My Package Sponsorship Policy

Stuart Herbert
Gentoo Developer
stuart@gentoo.org

Draft 2
25 July 2003

Warranty Notice

The information in this document represents represents the opinion of the author at the time of writing. No fitness for purpose or warranty is stated or implied. The author accepts no responsibility for the use of any of the material in this document.

Nothing in this document constitutes a legally-binding contract between any parties, or a partnership agreement of any kind.

This document is correct and valid for the date shown on the front cover. If your copy of this document is more than six months old, it may be out of date. The latest edition of this document can be downloaded from:

http://dev.gentoo.org/~stuart/sponsorship/

History

Published editions

1. Draft 1, 2003/07/23 – first version, for review
2. Draft 2, 2003/07/25 – incorporating Developer Relations feedback

Production Notes

Produced using OpenOffice running on Gentoo Linux. For more information, and to download your free copy of Gentoo Linux, please visit the Gentoo Linux website at http://www.gentoo.org/.

Typeset in Helvetica and Palatino.

This document is formatted for duplex printing.

Acknowledgements

I'd like to thank the following people for their feedback and support on this document:

- Todd Bergman <tberman@gentoo.org>
- Donnie Berkholz <spyderous@gentoo.org>
- Seemant Kulleen <seemant@gentoo.org>
- Jon Portnoy <avenj@gentoo.org>

Contents

Introduction

About This Document

This document is my personal handbook for adding and maintaining packages in Gentoo. It captures *my* preferred way of working as a Gentoo Developer.

It also presents the things that *you* have to do if you want me to add your packages to Gentoo Linux, and if you want me to maintain those packages. Please don't be concerned about the size of this document. Everything that I've captured here – every requirement that I've added – really does make *everyone's* life much easier.

If you have any concerns, comments, requests for clarification, or any other feedback about this document, please please *please* email me at stuart@gentoo.org, or come and have a chat with me on the Gentoo IRC channels. My IRC nickname is Stuart.

Background

I am a Gentoo Developer.

Gentoo is a Linux meta-distribution, available for Intel-based PCs and other hardware platforms. The key thing about Gentoo is that packages are downloaded as source code and compiled specifically for your machine[1]. Rather than use tried-and-troublesome solutions like RPM, Gentoo provides automated installation scripts called 'ebuilds', and a package management system called Portage. Ebuilds make it easy for Gentoo users to install, upgrade, and remove packages on their Gentoo machines.

Many ebuilds are created by ordinary users of Gentoo Linux, and submitted to our Bugzilla system where Gentoo Developers pick them up and add them into Portage.

As a Developer for Gentoo, my role is to create new ebuild scripts and add them into Portage. I also respond to bug reports filed in our Bugzilla system. Although I have a specific type of package that I'm responsible for, I can (and do) also add ebuilds for all sorts of packages to Portage.

Like all of my fellow Developers, my work is done on a voluntary basis. There is a limit to the amount of time that I can make available to work on Gentoo. As a consequence, this limits the number of packages that I can maintain inside Portage.

1 Don't worry – Gentoo also supports commercial packages that have been released as binary-only.

The best way for the Gentoo project to scale is for my role as a Gentoo Developer to include “sponsoring” the work of others. It takes me less time to audit ebuilds created by the original author(s) and/or maintainer(s) of packages, and to act as a conduit for getting those ebuilds into Portage. To help with “sponsoring” packages, I've written this short document to clearly state what you need to do if you want me to “sponsor” your package.

Rationale

I've drawn up this document because of three different motivations.

Firstly, when I first became a Gentoo Developer, I wanted to work out a simple but reproducible process for working with packages. Writing down this process is an essential part of making it reproducible ;-)

Secondly, and perhaps arrogantly ;-) I don't want to end up feeling like working on Gentoo Linux is an obligation and a pain; it should be exciting and fun. Getting the basics right when a package is first added should (fingers crossed) sort this out.

Finally, I want to ensure a minimum standard of quality for Gentoo Linux users in my work. To me, Gentoo is about getting the latest releases of packages to our users as quickly as we can. That means that my job (as I see it) is to provide high-quality installation scripts (known as ebuilds) first, and then to worry about whether the latest release of the package itself is good or not.

Disclaimer

All of the Policies that I've listed in this document are **not** official Gentoo Policies at all. They are all Policies that I have voluntarily adopted to cover my personal contributions to the Gentoo Linux project.

If you have an ebuild or package that you wish to see added to Gentoo Linux, you only have to comply with my Policies if you want me to sponsor your package.

If I decide not to add your project to Gentoo Linux, you're perfectly free to try and find another Gentoo Developer who would be willing to sponsor your package.

Explaining Sponsorship

What Is Sponsorship?

When I use the term s*ponsorship*, I am referring to a way in which you can get new software added to Gentoo Linux without having to become a full-fledged Gentoo Developer yourself.

Sponsorship's very simple. Basically, you provide one of:

1. A working ebuild script[2], or
2. The resources I need to write an ebuild script myself[3]

and together you and I work to satisfy the Policies in this document.

How Do You Get A Package Sponsored?

If you want me to sponsor your package, please log a new bug on our Bugzilla system – http://bugs.gentoo.org. Make sure that you select *Gentoo Linux* as the *product*, *Ebuilds* as the *Component* , and in the comment remember to mention that you want stuart@gentoo.org to sponsor your package!

2 *Ebuild scripts* are the installation scripts that allow other users to install software onto their Gentoo Linux machines. You can find out more about writing your own ebuild scripts at http://www.gentoo.org/doc/en/gentoo-howto.xml

3 This option is really only for commercial software

Acceptance Criteria Policy

Introduction

This is my Policy on the conditions that must be met before I'll sponsor a new package in Gentoo Linux.

Relationships

The work involved in sponsoring a package in Gentoo Linux doesn't end once a working ebuild has been added to Portage. A lot of effort goes into supporting Gentoo's users, through bugs logged in our Bugzilla system, and discussions held on our mailing lists and in our forums.

None of this is possible if I can't forge a good working relationship with the author / maintainer of the package. I'll need to be able to ask questions and seek advice about the package, and I'll need to be able to get bug fixes fast-tracked through as quickly as possible.

It goes without saying that packages without an active author / maintainer aren't packages that I'd like to add to Gentoo Linux ;-) Or sponsor, for that matter.

The other key relationship is with the community of users active on the package's mailing lists. Expert users can often provide insightful help and advice. A user community doesn't have to exist, but if the community is largely hostile to newcomers, I'm less likely to want to add the package to Gentoo Linux.

Administrative

New Releases

There must be a way for me to get automatic notification of new releases of the package via email. The preferred way is a low-volume announce list.

Security Notices

There must be a way for me to get automatic notification of any security holes identified in the package – and (of course) of any security-related fixes ;-) The preferred way is a low-volume announce or security list.

Bug Reporting

There must be a way for me to report bugs in the package. A support mailing list, a bugzilla system, a SourceForge tracker system, and similar are all acceptable.

I'm not keen on just sending emails to the package's author, but for now I'll give it a go and see how it works out.

Released Files

Released packages must be available for download from the Internet.

For packages released under an OpenSource license, the package must be available for download in a way that's compatible with Portage's automatic download feature. Basically, I need a URL that can everyone can use with something like the wget command.

For packages released under a closed-source license (typically commercial packages), I'd prefer it if the software was available for download via the wget command. Where that isn't possible or practical, I need to be able to provide instructions to the user to tell them which binary package to download from the company's site.

I am very happy to add ebuilds for commercial software to Gentoo Linux. However, the ebuilds will be written to download the commercial software from the company's own web servers only. Gentoo, and our mirrors, do not have enough diskspace to offer to mirror your commercial software. Of course, if you're interested in offering to provide a Gentoo mirror ... ;-)

Resources

This part of the Policy is mainly aimed at commercially licensed, binary-only packages.

Commercial Licences

If the package requires a commercial license, you *must* provide me with whatever licenses I need to test installation and use of the package. For networked packages (daemons), a single user license will often not be sufficient. In return, I'm willing to sign an agreement stating that I will only use donated licenses for sponsoring your package in Gentoo Linux.

Special Hardware

If the package requires special hardware that I don't have, then you *must* provide me with access to that hardware. I don't necessarily need to have the hardware sat under my own desk – root access via SSH or NX over the Internet to a suitable

machine sat in your office would be more than sufficient. Of course, if you'd like to ship hardware my way ... ;-)

Licenses Covering Supplied Resources

Any licenses or agreements covering any resources you provide *must* be agreed in advance – before I add ebuilds to Portage. If you change these licenses or agreements in future and I can't agree to your changes, I'll stop "sponsoring" your packages within Gentoo. I won't remove your packages, but neither will I provide ebuilds for new versions. Any packages dropped in this way will still be subject to my End-of-Life Policy; I've added this Policy to the end of this document.

Marking An Ebuild As Stable

Pre-Conditions

I'll only mark package as stable if all of the following conditions are met:

- The *relationships* listed earlier in this *Acceptance Criteria Policy* are in place.
- The *acceptance test scripts* required for the *Testing Policy* are in place.
- There are no *critical* bugs open in Gentoo's Bugzilla.
- There is an existing unstable ebuild for the same version of the package[4].

Process

I'll post an announcement to the gentoo-dev@gentoo.org mailing list. If there are no reasonable objects within seven calendar days, I'll mark the ebuild as stable.

Security & Bug Fixes – An Exception

If an existing stable ebuild needs updating to provide urgent security or bug fixes, then I will normally mark the new ebuild as stable straight away. An announcement will be sent to the gentoo-dev@gentoo.org mailing list when this is necessary.

Updating An Existing Package

When you want to submit an update to an existing package, make sure that:

- you base your update on the existing ebuild that's in Portage, and
- you submit your update as a diff file to our Bugzilla system (you can create a diff file using the 'diff -u' command).

4 There's been a fair bit of debate about this point, so I'll try and make sure I put this across as clearly as I can. All new packages will be marked as *unstable* at first. This is to allow time to find any real howlers either with the ebuild script, or with the package itself. When I mark the package as stable, that must be the **only** change that I make to the package when I do so. If I need to make any other changes, then the ebuild *in my opinion* is not ready to be marked as stable. When new versions of a package are released, any new ebuilds will be initially marked as unstable too – unless they are *critical* security or bug fix releases. Again, this is to allow a little bit of time to get the ebuild more widely tested. This is a compromise between getting packages out quickly, and not shafting users who currently have working machines.

Testing Policy

Introduction

Testing is not popular work, but it *is* an essential part of doing a job well.

The testing that I've written about in this *Testing Policy* is mainly aimed at the package itself, rather than the ebuild. What I'm concerned about here is the experience that the Gentoo User gets, and you should keep this in mind when writing tests required by this policy.

Acceptance Tests

Definition

An *acceptance test* is a test that helps prove whether a package has been installed correctly by the ebuild. Most packages will require quite a few acceptance tests to provide enough proof ;-)

I am *not* interested in proving that the package is robust and practical. The whole aim here is to ensure that the package has been installed correctly. If the package itself is of alpha or beta quality, and not suitable for production or regular use, that's not what the acceptance tests are concerned about.

Description

An *acceptance test* is a set of instructions to follow. If the instructions fail for whatever reason, this should mean that the test has failed.

It is important that the instructions are clear, concise, and repeatable (deterministic!). Anyone should be able to follow the instructions without knowing anything at all about the package. Don't worry, they don't have to be long or formal. Just write down some simple instructions which any competent Linux user could follow, and that'll do fine.

Acceptance tests do not have to be automated. That said, if you can make them automated, I won't object ;-)

Requirements

I can't write the acceptance tests for packages that I don't personally use; I simply don't know these packages well enough to write a sensible set of tests.

If you want me to sponsor your package or ebuild, you must provide a set of acceptance tests for me to use. And when the acceptance tests need to change to accommodate a new release, you must provide me with the updated tests.

Usage

I will run through *all* of the acceptance tests every time I intend to add a new ebuild to Portage. *All* of the acceptance tests have to pass if the ebuild is to be marked as stable.

All acceptance tests will be installed in the directory:

/ usr/share/gentoo-qa/<package>/acceptance/

if you have 'qadocs' in your USE flag settings.

Incident Tests

Definition

An *incident test* is a test that reproduces a known bug in a package or ebuild. Running an *incident test* script proves whether a bug exists in an updated package or ebuild or not. You'd be amazed at how many known bugs either never get fixed, or get re-introduced in a later version!

Description

See the Description of Acceptance Tests ;-)

Requirements

I will write the incident test scripts for every genuine bug reported against the package in our Bugzilla system.

Any incident test scripts that you can provide as well will be gratefully received ;-)

Usage

I will run through *all* of the incident tests every time I intend to add a new ebuild to Portage. I'll use the test results to maintain a set of release notes listing bugs fixed and still present, which I'll post to the gentoo-dev@gentoo.org mailing list. The release notes will be stored as *files/REL_NOTES* alongside the ebuild.

All incident tests will be installed in the directory:

/usr/share/gentoo-qa/<package>/incidents/

if you have 'qadocs' in your USE flag settings. Feel free to try them out yourself, to make sure I haven't forgotten to test them all ;-)

All incident test scripts will be forwarded on to the author / maintainer of the package, and hopefully a new release containing lots of bug fixes will come in the opposite direction soon after ;-)

Support Policy

Introduction

Gentoo Linux is provided as-is, with no liability or warranty implied or stated.

That said, the Gentoo team is rightly proud of the way our volunteers quickly respond to problems logged into our Bugzilla system. My Support Policy is aimed at maintaining the excellent reputation of Gentoo Linux.

The Policy

Responding To Bugs

I will do my best to respond to all bugs filed against packages I maintain or sponsor within 48 hours – even if it is just to acknowledge that the bug has been raised, and to re-assure the user that someone *is* listening to them.

When I'm away (on vacation, business, etc. etc.), I'll ask our bug-wranglers to add a note to new bugs stating when I'll be back to look at them.

Installation Bugs

I will do my best to diagnose and fix all installation-related bugs – the bugs caused by problems in the ebuild script.

Package Bugs

I will investigate all package-related bugs logged into Bugzilla – the bugs that exist in the package. Reproducible bugs will be documented with an incident test script, and will be forwarded on to the package's author for a fix.

I will do my best to get bug fixes from the package's author back into Gentoo Linux as quickly as possible.

I *may* attempt to fix the bug myself, but most of the time I won't know the package well enough to be able to do so.

Verifying A Fix

Whatever the type of bug, whenever a fix is available, I will contact the user(s) who logged the bug and ask him/them to confirm that the bug is fixed. I will not close a bug as FIXED until the user has confirmed that the bug is fixed.

If the user does not reply within a fortnight, I will close the bug.

End-Of-Life Policy

Rationale

Not everyone wants to run the latest and greatest release of a package. Not everyone can upgrade every day or week. And, for organisations deploying Gentoo across a number of machines, it can be helpful to be able to standardise on every version of a package on every machine.

The best way to address this is to publish an end-of-life policy, setting out in clear terms how long an ebuild will exist before it is removed.

Stable Ebuilds

A 'stable ebuild' is an ebuild that has been marked as 'stable'.

I'll keep stable ebuilds in Portage for a minimum of three months. During this time, I'll provide support for the version following the Support Policy listed earlier in this document.

(There is one exception to this. I'll look to remove ebuilds for older versions of packages that contain serious security problems, and I'll remove ebuilds for packages where the code is no longer available to download).

When the time comes to remove a stable ebuild, I'll post a message to the gentoo-dev@gentoo.org mailing list to announce the upcoming removal. If no-one has a compelling reason against, I'll remove the stable ebuild a fortnight after the announcement.

Unstable Ebuilds

An 'unstable ebuild' is an ebuild that is totally masked, or which has one or more ~ARCH keywords protecting it.

I'll aim to keep the last three unstable ebuilds for a package in Portage. However, at my discretion, if I believe that any unstable ebuild is no longer worth keeping, I'll remove it from Portage. I expect the main reason to do this is where there's a stable ebuild that's newer and better than the unstable ebuilds.

When the time comes to remove an unstable ebuild, I'll post a message to the gentoo-dev@gentoo.org mailing list to announce the upcoming removal. If no-one has a compelling reason against it, I'll remove the unstable ebuild a fortnight after the announcement.