# Recommendations from the Gentoo Working Group on improving the state of the stable tree

Gentoo Linux

September 19, 2016

## Contents



# 1 Mandate

This is the final report of the Gentoo Working Group established to come up with recommendations to the Gentoo Council on improving the state of the stable tree[1]

## 1.1 Participants

- k_f (chair)
- blueness
- dilfridge
- rich0
- pacho
- mjo
- kentnl
- kensington
- bircoph
- jmorgan

# 2 Definitions

**Architecture** Supported Gentoo platform, as described in the first field of profiles.desc

**Stable architecture** Architecture that has at least one stable profile, and a maintaining project that determines stable keywords are to be used for that architecture.

**Stable profile** Profile with a "stable" status in profiles.desc. This implies ensure dependency graph consistency, and not necessarily any stable keywords.

# 3 How do we want the stable tree to function

## 3.1 Applications

Maintaining a stable tree is important for providing a known-good option for those who wish to to use Gentoo in a production environment or simply to avoid the potential volatility of always using the "latest and greatest".

There is a careful balance to maintain between length of time in testing and ebuild freshness – too little time in testing and we risk bugs in stable, too much and we risk losing relevance due to excessively outdated ebuilds.

Security relevant issues and significant bug fixes needs to be part of the stable tree, either as a result of a version bump or due to backporting a patch fixing a revision of a version of the package in stable.

## 3.2 Kernel

There should be kernel packages in stable visibility available to end users for *sys-kernel/gentoo-sources* and other kernel variants referenced in the installation instructions of the handbook, in order to ease the configuration for new users.

For Kernel stabilisation it is not possible for an arch tester to test all possible configuration and hardware combinations. Other kernel variants, that are expected for more advanced users to begin with, likely does not make sense to stabilise given there the lack of stability guarantees. Restricing the number of stabilised kernels hence aids in reducing arch tester workload.

As discussed previously[2, 3] a consistent stabilisation policy on kernels is also important in order to ensure that the latest point release is available to end-users in order to ensure that security- and bugfixes are offered. In the experience of the WG the kernel upstream has a strict policy on maintaining backwards compatibility and not breaking userspace which has been proven over time.

As such it makes sense to

- Stabilise Long Term Stable (LTS) branches once they have been determined to be so and tested downstream
- Have an automatic policy of stabilising new point releases within the LTS
- non-LTS branches should not be stabilised

# 4 Current state of the stable tree

There are currently 46 supported architectures, with 9 of those architectures considered stable. Each architecture is maintained by a team consisting of a varying number of developers, but in practice between zero and two developers are actively working on stabilisation requests on any given architecture.

| Archictecture | Outstanding stable requests |
|---|---|
| alpha | 41 |
| amd64 | 265 |
| arm | 45 |
| hppa | 149 |
| ia64 | 209 |
| ppc | 194 |
| ppc64 | 180 |
| sparc | 251 |
| x86 | 310 |

Table 1: Outstanding stable requests per arch as of 2016-08-17

A common developer complaint[citation needed]is the delay between filing a stabilisation request and it being actioned by an arch team. This can block the maintainer from removing old versions that contain bugs, security issues, or that are simply no longer maintained. It may also delay filing of additional stabilisation requests due to dependencies on the still open requests, further multiplying the delay.

A common user complaint[citation needed]is that the latest stable version of package is old. Many developers run full testing (~ arch) systems and forget to or simply do not bother filing stabilisation requests.

## 4.1 Current policies

Current stabilisation policies can be found in the devmanual[4] and GLEP 40[5]. It states: For a package to move to stable, the following guidelines must be met:

- The package has spent a reasonable amount of time in testing (~ arch) first. Thirty days is the usual figure, although this is clearly only a guideline. For critical packages, a much longer duration is expected. For small packages which have only minor changes between versions, a shorter period is sometimes appropriate.
- The package must not have any non-arch dependencies.
- The package must not have any severe outstanding bugs or issues.
- The package must be widely tested.
- If the package is a library, it should be known not to break any package which depends upon it.

# 5 What methods can we use to migrate between current and wanted stable states

## 5.1 Targeted architectures

We need to evaluate the architectures we currently target for stabilisation and consider their ongoing relevance, hardware availability, general interest, and targeted audience. Options for any given architecture may include:

- Switching the profile to dev/exp and leaving keywords intact (this option has previously been used as the result of a Council decision)
- Removing all stable keywords while retaining a stable profile (to keep dependency graph consistency)
- Remove stable keywords from all non-core packages
- Review the suitability of stable keywords with each new stable request, dropping as necessary (this option has previously been self-initiated by arch teams)

## 5.2 Bugzilla workflow

The following workflow discussion regards the Gentoo package tree only, and only affects packages that have versions with stable visibility. It is further limited to where a the bug in question affects a version in stable. In the event of a testing (~ arch) only package, or where the bug is in a branch or version that has not yet been stabilized, the workflow will not be adjusted and a maintainer would directly mark a bug as RESOLVED:FIXED.

For other packages, there currently exists the InVCS keyword, defined in the current bugzilla configuration as "Fix has been added to a VCS(either CVS, SVN, Git, ...) repository. Will be closed when fixes are applied to a stable level package". Based on anecdotal evidence many developers are not using this keyword (or is even aware of its existence) and hence close bug reports before a fix has reached the stable tree.

Depending on the severity and how many users the bug is likely to affect a lack of tracking the bug until it has reached a stable status can constitute a hinder for both stability- and fresheness expectations of stable users. At the same time, some discretion seems warranted on the part of the maintainer for minor issues without a wider impact to be closed immediately without full tracking.

It is the expectation of this working group that adding an additional bugzilla status element will increase the awareness of the expectation that issues affecting the stable three are not considered resolved before a fix is present for stable users. Such a status element could for example be

named IN_TESTING and logically belongs in the UNCONFIRMED → CONFIRMED (optional) → IN_PROGRESS (optional) → IN_TESTING → RESOLVED workflow.

Historically an additional VERIFIED status has also been used for verification from QA team on the fix, but this was removed[6] to avoid confusion from overloading the use and responses in particular from users that were unsure about whether they should respond to a bug resolution or not. As Gentoo is rolling release without a specific review workflow, having no such state makes sense, and if later workflow should benefit from it, the change of state should be restricted to QA team or others performing reviews.

When changes to fields affecting workflow are implemented the bugzilla field descriptions[7] should be updated to match the new state, and disabled fields should be marked accordingly.

### 5.3 Stabilisation process

Once we have determined which architectures we wish to target for stabilisation, we must decide how we want to handle the stabilisation process. Options to consider may include:

- Waiting period in testing (∼ arch) (30 days)
- Formalising the ALLARCHES policy – better document suitable types of packages (dictionaries, man pages, pure non-compiled language packages...)
- Encourage stabilisation to be performed by maintainers on architectures they have access to – better document exactly what should be checked
- Once a runtime test has been performed on one architecture, stabilise on other architectures with only a build test
- Perform build-only testing, assuming that the 30 day waiting period will smoke out any runtime issues

Note that while these options may seem relaxed compared to the current "official" policy, in reality many stabilisations are currently performed with build-only testing or trivial runtime testing. We also assume the package maintainer is the best qualified person to determine if a package is suitable to be stabilised or not.

### 5.4 Stable tree freshness

Many packages have old stable versions simply because nobody ever filed a stabilisation request. We should encourage maintainers to regularly review their packages and make considered decisions about what should be stabilised. Again, it is important to maintain the balance between stability and freshness – a package that sees weekly releases likely doesn't need every version stabilised, but having a 5-year-old package in stable with bugs fixed in testing (∼ arch) isn't helpful either. To assist, we could consider:

- Reinstating automated stabilisation requests/reminders
- Producing per-maintainer stabilisation candidate lists (eg. imlate[1] output, with a link to open a pre-filled stabilisation bug)
- Destabilising packages that don't make sense in stable

## 6 Recommendations

lorem ipsum...

## References


[1] Announcement of the Gentoo Working Group: https://archives.gentoo.org/gentoo-dev/message/9cc27b25e7e1742653ffeaf77ada5c18

[2] gentoo-kernel: vanilla-kernel sources should not be marked stable for obsolete versions https://archives.gentoo.org/gentoo-kernel/message/4a2209db4f3a57cc96af0db8e007a7c6

[3] gentoo-dev: Gentoo-sources - should we stable? https://archives.gentoo.org/gentoo-dev/message/996b37c1da87caa22dddc993ce35e1c7

[4] Devmanual: Keywording: Moving from arch to arch: https://devmanual.gentoo.org/keywording/index.html#moving-from-~arch-to-arch

[5] GLEP 40: Standardizing "arch" keywording across all archs. https://wiki.gentoo.org/wiki/GLEP:40

[6] gentoo-dev: RFC bugs.g.o: Killing VERIFIED state, possibly introducing STABILIZED https://archives.gentoo.org/gentoo-dev/message/ecae36137bfa350876473c48ec608318

[7] https://bugs.gentoo.org/page.cgi?id=fields.html#bug_status


[1] Part of *app-portage/gentoolkit-dev*