

# Requirements for application

As required by Gentoo SoC application I provide links to [my post](#) on gentoo-soc mailing list and commits in [git](#) and [SVN](#) repositories for two projects I am involved in to some degree.

## Project mission

Main goal of this project is to create easy way for Gentoo developers to maintain database of files present in packages residing in Portage tree. This will open new possibilities for both Gentoo team and users alike. Gentoo QA team will be able to improve existing ebuilds and common users will be able to check package contents before installing packages.

## Abstract

Gentoo emerge is checking for file collisions before installing packages. However this collision checking only works after package has been compiled. This is natural since Gentoo is source based and package contents are different for every architecture, set of USE flags and compiler options. Proposed project would provide users and more importantly Gentoo QA team access to database of files in packages from Portage tree. Information about package contents would enable Gentoo QA team to improve existing ebuilds by solving problems resulting from collisions. Users would be also able to see approximate size of package and query package contents before installing it. When invoking non-existing command on shell, shell handler could check file database and offer packages to install. There are definitely other uses waiting to be discovered.

## Deliverables

### Tool for aggregating package contents

Also information from stat() syscall for files. Most probably this part will be database and wrapper set of tools. More sources of package contents would be possible:

1. tinderbox set-up exactly to collect this information (semi-automatic, fixing compile errors manually)
2. binary packages (perhaps interaction with GSoC project [Improved binary package support?](#))
3. restricted user supplied package contents (this would need to be analyzed more deeply from security/privacy point of view)

Public API for read-only access and/or committing package contents could be provided depending on analysis and gentoo team preferences.

### Web interface for package contents database

After database of packages is successfully created this web interface would provide world-readable access to package contents through web browser with ability to search for files and packages.

### Console tool for querying package contents

Depending on the size of database, it could be downloaded from server or client could query public server for package contents and/or other information provided. I would like to keep dependencies of

this tool to the minimum (Python, depending on the database format maybe sqlite).

All deliverables will include documentation.

## Timeline

15. April - 23. May: Getting to know Gentoo community spirit a bit more and some lightweight design discussions for first deliverable

24. May - 7. June (2 weeks): Discussion to design first deliverable

8. June - 21. June (2 weeks): Coding First deliverable with basic documentation and design for 2nd and 3rd deliverable

22. June - 13. July (3 weeks): Second deliverable (web interface)

14. July - 4. August (3 weeks): Third deliverable coding (client utility)

4. August - 17. August (2 weeks): Finalizing, fixing bugs, cleaning up code and documentation

Obviously this is only very rough idea. Lot of things will depend on analysis and design decisions for first deliverable.

## Biography

I am grad student at Slovak University of Technology in Bratislava studying Software Engineering. I have participated on few OSS projects (musicpd, gstfs, gstreamer) before, mostly with bugreports and (admittedly very few) patches. My bachelor's project was "Userspace process access restriction in Linux and FreeBSD". I am quite confident with C/C++, Java, basic shell scripting and Python. I have also used quite a few other technologies, but usually only for one or two projects. Of course I would like to learn more and try something new. As far as my Linux usage goes, I have been happy Gentoo user for almost 4 years now. Before that I used Linux From Scratch based distribution for one year so I have a lot of experience with build errors, which should come in handy :-).

My work experience includes participation on coding part of Ripple control system for energetics companies (together with more smaller projects for MicroStep-HDO, Slovakia). I've also spent 6 months as intern in Edisoft (Portugal) where I worked on model-driven development tool for FPGAs.

For more information do not hesitate to contact me personally.