

# Release time!

The code has been written, the features have been added, and the release deadline creeps ever closer...

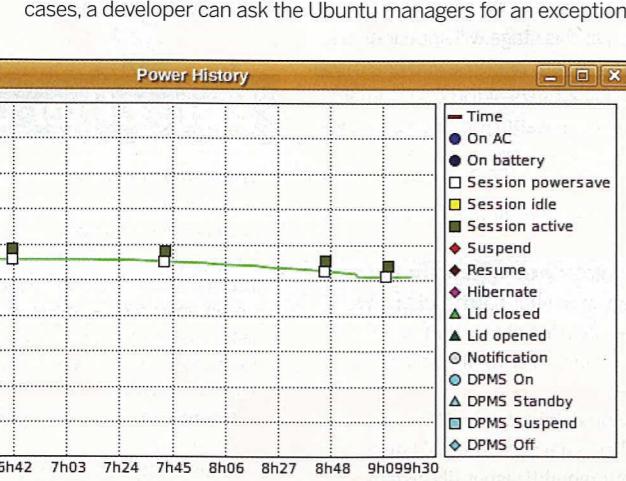
**T**owards the end of the development cycle, things get a little hectic as the number of bug reports starts to swell. Brian Murray heads up the Quality Assurance team for Ubuntu, and monitors the level of feedback from the community. "With a completely open bug reporting system, like Ubuntu has, it seems that we receive the greatest quantity of bug reports after the final version of the release has been made and it becomes more widely used. However, the Desktop CD with its Live environment provides a great way for people to test the development release and contribute to the process."

So, it's a catch-22 situation: the Ubuntu team want bug reports to make the final release super-polished, but they'll get the most bug reports when the distro has actually been released! That said, if a major stability problems occur, the release date for snapshots and the final version can be altered: "We would rather provide our users with a great experience than meet a target release date."

This was the case with Ubuntu 6.04 – in order to fix some niggling stability issues, the team pushed the release back two months for some extra work. Consequently, it was numbered 6.06 on launch.

Around six weeks before the release is due, the "Upstream Version Freeze" kicks into effect: no new package versions, either from individual software developers or existing Debian packages, can be added to the distro. So if you're the *AbiWord* packager with 2.4.6 built for Gutsy, and then 2.5.0 comes out, you have to leave it for the next Ubuntu release. Developers can still issue new packages with bug and security fixes, but the goal at this point is to have a known platform that the team can begin to stabilise.

About a month after this, "Feature Freeze" comes into play – from this point onwards, no new features which alter the workings of the distro may be added. As with the version number freeze, this is essential in keeping the distro together: you don't want some developer completely rewriting the boot scripts a couple of weeks before the release. If a planned feature isn't completed by now, it has to be delayed for the following distro release. In rare cases, a developer can ask the Ubuntu managers for an exception.



## My favourite Gutsy feature: Corey Burger



"I'm looking forward to all the server work that is being done. There's a lot to do to make Ubuntu Server as polished and easy to use as the desktop. Work is being done on LDAP libraries and packaging of *Ebox*, a web-based network services control tool."

Creative Commons image, elkubuntu, <http://tinyurl.com/2s9sch>

## End in sight

Two more freezes follow: artwork and "strings". The latter stops developers from changing text in their programs, and coupled with the former it means that documenters can prepare screenshots and help text without the distro suddenly changing its appearance. One week after the string freeze, and a month before the final version is due, a beta release is produced which contains working – albeit potentially unstable – implementations of the distro's new goodies. Ubuntuers term it "feature complete", meaning that it has everything in place for the final version, but needs testers.

At this point the bugfixing team gets into overdrive, as Murray explains: "As the final release nears more people take it for a test drive and therefore the quantity of bug reports can increase. Subsequently the challenge becomes ensuring that those new bug reports are triaged quickly and then that the high priority ones are identified and resolved."

Aside from the technical issues, parts of the community start working on the distro's image. Press releases need to be prepared, release notes drawn together and screenshot-laden new feature guides need making. Corey Burger is in charge of marketing at Ubuntu: "The marketing team and Canonical usually chat just before release to see what each is doing. That usually happens via myself calling one of the Canonical marketing people and figuring out what they are doing. For the Feisty release, this meant Canonical handled the press release and any incoming press inquiries (at least in the English press) and the community created a lot of the online collateral, such as the 7.04 Features tour."

One week before the final release date, the team uploads an RC (release candidate) build of the distro: potentially the final version unless any last-minute "show-stopper" bugs are found. Development is halted; only fixes for extremely notable bugs will be allowed in at this stage. Most of the community waits anxiously, hoping that no disk-frying nightmares surface on some obscure combination of hardware. If all's well, the team produces a final build of the packages and disc ISO images, sends them to mirror servers around the world, and announces the new release on the Ubuntu website.

Meanwhile, some developers start piecing together plans for the next Ubuntu release. The cycle starts all over again...

# Getting involved

Now that you've got the lowdown on Ubuntu's development process, why not chip in some help with the distro – or indeed any other distro?

**A**s Ubuntu continues to flourish, the project is constantly looking out for new talent – and it's easy to get involved. You don't have to be a long-time coder to help out with the distro, as there are many ways to dive in and add your improvements. A new distro release is the product of many different talents, all working together with a single goal: to make a usable operating system. So even if one aspect of the development process seems alien or intimidating, never fear – you can work on your own projects without having to dip your toes into everything else.

As we've seen, some high-level decisions regarding Ubuntu's direction are made by Canonical, but the vast majority of development activity is open for all to see. Helping out with a distro is rewarding and gives you great experience for the future, so here are some areas worth looking into (and they're equally applicable to other distros such as Fedora or OpenSUSE):

» **Coding** Creating packages, fixing bugs and issuing security patches is the bulk effort of any distro; it doesn't require extensive programming knowledge though. Basic knowledge of C/C++ and Bash scripting is useful. If you're familiar with the command-line, editing text files and building software from source, you're equipped to produce Ubuntu .deb packages and see if you can work them into the Universe repository. If you're a dab-hand with programming, you can try one of Ubuntu's paid bounty projects.

» **Testing** Download mid-development snapshots and new packages, and report bugs if they don't work at <http://bugs.launchpad.net>. Keep your bug reports snappy but informative; saying "Doesn't work, please fix" will only annoy developers, whereas listing exact error messages and your system spec will help them to narrow down the problem.

» **Documentation** One problem with Linux – regardless of distro – is the piecemeal documentation. Coders love to write code, but don't relish the prospect of writing user guides. You can help out by writing a manual for an as-yet undocumented program, or improving the user guides for well-known apps that only have sketchy docs. Useful knowledge here is XML and DocBook.

» **Artwork** Like user guides, this is something that hackers sometimes forget, leading to ugly or weakly designed interfaces.



» **Coders, testers, artists, documentation writers – many skills are essential in making a well-received Ubuntu release.**

## My favourite Gutsy feature: Matt Zimmerman



"I'm personally looking forward to the launch of our mobile product, Ubuntu Mobile and Embedded, which will come as part of the 7.10 release. More about that at <http://wiki.ubuntu.com/MobileAndEmbedded>."

Create icons, images, desktop wallpapers and interface mock-ups to make your distro look smoother and easier to understand. If you can make these graphics using open source software such as Gimp and Inkscape, even better!

» **Support** Ubuntu's community is up there with Gentoo's as one of the most welcoming and friendly; try helping out at [www.ubuntuforums.org](http://ubuntuforums.org) by suggesting answers to common user problems and making sure that nobody feels alienated when switching to Linux. Your fuse may be shortened by annoying "ZOMG teh linux doesn't work Y not"-type posts, but stay friendly and approachable, and you'll earn much kudos in the community.

» **Localisation** As highlighted by Ladislav in this month's DistroWatch on page 38, in many parts of the world, due to the requirement of knowledge of a second language, the benefits of computer use are denied to much of the populace. If you're bi- or multi-lingual, particularly with knowledge of a language that isn't supported by proprietary OSes, your help can be invaluable in every area, especially support, translating existing documentation and advising GUI developers on interface updates.

If you want to dive headlong into Ubuntu development, hoping to work your way into the MOTU and Core teams as described earlier, you should get involved in the mailing lists (<https://lists.ubuntu.com>). Suggesting ideas is good, but as Linus Torvalds once said, "Talk is cheap. Show me the code." If you can package up a piece of software that's not yet in the Ubuntu repositories, and produce stable, clean .deb files, you'll be able to develop contacts and get yourself involved in the mailing lists. Similarly, if you have an idea for the base Ubuntu distro and can implement it in a clear, safe and easily-testable way, code it up and get posting on the lists..

## Write, right?

If programming isn't your thing, but you'd like to try your hand at improving the distro's documentation, visit <http://doc.ubuntu.com>. Or if you've got more of an artistic and graphic design leaning, try contributing icons, logos and desktop backgrounds at <https://wiki.ubuntu.com/Artwork>. Whatever you choose to do, the Ubuntu joining-in page at [www.ubuntu.com/community/participate](http://www.ubuntu.com/community/participate) provides links to all manner of resources – mailing lists, wiki pages and Launchpad sections. If you're friendly and describe your work or suggestions in detail, you'll get useful feedback and more stepping-stones to further development.

Or why not lend a hand with your own favourite distribution? See the box at the bottom of page 45 for more. Good luck! 

Creative Commons image, silaw, <http://tinyurl.com/3489hn>