGE="https://www.strawberrymusicplayer.org/"
# [...]

LICENSE="GPL-3 discord? ( MIT )"
SLOT="0"
# [...]

COMMON_DEPEND="
    dev-db/sqlite:3
    dev-libs/glib:2
    dev-libs/icu:=
    dev-libs/kdsingleapplication[qt6(+)]
    dev-qt/qtbase:6[concurrent,dbus?,gui,network,ssl,sql,sqlite,widgets,X?]
# [...]
```

```
EAPI=8

MY_PV=${PV/_rc/-rc}
PYTHON_COMPAT=( python3_{11..14} )
VERIFY_SIG_OPENPGP_KEY_PATH=/usr/share/openpgp-keys/icu.asc
inherit autotools flag-o-matic multilib-minimal python-any-r1 toolchain-funcs verify-sig

DESCRIPTION="International Components for Unicode"
HOMEPAGE="https://icu.unicode.org/"
# [...]

LICENSE="BSD"

# Subslot is the major package version, up to first .
SLOT="0/${PV%.*}"
# [...]
```





## quality control: fully specify dependencies

- build must not depend on presence / absence / version of packages at build time *beyond declared dependencies*

```
LICENSE="imagemagick"
# Please check this on bumps, SONAME is often not updated! Use abidiff on old/new.
# If ABI is broken, change the bit after the '-'.
SLOT="0/${ver_cut 1-3}-18"
IUSE="bzip2 corefonts +cxx djvu fftw fontconfig fpx graphviz hardened hdri heif"
IUSE+=" jbig jpeg jpeg2k jpegxl lcms lqr lzma openc1 openexr openmp pango perl"
IUSE+=" +png postscript q32 q8 raw static-libs svg test tiff truetype webp wmf"
IUSE+=" X xml zip zlib"

# [...]

RDEPEND="
!media-gfx/graphicsmagick[imagemagick]
dev-libs/libltdl
bzip2? ( app-arch/bzip2 )
corefonts? ( media-fonts/corefonts )
djvu? ( app-text/djvu )
fftw? ( sci-libs/fftw:3.0 )
fontconfig? ( media-libs/fontconfig )
fpx? ( >=media-libs/libfpx-1.3.0-r1 )
graphviz? ( media-gfx/graphviz )
heif? ( media-libs/libheif:=[x265] )
jbig? ( >=media-libs/jbigkit-2:= )
jpeg? ( media-libs/libjpeg-turbo:= )
jpeg2k? ( >=media-libs/openjpeg-2.1.0:2 )
```

```
local myeconfargs=(
    $(use_enable static-libs static)
    $(use_enable hdri)
    $(use_enable openc1)
    $(use_enable openmp)
    --with-threads
    --with-modules
    --with-quantum-depth=${depth}
    $(use_with cxx magick-plus-plus)
    $(use_with perl)
    --with-perl-options='INSTALLDIRS=vendor'
    --with-gs-font-dir=${EPREFIX}/usr/share/fonts/urw-fonts
    $(use_with bzip2 bzlib)
    $(use_with X x)
    $(use_with zip)
    $(use_with zlib)
    --without-autotrace
    --with-uhdr
    $(use_with postscript dps)
    $(use_with djvu)
    --with-dejavu-font-dir=${EPREFIX}/usr/share/fonts/dejavu
    $(use_with fftw)
    $(use_with fpx)
    $(use_with fontconfig)
    $(use_with truetype freetype)
    $(use_with postscript gslib)
# [...]
)

CONFIG_SHELL="${BROOT}/bin/bash econf "${myeconfargs[@]}"
```



# new binary package format — gpkg (GLEP 78)

- single file encapsulation
- standard nested tarballs, no tricks
- identifiable by *file*
- metadata separate from image
- Manifest OpenPGP signature
- detailed specification: GLEP 78

poppler-25.12.0-1.gpkg.tar

poppler-25.12.0-1/	
gpkg-1	identifier, req
metadata.tar\${comp}	required
metadata.tar\${comp}.sig	optional
image.tar\${comp}	required
image.tar\${comp}.sig	optional
Manifest	req, cleartext
	sig optional

relative order of the archive members should be preserved  
 POSIX ustar format as defined by POSIX.1-2017 [3] or a subset of the ustar-compatible GNU  
 tar format as described in the GNU tar manual [4] with the following (optional) extensions ...

## Gentoo's de-facto certification authority

Key fingerprint	Gentoo package	Description	Created	Expiry
<b>13EBBDBEDE7A12775DFDB1BABB572E0E2D182910</b>	sec-keys/openpgp-keys-gentoo-release	Gentoo Linux Release Engineering (Automated Weekly Release Key)	2009-08-25	2026-07-01
<b>DCD05B71EAB94199527F44ACDB6B8C1F96D8BF6D</b>	sec-keys/openpgp-keys-gentoo-release	Gentoo ebuild repository signing key (Automated Signing Key)	2011-11-25	2026-07-01
<b>EF9538C9E8E64311A52CDEDA13D0EF1914E7A72</b>	sec-keys/openpgp-keys-gentoo-release	<a href="#">Gentoo repository mirrors (GitHub)</a> (automated git signing key)	2018-05-28	2026-07-01
<b>D99EAC7379A850BCE47DA5F29E6438C817072058</b>	sec-keys/openpgp-keys-gentoo-release	Gentoo Linux Release Engineering (Gentoo Linux Release Signing Key)	2004-07-20	2027-06-01
<b>ABD00913019D6354BA1D9A132839FE0D796198B1</b>	sec-keys/openpgp-keys-gentoo-auth	Gentoo Authority Key L1	2019-04-01	2026-07-01
<b>18F703D702B189591373148C55D3238EC050396E</b>	sec-keys/openpgp-keys-gentoo-auth	Gentoo Authority Key L2 for Services	2019-04-01	2026-07-01
<b>2C13823B8237310FA213034930D132FF0FF50EEB</b>	sec-keys/openpgp-keys-gentoo-auth	Gentoo Authority Key L2 for Developers	2019-04-01	2026-07-01
<b>ABA5E4E7F4E407ABE9CA7EC7422C9066E21F705A</b>		Gentoo Authority Key L2 for Infrastructure	2024-04-19	2026-07-01



## client configuration — local trust anchor

- small utility `getuto`, called by the package manager
- if `/etc/portage/gnupg` does not exist yet,
  - generates local trust anchor key
  - locally signs Gentoo Release Engineering key(s)
- the key signature(s) define valid package signatures
- obviously, many tuning and configuration options for experts

```
pinacolada ~ # getuto
* Initializing /etc/portage/gnupg ...
gpg: Generating Portage local OpenPGP trust key
gpg: done
```



## package building — Gentoo in a systemd-nspawn

- surprisingly boring
- Gentoo in several systemd-nspawn directories
- btrfs subvolumes for snapshotting
- daily updates
- consuming and building packages
- shared package directory

```
nspawn_command=(systemd-nspawn --bind /dev/steve --bind-ro /etc/sandbox.d/90steve
--bind /var/cache/distfiles
--bind /var/lib/machines/${NSPAWN_PKGDIR}:/var/cache/binpkgs
--bind-ro /var/db/repos/gentoo
[-tmpfs=/var/tmp:mode=1777,size=${TMPFS_SIZE}g)

for n in "${NSPAWN_NAMES[@]}" ; do

    machinedir=/var/lib/machines/${n}

    echo Machine ${n} &>> ${TMPFILE}
    echo &>> ${TMPFILE}

    cd ${machinedir}/etc/binhost &>> ${TMPFILE}
    git reset --hard &>> ${TMPFILE}
    git pull &>> ${TMPFILE}

    cd /root &>> ${TMPFILE}

    "${nspawn_command[@]}" -M ${n} /root/bin/run-update &>> ${TMPFILE}
```



## steve the jobserver



- avoid contention, while still using maximal parallelism
- imagine 4 x webkit-gtk in parallel
- new development, recently introduced
- global, token-based jobserver for make (+ ninja now too)



## several configurations, also via ephemeral nspawnns

- kde, gnome, openrc, server
- different Python versions to build against
- a random lucky package from Gentoo stable
- a ton of gcc and clang versions (yes we target developers!)
- ...
  
- different USE-flag settings generate distinct binary packages
- different dependency specifications generate distinct binary packages
- different dependency subslots generate distinct binary packages
  
- essentially, it's a bit like NixOS without S3 ;)



## several configurations, also via ephemeral nspawnns

- kde, gnome, openrc, server
- different Python versions to build against
- a random lucky package from Gentoo stable
- a ton of gcc and clang versions (yes we target developers!)
- ...
  
- different USE-flag settings generate distinct binary packages
- different dependency specifications generate distinct binary packages
- different dependency subslots generate distinct binary packages
  
- essentially, it's a bit like NixOS without S3 ;)



## instruction set, CFLAGS, ...

- amd64: two identical builders with `-march=x86-64` and `-march=x86-64-v3`
- arm64: baseline aarch64 ISA
- `CFLAGS='-O2 -pipe -march=x86-64 -mtune=generic  
-Werror=odr -Werror=lto-type-mismatch -Werror=strict-aliasing'`
- `CXXFLAGS`, `FCFLAGS`, `FFLAGS` identical
- system packages: in addition, use LTO (`-flto`)
- we know that these are conservative and safe settings — for convenience and stability
- you can still compile with `-Omg-so-fast -funroll-loops` (if that makes your day)



## instruction set, CFLAGS, ...

- amd64: two identical builders with `-march=x86-64` and `-march=x86-64-v3`
- arm64: baseline aarch64 ISA
- `CFLAGS='-O2 -pipe -march=x86-64 -mtune=generic  
-Werror=odr -Werror=lto-type-mismatch -Werror=strict-aliasing'`
- `CXXFLAGS`, `FCFLAGS`, `FFLAGS` identical
- system packages: in addition, use LTO (`-flto`)
- we know that these are conservative and safe settings — for convenience and stability
- you can still compile with `-Omg-so-fast -funroll-loops` (if that makes your day)



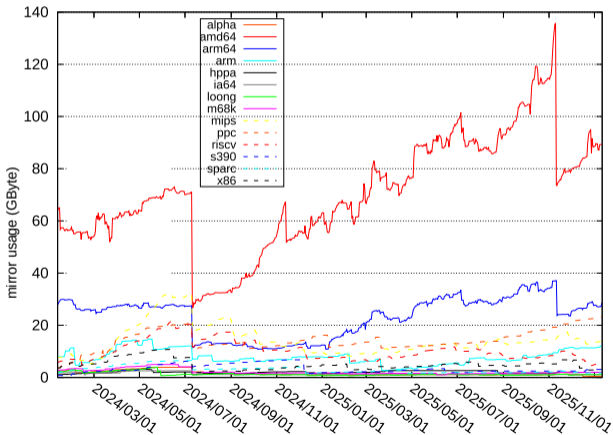
# hardware and output ...

## x86-64:

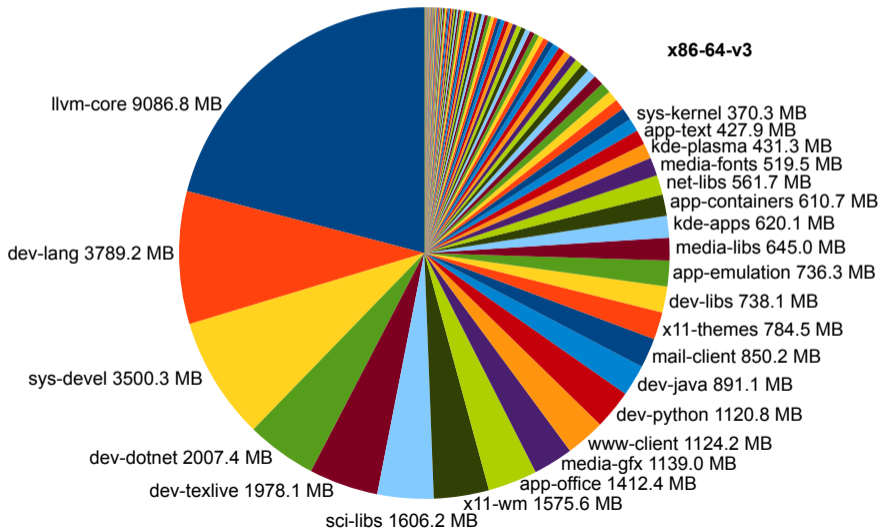
- two servers at Hetzner (AX102)
- AMD Ryzen 9 7950X3D, 16 Zen4 cores, 128G ECC-RAM, 4T NVME
- also used for stage builds (native and qemu-based)

## aarch64:

- server at OSUOSL (Amp. Altra Max)
- Neoverse-N1, 96 Armv8.2+ cores, 256G RAM, 4T NVME
- Prov. by Arm Ltd., "Works on Arm"
- also used for stage builds (arm64 and arm) and development

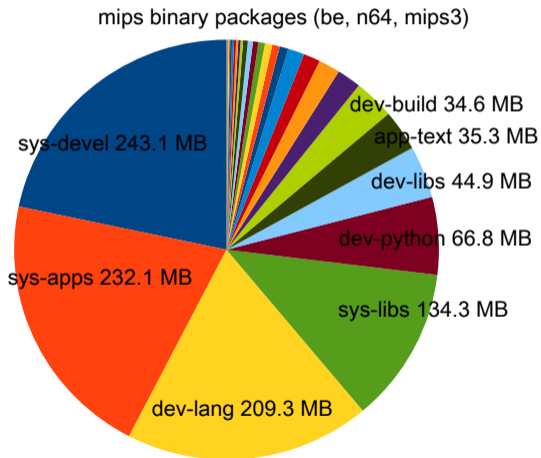


# ... across package categories



## less popular systems — reusing the catalyst cache

- alpha, hppa, mips, sparc, riscv, ...?
- musl, Gentoo Hardened, LLVM builds, ...?
- limit to installation stage content, @system set
- “base ISA”
- export cache of stage builder
- still a lot in total
  - mips: **20 directories**  
be/le, o32/n32/n64, t32/t64, ...
  - riscv: **9 directories**  
rv64/rv32, lp64d/lp64, ...
  - ...



## default configuration

- binary server preset in installation stage (base ISA)
- default binary package format gpgkg
- infrastructure for package signature verification is in place
- “emerging” a package: source build by default, exactly as before
- with the switches `-g` or `--getbinpkg` this changes however

```
[binary R ] app-editors/nano-8.7-2 USE="-verify-sig%"
[binary R ] sys-apps/less-685-2 USE="-verify-sig%"
[ebuild R ] app-misc/colordiff-1.0.21 USE="-verify-sig%"
[binary U ] sys-apps/util-linux-2.41.3-1 [2.41.2]
[ebuild U ~] dev-libs/icu-78.2 [78.1]
[binary U ] dev-libs/libxslt-1.1.45-1 [1.1.43-r2]
[binary R ] sys-fs/e2fsprogs-1.47.3-r1-2 USE="-verify-sig%"
[binary R ] sys-apps/pciutils-3.14.0-4 USE="-verify-sig%"
[ebuild U ] dev-debug/strace-6.18 [6.17]
[ebuild U ] app-arch/libdeflate-1.25 [1.24]
[binary U ] sys-fs/cryptsetup-2.8.3-1 [2.8.1-r1]
[ebuild U ] app-il8n/man-pages-l10n-4.28.0 [4.27.0-r1]
[binary U ] dev-python/trove-classifiers-2025.12.1.14-1 [2025.11.14.15]
[binary U ] dev-python/filelock-3.20.1-1 [3.20.0] USE="-verify-provenance%"
[ebuild U ] sys-kernel/dracut-108-r5 [108-r4] USE="systemd%*"
[ebuild U ] sec-keys/openpgp-keys-gentoo-developers-20260112 [20260105]
```



## selectivity of emerge

- binary package use controllable on command line or in make.conf
- build dependencies only needed for source builds
- it's possible to finetune which binary packages are used and which not
  - `--binpkg-respect-use=y/n`  
Should we consider binary packages that have different USE-flags (feature specifications)?  
*imagine you need libreoffice, and you don't care whether it has pdf import enabled or not*
  - `--binpkg-changed-deps=y/n`  
Should we consider binary packages where the dependency specification in the ebuild has changed in the meantime?
- safe defaults in the package manager



## implicit libc dependencies

- downgrading glibc is a sure way to destruction
- no package built against a later glibc version will run with an earlier one
- build logic inserts a hard version dependency into binary package metadata

```
* Checking general environment sanity.  
* Sanity check to keep you from breaking your system:  
* Downgrading glibc is not supported and a sure way to destruction.  
* ERROR: sys-libs/glibc-2.41-r9::gentoo failed (unpack phase):  
* Aborting to save your system.  
*  
* Call stack:  
* ebuild.sh, line 143: Called src_unpack  
* environment, line 3838: Called sanity_prechecks  
* environment, line 3490: Called die  
* The specific snippet of code:  
* [[ ${I_ALLOW_TO_BREAK_MY_SYSTEM} = yes ]] || die "Aborting to save your system.";
```



## future — fine-grained binary usage

- for now, binary packages switched on / off globally
- planned: disable / enable per package — e.g., only qtwebengine, firefox :)
- contributed feature, WIP by new contributor Jethro Donaldson

```
Calculating dependencies... done!
Dependency resolution took 2.29 s (backtrack: 0/20).

[binary N ] virtual/minizip-1.3.1-1:0/1::gentoo USE="--static-libs" ABI_X86="(64) -32 (-x32)" 0 KiB
[binary N ] dev-libs/nspr-4.37-1::gentoo USE="(debug)" ABI_X86="(64) -32 (-x32)" 0 KiB
[binary N ] app-arch/snappy-1.2.2-r1-1:0/1.1::gentoo USE="--test" ABI_X86="(64) -32 (-x32)" CPU_FLAGS_X86="--avx -avx2" 0 KiB
[binary N ] dev-libs/nss-3.112.2-1::gentoo USE="utils -cacert -test -test-full" ABI_X86="(64) -32 (-x32)" CPU_FLAGS_X86="--avx2 -sse3" 0 KiB
[binary N g ] sys-apps/pciutils-3.14.0-4::gentoo USE="kmod udev zlib -dns -static-libs -verify-sig" ABI_X86="(64) -32 (-x32)" 200 KiB
[binary N ] dev-qt/qtwebchannel-6.10.1-1:6/6.10.1::gentoo USE="qml -custom-cflags -test" 0 KiB
[binary N g ] dev-qt/qtwebengine-6.10.1-4:6/6.10.1::gentoo USE="alsa bindist jumbo-build opengl pdfium pulseaudio qml screencast system-icu vulkan
i -webdriver" 85530 KiB

Total: 7 packages (7 new, 7 binaries), Size of downloads: 85730 KiB
```

```
amd64-stable ~ # emerge -pvlg --getbinpkg-include=dev-qt/qtwebengine dev-qt/qtwebengine
[gentoo] Local copy of remote index is within TTL and will be used.

These are the packages that would be merged, in order:

Calculating dependencies... done!
Dependency resolution took 1.89 s (backtrack: 0/20).

[ebuild N ] virtual/minizip-1.3.1:0/1::gentoo USE="--static-libs" ABI_X86="(64) -32 (-x32)" 0 KiB
[ebuild N ] dev-libs/nspr-4.37::gentoo USE="(debug)" ABI_X86="(64) -32 (-x32)" 1033 KiB
[ebuild N ] dev-libs/nss-3.112.2::gentoo USE="utils -cacert -test -test-full" ABI_X86="(64) -32 (-x32)" CPU_FLAGS_X86="--avx2 -sse3" 74828 KiB
[ebuild N ] sys-apps/pciutils-3.14.0::gentoo USE="kmod udev zlib -dns -static-libs -verify-sig" ABI_X86="(64) -32 (-x32)" 0 KiB
[ebuild N ] app-arch/snappy-1.2.2-r1:0/1.1::gentoo USE="--test" ABI_X86="(64) -32 (-x32)" CPU_FLAGS_X86="--avx -avx2" 0 KiB
[ebuild N ] dev-qt/qtwebchannel-6.10.1-1:6/6.10.1::gentoo USE="qml -custom-cflags -test" 0 KiB
[binary N g ] dev-qt/qtwebengine-6.10.1-4:6/6.10.1::gentoo USE="alsa bindist jumbo-build opengl pdfium pulseaudio qml screencast system-icu vulkan widgets
i -webdriver" 85530 KiB

Total: 7 packages (7 new, 1 binary), Size of downloads: 161390 KiB
```



## future — debug information

- no build-id in Gentoo so far (limited usefulness for source distribution)
- now, (not only) for binary packages
- packdebug support in portage (3.0.74)
- planned debuginfod infrastructure

```
/usr/lib/debug/.tarball/  
├── app-accessibility  
│   ├── at-spi2-core  
│   │   ├── at-spi2-core-2.56.5-23-debug.tar.xz  
│   │   ├── at-spi2-core-2.56.5-24-debug.tar.xz  
│   │   └── at-spi2-core-2.56.5-25-debug.tar.xz  
│   ├── espeak-ng  
│   │   ├── espeak-ng-1.52.0-34-debug.tar.xz  
│   │   ├── espeak-ng-1.52.0-35-debug.tar.xz  
│   │   └── espeak-ng-1.52.0-36-debug.tar.xz  
│   └── speech-dispatcher  
│       ├── speech-dispatcher-0.12.1-36-debug.tar.xz  
│       ├── speech-dispatcher-0.12.1-37-debug.tar.xz  
│       └── speech-dispatcher-0.12.1-38-debug.tar.xz  
├── app-admin  
│   ├── apache-tools  
│   │   ├── apache-tools-2.4.66-6-debug.tar.xz  
│   │   ├── apache-tools-2.4.66-7-debug.tar.xz  
│   │   └── apache-tools-2.4.66-8-debug.tar.xz  
│   ├── chrpath  
│   │   ├── chrpath-0.18-4-debug.tar.xz  
│   │   ├── chrpath-0.18-5-debug.tar.xz  
│   │   └── chrpath-0.18-6-debug.tar.xz  
│   └── entr  
│       ├── entr-5.7-50-debug.tar.xz  
│       ├── entr-5.7-51-debug.tar.xz  
│       └── entr-5.7-52-debug.tar.xz
```



## future — LTO for everyone (and everything)

mostly a general quality control issue?

```
/tmp/lto $ gcc a.c b.c -flto -Werror=lto-type-mismatch
b.c:5:6: error: type of 'process_foo' does not match original declaration [-Werror=lto-type-mismatch]
  5 | void process_foo (foo myfoo);
    |                   ^
a.c:4:6: note: type mismatch in parameter 1
  4 | void process_foo (foo myfoo) {
    |                   ^
a.c:4:6: note: 'process_foo' was previously declared here
a.c:4:6: note: code may be misoptimized unless '-fno-strict-aliasing' is used
lto1: some warnings being treated as errors
lto-wrapper: fatal error: gcc returned 1 exit status
compilation terminated.
/usr/x86_64-pc-linux-gnu/binutils-bin/9999/ld: error: lto-wrapper failed
collect2: error: ld returned 1 exit status
```

(diagnostics: gcc PR114372, gcc PR116094, gcc PR114218, ...)



## LTO vs static libs

```
lto1: fatal error: bytecode stream in file 'a.o' generated with LTO version 15.1 instead of the expected 16.0
compilation terminated.
lto-wrapper: fatal error: gcc-16 returned 1 exit status
compilation terminated.
```



## LTO vs static libs (2)

- binutils improvements
- dot-a



## LTO vs static libs (3)

```
$ wc -l dot-a.eclass tests/dot-a.sh
128 dot-a.eclass
742 tests/dot-a.sh
870 total
```



# Thanks

## Binhost team

- Eli Schwartz (eschwartz, ztrawhcse)

RinCat for the GLEP 78 + GnuPG integration

Arsen Arsenović (arsen) for packdebug support

## Portage team

- Michał Górny (mgorny)
- John Helmert III (ajak)
- James Le Cuirot (chewi)
- Mike Gilbert (floppym)
- Robin Johnson (robbat2)

## Release Engineering team

- Matt Turner (mattst88)
- Ben Kohler (bkohler, iamben)
- James Le Cuirot (chewi)
- Robin Johnson (robbat2)

## Infrastructure team

- Robin Johnson (robbat2)
- John Helmert III (ajak)
- Arthur Zamarin (arthurzam)
- Michał Górny (mgorny)
- Matthew Thode (prometheanfire)

& of course everyone else who contributed



# Thank you! — Questions?

