The Portage Package Management System

Andrew Michaud

11/29/13
Gentoo Linux was initially developed by Daniel Robbins. Working on his own, he needed automation scripts to develop a Linux distribution from scratch. Created the ebuild system and scripts to automate unpacking sources, compiling, installing, and packaging.

After a hiatus using FreeBSD instead of Linux, decided to remake Gentoo’s autobuild system integrating FreeBSD features.

Began calling system “Portage”

Focused on stability, and allowed users to see how a Linux distribution comes to be.
First, update the local portage tree.

- The Portage tree contains ebuilds, which contain everything Portage needs to install and maintain software.
- Portage looks at the ebuilds on your system, so if you want updated software, you need updated ebuilds.
- **To update:** `emerge --sync`
  - Note: Only run this once per day, it stresses the Gentoo servers.
  - `emerge` is the command-line way to access Portage. Non-command-line interfaces exist but I’ve never found any that worked nicely.
Next, search for software.

- By default, you can use `emerge --search <keyword>` to search within titles.
- You can also use `emerge --searchdesc <keyword>` to search within descriptions.
- There are better ways. I particularly liked eix for indexing and searching my Portage tree when I ran Gentoo.
- But that isn’t part of standard Portage.
Installing software.

- Installing software is pretty simple. Just use `emerge <package name>`.

- Caveats:
  - Portage installs everything from source. If you’re installing Firefox or LibreOffice, it’s gonna take a few hours.
  - Portage will install whatever dependencies a package claims it needs. Maybe Vim wants X11 support. Maybe VLC wants PulseAudio support. If a package wants it, Portage will install it for you.
  - How to stop this? USE flags, in a couple sections.

- In general, use `emerge -vp <package name>` so Portage will tell you what it’s going to do before it does it.
Uninstalling software.

- Uninstalling software is also pretty simple. `emerge --unmerge <package name>`.
- Unsurprisingly, there are also caveats here
  - Portage won’t check if other packages need whatever you’re uninstalling. Check with `emerge -pv --depclean <package name>` to see if there are any reverse dependencies on what you’re uninstalling.
  - Portage will warn you before you uninstall something system-critical, but you can still do it. Don’t try uninstalling gcc or python if you enjoy having a working system.
USE flags.

- USE flags let you specify what should and should not be on your system.

- Global USE flags reside in `/etc/portage/make.conf`, in the USE variable.

- Use “+X11” to force everything to enable visual support (if it can). Use “-alsa” to disable support for the ALSA sound system. There are MANY, MANY more use flags you can set.

- You can also set per-package USE flags in `/etc/portage/package.use`.

- Occasionally Portage will tell you certain USE flag changes need to be made. Add these flags if you want your packages to install.
System updates.

- Portage keeps track of everything you’ve manually installed in `/var/lib/portage/world`.

- To update everything in that file, use `emerge --update --ask world`.

- This won’t get dependencies for those programs. To update dependencies (programs required by other programs) add `--deep` before `world`.

- To update packages that are only required during the compile and build process (which you shouldn’t often need to do) add `--with-bdeps=y`.

- Finally, use `--newuse` if you’ve changed use flags recently and want Portage to check if anything needs to be recompiled or newly installed.
But wait, there’s more.

- When you fully update your system, some packages are no longer needed. To delete them, run `emerge --depclean`.
  - Be careful with this one (try running it with `--pretend` first). I’ve seen it delete kernel sources I hadn’t finished updating before.

- To rebuild any applications that broke, run `revdep-rebuild`, which comes from the `gentoolkit` package.
If Portage says things are blocked, or messes up dependencies:

- Give up.
- Shout at your computer.
- Cry.
- Reinstall all the packages and dependencies Portage is complaining about until it stops complaining.
- Uninstall whatever Portage is unhappy about and give up.
- I’ve only had luck with the last two approaches.
Why Should I Use Portage/Gentoo?

Portage is basically what makes Gentoo Gentoo. Why should you bother with Gentoo at all?

- There are few better ways to learn Linux. If you have to do everything yourself, you learn quickly if you want a working computer.
  - Linux From Scratch is probably better, but a lot more work.
- You can have whatever set of packages you want. Install whatever browsers you want. Use KDE or GNOME. Use both!
- You can ban whatever packages you want. Want to keep EMACS/Vim off your system? Use “-emacs/-vim” globally.
- Remember those USE flags from before? Tinker with them to your heart’s content.
Portage is basically what makes Gentoo Gentoo. Why should you bother with Gentoo at all?

- You can compile specifically for your system. Just tell Portage your system architecture.
  - Packages compiled for your system specifically could run faster, but you likely won’t notice the difference.
- Want to run on PowerPC, Sparc, mips, or ARM? Go ahead! You could put it on a Raspberry Pi if you wanted, but enjoy your painfully long compile times.
- You can be sure of exactly what you’re putting in your system. Just read the source code before you compile it.
Why Should I Never Use Portage/Gentoo?

Gentoo requires a great deal of work to get running and to maintain.

- (Some) Ways I broke my computer when I ran Gentoo:
  - Messing up my xorg.conf and losing graphics capability.
  - Messing up grub.conf and having an unbootable system.
  - Misconfiguring a kernel and having an unbootable system.
  - Breaking the C Compiler and being unable to install anything.
  - Uninstalling Python and being unable to install anything.

- Time it took until my Gentoo installation worked “properly”: several weeks. Maybe a couple of months.

- Compiling from source is sloooooooooow.

- The speed benefits of compiling from source are barely noticeable.

- People constantly ask you why you run Gentoo.
Summary

- Portage is a great package management system, but very complicated to use properly.
- There are lots of good reasons to use Gentoo (and Portage).
- There are equally many, or more, reasons to never use Gentoo.

Conclusion

- If you want to learn Gentoo, spin up a machine in Virtualbox and give it a try.
- It just takes patience and a willingness to learn.
- The Gentoo handbook will tell you everything you need to know (www.gentoo.org/doc/en/handbook), and general Googling will tell you anything it misses.