HotPicks

ON the DISC

The best new open source software on the planet!

Mike Saunders
A coder since Amiga times, Mike's a Linux and BSD guru.

This is the place where we get to profile some of the hottest software around.

Each month we trawl through the hundreds of open source projects that are released or updated, and select the newest, most inventive and best for your perusal. Most of the HotPicks are available for you to try out on our coverdiscs, but we've provided web links if you want to make sure you have the very latest version.

If you have any ideas for software that we should cover, email me at **mike.saunders @futurenet.co.uk**.

HOTPICKS AT A GLANCE

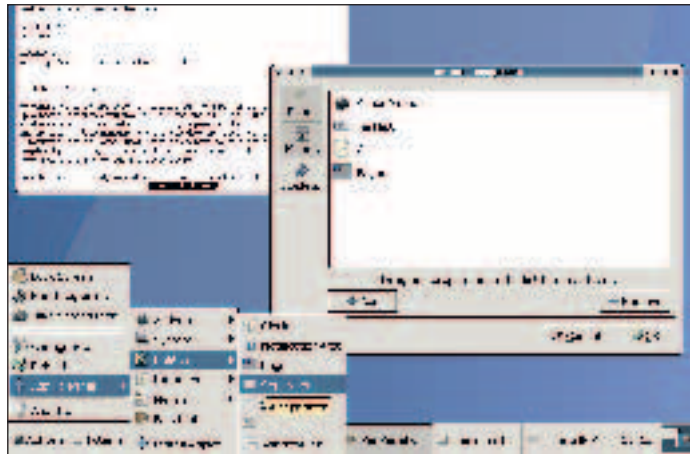## LOOK OUT FOR THE HOTPICKS AWARD

Everything covered in our HotPicks section is unmissable, but each month we single out one project for outstanding brilliance. Only the very best is chosen!

PANEL PROGRAM FOR THE DESKTOP

PerlPanel

■ **VERSION** 0.9.1 ■ **WEB** http://jodrell.net/projects/perlpanel

PerlPanel in *Xfce*, sitting along the bottom with the config tool in front.

Newcomers to Linux are often bamboozled by the array of components that make up a modern desktop environment. In Gnome and KDE, for instance, you have a window manager for manipulating windows, a file manager that creates desktop icons and navigates the filesystem, and a panel containing menus, switchers and applets. Most users aren't aware of the underlying complexity – as long as the result is a cohesive whole, there is no need to poke around and fathom out its construction.

Nonetheless, the ready-made desktops, while attractive on the outside, tend to be bloated in places and put strain on system resources. Thankfully, Linux's flexibility means it's possible to assemble piecemeal desktops – a window manager from here, a file manager from there, and so on. Reduced integration is usually the price to pay, but on the upside, the results are typically much lighter. *PerlPanel* is a desktop panel intended to go alongside a variety of WMs, providing a comfortable interface without massacring RAM.

It was built with *GTK 2.x* and written in Perl, and a plethora of modules are required to get it running – these range from commonly available to rather obscure. For most of the Perl modules, a search on **www.cpan.org** will bring up good results. It's unlikely that your distro will have all the modules so a bit of handiwork is needed. These installation hurdles tend to put off many first timers, so we'd like to see some tidying up in future releases to make things simpler.

Added extras

PerlPanel will work with most modern window managers, although the level of functionality and features depends on the WM's adherence to standards. When started up, it creates a plain desktop panel along the bottom of the screen, containing a menu button, a taskbar of running apps, a clock and a workspace switcher. No major surprises there – but that's a Good Thing. Users of Gnome, *Xfce* and other *GTK 2*-based desktop tools will find *PerlPanel's* default setup and

appearance very satisfactory. Naturally, it picks up the current theme, and so fits in charmingly with other *GTK* apps.

Clicking the clock brings up a spiffy little Gnome-esque calendar with a reminder feature, and context menus have been scattered around for managing windows from the taskbar or tweaking many of the items. In *PerlPanel* parlance, these items are known as applets – tiny programs loaded on an as-needed basis that boost the panel's capabilities. Applets cover a useful range of desktop tasks: mounting drives, taking screenshots, controlling the *XMMS* music player, entering commands at a mini prompt and locking the screen.

You can add the Gnome menu as another way to start apps, drop in applets to monitor CPU, battery, Wi-Fi or temperature status, and enable a notification area for programs such as *Gaim*. It's all well constructed and polished – each applet is accompanied by a small description and managing them is a doddle. Pleasingly, *PerlPanel* sports its own config tool, used for setting its location (only top and bottom at present – no vertical panels), size and icon set. This little extra gives *PerlPanel* plenty of extra points, as fiddling with config files isn't much fun for most users.

But what about performance? Memory use will depend on configuration and the number of applets loaded, but *PerlPanel* typically only used a few more MB than its Gnome equivalent. That's not spectacular, but because it doesn't need the whole Gnome infrastructure to run, it's much lighter overall. *PerlPanel* is an ideal addition to bare-bones window managers, where a good-looking, feature-rich and friendly panel finishes it off.

www.linuxformat.co.uk

9/6/05 5:50:19 pm

SMB SHARE BROWSER
Smb4K

■ **VERSION** 0.5.2 ■ **WEB** http://smb4k.berlios.de

SMB, the Server Message Block protocol used for sharing files and printers on Windows boxes, is catered for in Linux (and other Unixalikes) by the ever-popular *Samba* project. *Smb4K* brings a snazzy KDE front-end to the *Samba* suite, aiming for ease of use and a wide range of features.

You'll need versions 3.2.x of both KDE and *Qt* to compile *Smb4K* from source; once extracted, the normal **./configure**, **make** and **make install** (as root) procedure should build it without any difficulties. *Smb4K* drops an icon into the system tray on startup for easy access.

With a resizeable three-pane layout, *Smb4K's* interface fits snugly into lower resolutions and novices shouldn't have any trouble navigating their way around. A collapsible tree

widget on the left lists networks, hosts and shares found via scanning or by-hand mounting – there's also a tabbed info panel which provides at-a-glance details. *Smb4K* doesn't act as a file browser itself, because *Konqueror* does this job admirably, so the program just offers a quick preview feature to see which files are available. It's all sanely laid out and swift to boot.

Featurewise, *Smb4K* boasts a bookmarks system for quick access to regularly used shares, WINS server support, a search facility and the option to specify a default user/password combo (a useful time-saver). Similarly, it's pleasingly configurable, with stacks of low-level SMB/CIFS options available for tuning along with general interface settings. There's little in the way of documentation, but
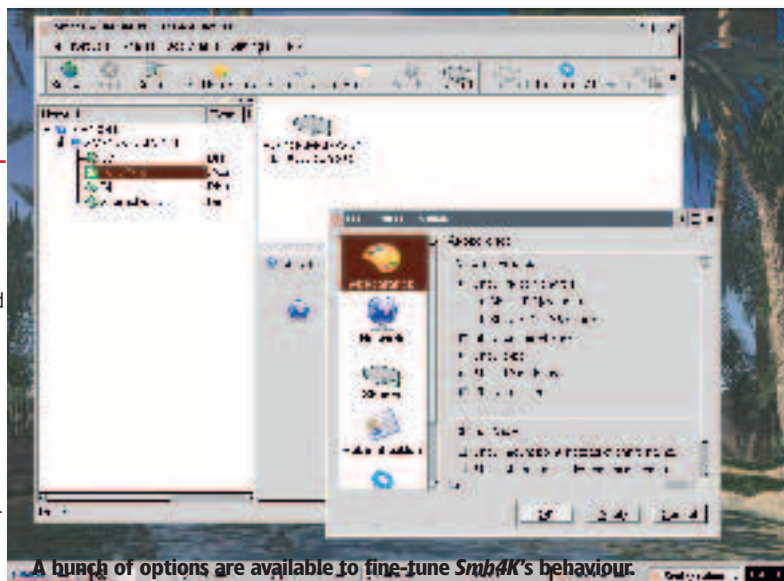
A bunch of options are available to fine-tune Smb4K's behaviour.

fortunately most of the options are self-explanatory, and anyone familiar with Windows networking shouldn't be too perplexed.

So, what about the crucial areas of stability and speed? *Smb4K* held up well in our testing, showing no signs of flakiness or sluggish behaviour. It's

clear that much effort has been put into improving the clarity and user-friendliness of the app. It could do with some spit and polish in places, and a proper help guide – but otherwise it's a solid little app, and deserves investigating if you're using KDE boxes on a Windows network.

AUDIO SEQUENCER
Jackbeat

■ **VERSION** 0.5.0 ■ **WEB** www.xung.org/jackbeat

Writing music is a great creative release for many, although the variety of software available can make it difficult to pick the perfect solution. Just as with graphics tools, where there is *Gimp* at one end and *mtPaint* at the other, there are music-making programs targeted at power users and beginners. Catering for the latter group, *Jackbeat* is an

audio sequencer that strives to make editing easy for newcomers, and is progressing well.

To get *Jackbeat* up and running, you'll need *GTK 2*, along with the *libsndfile* and *libsamplerate* libraries that can be found on our coverdisc. The *Jack* audio server is required too. Once these are installed, the regular build procedure will suffice – or

failing that, you could try one of the available binary packages. *Jackbeat's* interface is pretty bare, lacking in iconic goodness or polish, but it's a doddle to get started. Just load samples for the tracks listed down the left, set their pitch and volume, and then toggle the right-hand buttons to define when they're played.

This is more approachable than MOD tools like the fabulous *SoundTracker*, as simple tunes can be knocked together in seconds, although it's not as packed with features. You can alter the number of tracks, beats and BPM on the fly, and also edit the music as it plays. Top stuff. Various resampling options are available, organised from highest to lowest CPU usage; additionally, you can transpose the volume for all tracks at once, and the resulting music is saved in a tar archive containing both the samples and an XML file with the details.

There's little in the way of other features, although with the *Jack Rack* and *LADSPA* plugins you can add all manner of effects such as

reverb, amplification and filters on a per-track basis. This is an ideal setup. The program isn't over-burdened with features that advanced users might require and, as a result, is a pleasing sequencer that takes very little learning. It is an excellent way for you to start making tunes without getting bogged down in the complexities that seem inherent in music software.
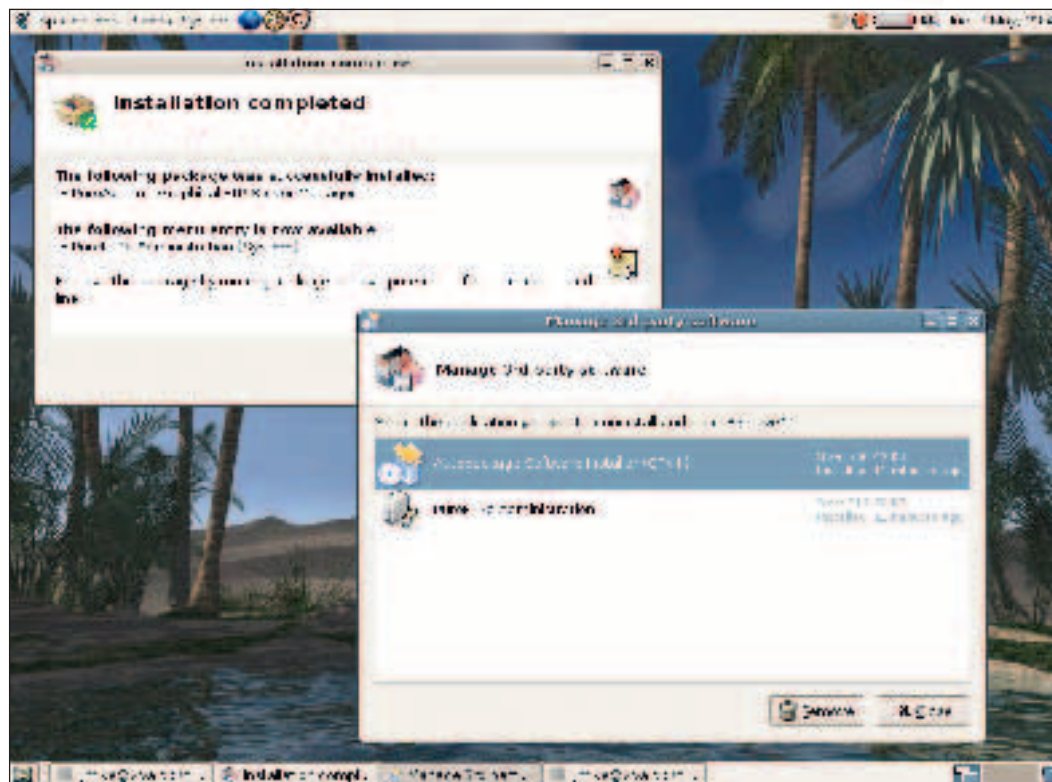

The *Jackbeat* music synthesizer in action – the depressed buttons show where samples will be played.

LXF69.hot 43   9/6/05 5:50:25 pm

SOFTWARE INSTALLER

Autopackage

■ **VERSION** 1.0.1 ■ **WEB** www.autopackage.org

The results of a successful installation, alongside the package removal dialog.

Software installation on Linux can be tricky at the best of times. Despite the emergence of numerous packaging formats, many of which offer superb flexibility and features, the process of installing a new application can still involve a long trek down dependency avenue. The speed at which distros and libraries develop makes it difficult for programmers to target a broad user base with one package; usually, several packages have to be assembled to cover most bases. Even distros with vast package repositories suffer in the long run as many new apps require library updates that may not be available without a total distro upgrade.

Various solutions have been proposed. One of these is *Autopackage*, a system designed to avoid the hassles of dependency hell and endless hunts for distro-specific packages. The goal: Joe User can simply hop on to a website, grab the *Autopackage* of his favourite app, and it will install (and fathom out dependencies) with minimum interaction. Sounds too good to be true? Well, this isn't one of those

'planning stage' SourceForge projects full of great ideas but no real-world implementation – it's a maturing system with some major programs already available.

Autopackage packages aren't too taxing to create. The specifics deserve a separate article, but if you have experience of creating packages in other formats (such as RPMs or Debs) there won't be any stumbling blocks working it out. Helpfully, the *Autopackage* team have written some marvellously detailed documentation explaining it all. Some of the apps available in *Autopackage* format are testament to the system's capability. *AbiWord*, *Firefox*, *Gimp* and *Gaim* – all well-known projects with masses of code – have been packaged up, and the range of apps is slowly (but steadily) improving as *Autopackage* generates more interest.

Kick back and watch

From an end user's point of view, *Autopackage* is a delight to watch in action. You don't even need to install any framework manually. Grab a .package file for an app, make it

executable, run it and the software does the rest. On virgin systems, it will fetch the *Autopackage* tools and *GTK/Qt* front-ends from the net (giving you copious feedback), then proceed to install the software. Progress bars and explanatory text abound as dependencies are calculated (either in the native package format or as stand-alone files), desktop menu entries are added and the app is made ready to run. On our test Ubuntu box, going from scratch to an *Autopackage*-enabled system with a small app installed took less than a minute.

To manipulate packages already installed, *Autopackage* includes a command-line tool offering file listing, removal, verifying and other operations. Anyone familiar with workalike CLI tools could use it, and for others there's a smashing graphical front-end which lists packages, their sizes and when they were installed, with accompanying icons. It doesn't flummox the user with excessive detail or technobabble – it just gets straight to the point. The

Windows 'Add/Remove Programs' dialog has been the inspiration here.

Even though *Autopackage* could simplify software installation beyond the efforts of other packaging formats and tools, the developers are more ambitious. On most systems, users are presented with two ways to 'access' a program – the installer, and the software itself. While this is no problem to those with computing experience, novices often don't understand the difference and it would be less confusing if there were just *one* way to use an app. Ideally, you could 'run' a program (ie install, then run) straight off a website, and then send it to a friend without any added complexity. It's a lofty goal, but *Autopackage* is one step closer to achieving this.

Calling all distros

Ultimately, whether *Autopackage* succeeds as the dominant installation system or disappears into the void of quixotic projects depends on developers, and distro vendors in particular. For all the talk of 'software Darwinism', where the technically best programs defeat the rest, a system like *Autopackage* needs strong backing to really make an impact.

But wouldn't it be fantastic if one binary package installed on all the major distros with no faffing around? It would make life sweeter for developers and end users alike. *Autopackage* could clear one of the last major obstacles on the road to widespread Linux desktop uptake, so we wish it well.

HOT GAMES

STRATEGY GAME
Freeciv

■ **VERSION** 2.0.1 ■ **WEB** www.freeciv.org

Sid Meier's *Civilization*, an empire–building strategy romp that amassed a massive fanbase in the early nineties, is still highly regarded among game critics today. Naturally, with thousands of open source coders scratching all possible itches, it was inevitable that lookalikes and workalikes would appear.

The one that we're looking at here, *Freeciv*, has been in development for almost a decade now. Handily, it offers a selection of front-ends, including *GTK1*, *GTK 2* and plain old *X* widgets. This is a great bonus when building it on older distros. Once compiled, running **civclient** at a shell will bring up the game, ready to select a play mode.

Cosmetically, *Freeciv* isn't anything to write home about – the sprites and landscapes are attractively drawn, but it doesn't have many flashy effects. Still, interface design is far more important in strategy games,

and in this department *Freeciv*'s jam–packed menus and keybindings provide access to all aspects of play. Essentially, your task is to build cities and armies, defeating other players along the way and looking after your inhabitants as the centuries pass and technology develops. By managing military units, keeping track of food production and doing scientific research, your position strengthens.

A gargantuan array of settings is available, affecting all areas of the game. Being able to tweak the amount of hills and water on the terrain, for example, the behaviour of a city's inhabitants or your military options will make *Freeciv* a tougher challenge for experienced players. Following the handful of supplied scenarios is a good way to learn the game's workings, but it's in networked multiplayer mode where *Freeciv* really comes to life. It's not hard to find active servers with players waiting; a

certain amount of etiquette and familiarity is crucial here.

Freeciv demands patience and attention – you can't just fire it up for five minutes of fun. The absorbing gameplay is its great strength, though, and with the lively community and

stacks of documentation there's plenty to sink your teeth into. It's unmissable for all *Civilization* fans, but even if you never played the original you should also give it a go. This is an epic, engrossing game with masses to learn.

Time to build a few new Commercial Zones... Oh wait, wrong game. Again.

STRATEGY GAME
Glest

■ **VERSION** 1.0.10–r7 ■ **WEB** www.stud.uni–karlsruhe.de/~uxsm/glest

A bit of a strategy game theme this month, with *Freeciv* reaching 2.0 and *Glest* emerging as a promising battler. Last year, *Glest* picked up a video game design award from Sony,

and even though it's still under development it deserves the acclaim. With customisation in mind, *Glest*'s developers are working on the game's engine but have already included an

entertaining single–player game. The quickest way to install is via the *Loki* installer – just make it executable and run it, with everything else completed with a friendly dialog.

Glest is gorgeously presented. With thundering orchestral music opening the title screen, and the camera panning across a rainy landscape with castles and trees, you know you're in for something special. These touches continue into the game – top–notch sound effects, great textures and lighting, and plenty of graphical splendour all round. Arguably, visuals aren't vital in games with a heavy strategic bent, but the atmosphere created in *Glest* is lovely to see. You'll need a decent graphics card, but still, the graphics can be tweaked for slower boxes.

In the game itself, you're responsible for directing a band of battlers across a selection of

landscapes, collecting resources and obliterating enemies in spectacularly fierce battles. The strategy side comes into play as you manage resources, assign the right team members to the right jobs and explore the surrounding territory. Each character has a different strength – some are geared towards fighting, whereas others can be used for gathering or even generating new accomplices. It's dangerous to hang around though; once your fort is destroyed, it's game over.

There isn't a huge amount to explore in the game at present, and a multiplayer mode is sorely missing – being able to play over the network would be great. Nevertheless, the single player mode is still top fun, and the glorious presentation is worth experiencing. *Glest* has heaps of potential, and we'll be keeping a close eye on its progress.

Just like the streets after closing time on a Saturday night (but with more dragons).

LOG FILE ANALYSER
Epylog

■ **VERSION** 1.0.3 ■ **WEB** http://linux.duke.edu/projects/epylog

Log analysis tools are a key part of a busy sysadmin's armoury. When you have to keep track of several servers and desktops, rummaging through plain-text log files to pinpoint problems can be a chore, especially when the logs have grown to gigantic sizes. Analysis software is a massive time-saver in these cases, producing summaries and reports of system activity in an easy-to-digest form, and we've looked at a few before in HotPicks.

Epylog is an industrial-strength log analyser, designed with large clusters in mind where a number of machines are outputting their logs to the same location. It's written in Python and you'll need version 2.2 of that scripting language to run *Epylog*, along with *libxml2-python* (there are no obscure dependencies, thankfully). The regular build procedure drops the main script into **/usr/sbin**, with its supporting config files in **/usr/etc/epylog**.

When *Epylog* is run straight from the shell prompt, you can specify a duration for it to cover such as logged messages from the last hour, day or week. Alternatively, it can be dropped into a *cron* job to automate the process. *Epylog* runs through various files in **/var/log**, plucking out information for authentication, mail servers, packet filtering and others, and the 'Weedeater' aims to avoid filling the output with non-essential entries.

This output is either in plain text form mailed straight to root or a more attractive HTML version. Here, colours and basic tables organise the results into a simplistic but easy to read form, and unparsed strings (log entries that *Epylog* can't decipher) are quoted at the bottom just in case anything dangerous needs attention.

Epylog is satisfyingly easy to install and configure, and while it doesn't have all the trimmings of some other analysis programs we've seen, the reports are ideal for at-a-glance checking of a machine's (or bunch of machines') behaviour. To top it off, it's well documented and runs at a very respectable pace too. Worth a look if you need an all-round log analysis tool.

***Epylog*'s HTML output isn't all that snazzy, but it does the job well.**
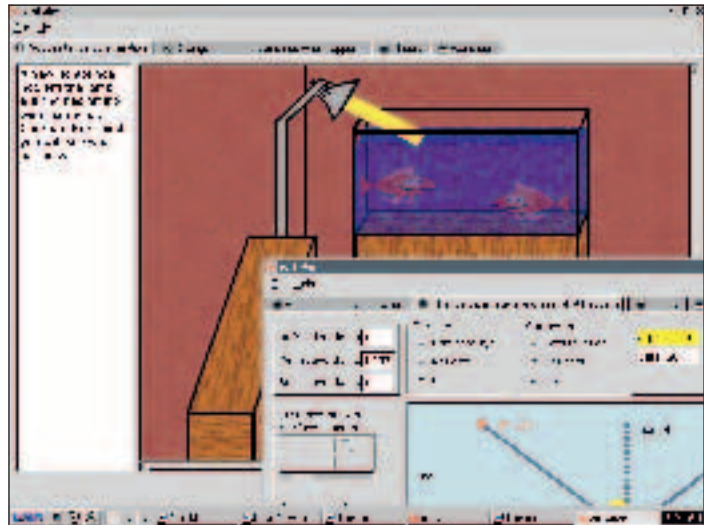
A lesson in optics, complete with a cute demo, thorough explanation and dodgy Dutch to English translation.

EDUCATION SOFTWARE
EduKator

■ **VERSION** 0.6.1 ■ **WEB** http://users.pandora.be/muylkens/edukator

With the right approach, computers can be highly effective learning tools, providing a degree of interactivity and reward that plain old books just can't match. *EduKator* aims to teach physics and maths with illustrations and hands-on involvement.

As with many KDE apps, *EduKator* doesn't skimp on gratuitous use of Ks wherever possible, giving us yet another name that would be more at home in Greenlandic or dolphin chatter rather than English. Building from source should be hassle-free – you'll just need recent versions of the KDE and *Qt* development packages.

When started, *EduKator* pops up some brief explanations of the topics covered. These suffer in places from patchy Dutch to English translation. The language is awkward in places and could do with some editing; in general, though, it's comprehensible. This isn't a major cause for concern in a science-based teaching tool, though. Accompanied by animations and point-click interactivity, the four sub-programs illuminate optics, Archimedes' principle, doublerates and extreme mathematical problems.

The program's interactive aspects stand out when you modify the parameters in each scenario. With slider and spinner widgets aplenty, you can get an instant representation of how your changes would work out –

it's much more engaging than having a pre-set list of possibilities. In the optics demonstration, for instance, you can alter the surface, angle and co-ordinates and see a graphical display of the results straight away. Some detailed theory is provided after experimentation, which explains the underlying science with a more technical slant.

Then, to top it all off, a handful of exercises are included, posing questions related to the topic and elaborating on answers after a mouse click. These vary in complexity to tax students at different levels.

We see a lot of Linux educational software directed at young children, so *EduKator* is refreshingly advanced. Despite the dodgy English in places it's well put together. Worth a try if you're doing a related subject or just want to give the grey matter a tune-up.

9/6/05 5:50:44 pm

HotPicks REVISITED

## TEX EDITOR
Kile

■ **VERSION** 1.8b2 ■ **WEB** http://kile.sourceforge.net

It's been nearly two years since *Kile* snapped up our Hottest Pick award in *LXF49*. A KDE-based editor for the *TeX* typesetting system, *Kile* impressed us greatly with its broad feature set, polished interface and overall stability. If you haven't come across it, *TeX/LaTeX* is a firm favourite in the scientific and academic worlds for producing high quality, complex documents – many users find it much more capable and usable than typical word processors.

You'll need version 3.2.x of both KDE and *Qt* to build *Kile* from source. As this is a test release leading up to 1.8 final (hopefully available as you read this), no binary packages have yet been assembled. Few changes have been made to *Kile's* cluttered but workable interface; the resizeable multi-pane view for navigating and entering code remains, as does the handy embedded *Konsole* terminal.

Since our look at 1.6, the busy *Kile* team have grafted away at a pleasing collection of new goodies:
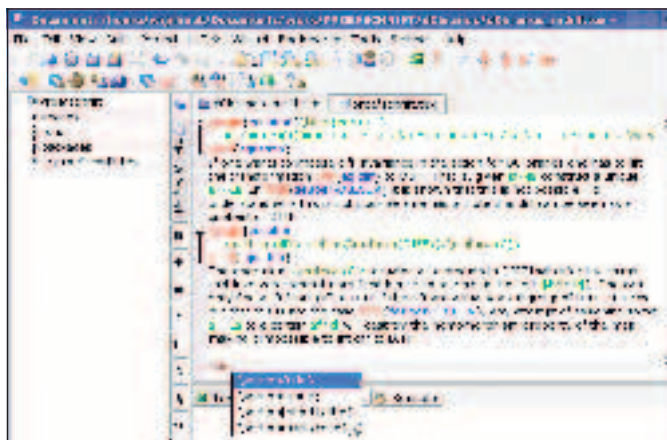
***Kile's* interface crams plenty into a small space, but it's not overbearing.**

autocompletion of *TeX/LaTeX* commands; a System Check function that makes sure all necessary components are installed to avoid cryptic errors; a QuickPreview option to generate output for a selected chunk of the document (great time-saver); the ability to use other systems than *LaTeX*; plus a few new wizards and plenty of bugfixes.

These new features alone make *Kile* deserving of another try if you gave the older version a spin. It's shaping up nicely, and despite it being a beta release we didn't encounter any critical stability problems in our testing. This is a solid performer all round, and all KDE-loving *TeX* users should take a look at it.

CLI-BASED GUI
OneFinger

■ **VERSION** 4.0 ■ **WEB** http://onefinger.sourceforge.net

Eagle-eyed readers will note that it's not long since we featured this curiosity in HotPicks – *LXF61* to be exact. *OneFinger* is an attempt to throw the GUI and command line into a mixer, pouring out a surprisingly sweet-tasting app that takes a whole new approach to using a computer. Built with the KDE libraries, *OneFinger* needs that desktop's Python bindings installed to get it up and running.

In a nutshell: you create the functional equivalent of shell commands by clicking in boxes associated with actions and files, and the program strives to learn the way a user works – that is, which programs are commonly used. It's rather strange for someone coming from a traditional desktop background, but works efficiently after a bit of practice.
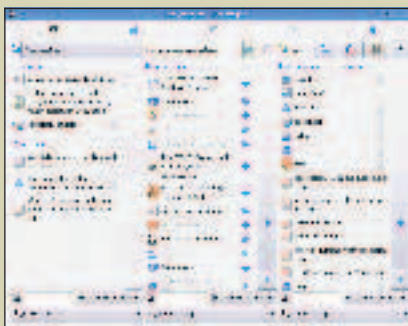
***OneFinger* looks almost unmanageable at first – after a bit of use it makes more sense.**

It's great to see Maurizio Colucci, *OneFinger's* author, pay attention to even minor suggestions that the community has made for improvements, and the program is growing more versatile and mature as a result. Version 4.0 has seen more

changes than a train journey from Dundee to Barnstaple: a *Kicker* applet for the KDE panel; history list taken from other KDE programs; executing commands as root; selecting files based on patterns; general speedups; and tons more. A lot of work has been done on usability, with glitches corrected, certain widgets made more prominent, clutter removed, new toolbars, an info bar, and better integration with the selected KDE icon set. Updates have been made all over the source tree – it's still reliable though.

Even if the concept of *OneFinger* doesn't immediately appeal, we recommend everyone tries it out – it's a satisfyingly fresh way of working. **LXF**