

Java on Gentoo



Gentoo Java team

- 8 developers
 - Some are not very active
- Current project lead is Joshua Nichols
- Power users help by committing stuff to the java overlays
 - For example gcj as jdk is currently maintained like this
- Some power users eventually become devs



Statistics

- 473 packages currently use java eclasses
 - Plenty more waiting in the overlays
- Main jdks currently are Sun and IBM jdks.
- The FOSS Java stack is there if the users want it
 - Our users have been more interested in 1.5 than a FOSS implementation
 - Needs work to integrate all the vms to the generation 2 setup



Where are we now?

- Generation 2 is stable on every arch
- Ebuilds using generation 2 are slowly being stabled
- Still 117 slots to migrate to generation 2
 - Some of these are stale/dead packages that we are considering to move to a graveyard overlay
- dev-java/ant-tasks is splitted to individual packages



Some benefits of the current system

- You can choose the bytecode version of your installed class files
 - There are no instructions for this yet because it enables you to shoot yourself to the foot unless you know what you are doing
- It's very easy to switch between used compilers and jdks
 - Very easy to test something with a bunch of jdks
- Separation of a user and system vm



How is this done?

- We automatically rewrite all the build.xml files to generate the bytecode indicated by the dependencies
 - Use the lowest version pulled by dependencies
- Use `-Dbuild.sysclasspath=ignore` to ant
- Don't make any optional ant tasks available to ant unless requested by the ebuild
- Some automated QA checks integrated to the eclasses



Generation 2 jdk concepts

- Three different jdks
 - Build vm (used to emerge stuff)
 - System vm (used by root and default for users)
 - User vm (used by users if set)
- I can run Netbeans with sun-jdk-1.6 to develop things but still compile every package with ibm-jdk-bin-1.5 to catch com.sun.* usage



Good things gotten by integrating with the package manager

- Dependency handling by our automatic wrapper script creator. Uses the dependencies from ebuilds to put everything needed to the CLASSPATH.
- You can tweak options passed to java on a launcher by launcher basis using bash.



Things to do in 2007

- Improve the FOSS java stack
 - The core devs usually work on improving one big thing at a time
- Integrate maven
 - Probably use the work done by JPackage as base
- Package more J2EE stuff
 - Things like JBOSS



A tour of java ebuids

- Next let's look at some java ebuids and how it's all done, but before that does anyone have any questions?

