

# LINUX

www.linuxformat.co.uk

## FORMAT



## HOW TO OPTIMISE YOUR LINUX BOX

Practical tips and advice for a faster,  
more efficient server or desktop **p52**



## NETBOX CUBIT

Small and well specced – is this the best Linux mini-server? **p26**

## PRESENT WITHOUT POWERPOINT!

Get to grips with  
OpenOffice.org's  
MS-beating  
presentation  
creator **p84**



## Helix Explained

Streaming digital media the  
Real Networks way **p64**

**CORBA** Middleware without SOAP **p76**

**PHP ENCODING** Protect your code **p34**

**WEBMIN** Admin tool reaches 1.0 **p30**

**MESSAGING** Huge client roundup **p40**

A box for RealSoft3D software, showing a 3D rendered scene with a car and a person.

## RealSoft3D

Affordable, easy to use,  
powerful 3D modelling and  
rendering on Linux **p22**

**FOSDEM 2003** Interviews and  
opinions from the developer  
event of the year **p12**

DVD issue also available Printed in the UK  
**LXF36 JANUARY 2003** \$5.99  
 9 771470 425019  
(£5.99 outside UK and Republic of Ireland)

THE UK'S BEST-SELLING LINUX MAGAZINE

# Faster, faster...

**W**hen Linux was in its infancy, there wasn't much scope for system tweaking. Distros were typically bare-bones affairs anyway, and with not much software, there weren't too many things you could actually squeeze more out of.

These days the average Linux distro comes on multiple CDs or a DVD, and contains hundreds of packages. Even if you are conservative with what you install, you are still looking at a huge codebase. And chances are, everything from the kernel up will have been compiled and setup with a 'one size fits all' philosophy. That's fine from the point of view of getting a system up and running without hassle, but isn't ever going to give you the best performance.

To get the best from your Linux box, rather like a Rally car team, you have to dump out all the bits that don't suit your purpose, tweak up the engine and customise. Where do you start with Linux? Our main feature this month will show you a whole variety of ways you can get

better performance from your system, from five minute tweaks to more extensive customisation of software, helping you squeeze every extra clock-cycle from your hardware. Take a look at page 52 to get started, and if you have some more tweaking tips, do send them in, we'd love to hear them.

We're also taking a look at alternative GUI driven desktop OSs this issue. There are certainly things to be learnt from different approaches, and we think you'll find it educational and thought provoking.

On the reviews front we have a very interesting micro-ATX box, some fantastic 3D software, a look at the full release version of *Webmin* and lots more, including systems to obfuscate PHP code.

If all that isn't enough for you, there's over 20 pages of tutorials, and don't forget to take a look at *Linux Pro!* We really appreciate your feedback so please keep emailing us.

Hope you enjoy this first issue of 2003!



**Nick Veitch** EDITOR

Speed up your Linux server or desktop to optimum efficiency – we give you the inside knowledge **p52**

Small, colourful and undemanding – Cubit break away from beige **p26**

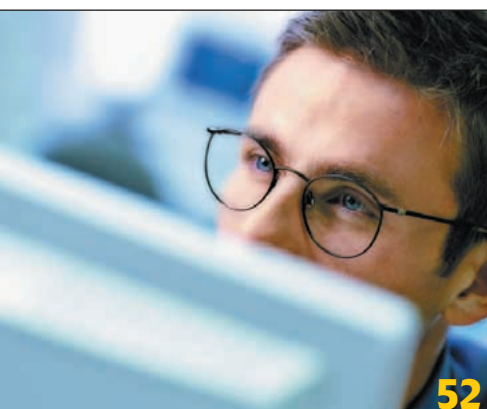
3D software for advanced modelling and rendering from Realsoft3D **p22**



**26**



**22**



**52**



## AIMS OF THE MAGAZINE

**Linux Format** is a magazine dedicated to Linux and the Open Source community. We aim:

- To provide the most accurate, unbiased and up to date information on all things Linux.
- To promote the use of Linux in business and the home, for servers and on the desktop.
- To support the Open Source community by providing a resource of information, and a forum for debate.
- To help all readers get more from their Linux experience by providing insightful and useful tutorials.

## MEET LINUX FORMAT'S TEAM OF WRITERS...



**Andrew Channelle**  
Now studying 'culture' or some such nonsense, Andy still finds plenty time to write the news!



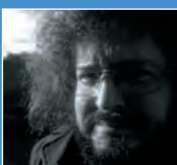
**David Coulson**  
Our Answers guy is a networking and security guru with plenty of sysadmin experience.



**Mike Saunders**  
After rounding up so many pieces of Instant Messaging software, Mike is all chatted out.



**Hoyt Duff**  
Fishing pier proprietor Hoyt spends his spare time installing Linux on anything that stays still long enough.



**Charlie Stross**  
Master of Perl, Charlie has been writing about Linux for more years than anyone can remember.

**David Cartwright**  
Veteran journalist and Linux consultant, he knows his stuff when it comes to real-world Linux usage.

**Jono Bacon**  
Jono is a core KDE developer, web developer and writer. Jono is also a musician and sound engineer.

**Richard Drummond**  
As he sails off across the Atlantic, Rich keeps telegraphing articles to the UK.

**Richard Smedley**  
Discovered under a bush on the LXF allotment, only Rich2's Linux experience saved him from composting.

**Paul Hudson**  
PHP veteran Paul likes nothing better than implementing everything as a script.

## CONTACT US

Letters for publication:  
[lxf.letters@futurenet.co.uk](mailto:lxf.letters@futurenet.co.uk)

Subscriptions/back issues:  
[subs@futurenet.co.uk](mailto:subs@futurenet.co.uk)

Technical help/Ask the Experts:  
[lxf.answers@futurenet.co.uk](mailto:lxf.answers@futurenet.co.uk)

Disc problems:  
[lxf.support@futurenet.co.uk](mailto:lxf.support@futurenet.co.uk)

General enquiries:  
[linuxformat@futurenet.co.uk](mailto:linuxformat@futurenet.co.uk)

Website: [www.linuxformat.co.uk](http://www.linuxformat.co.uk)

Or send your letters to:  
**LINUX Format, Future Publishing,**  
30 Monmouth Street, Bath, BA1 2BW  
Phone: 01225 442244  
Fax: 01225 732295

More contact info on p114



LXF36 January 2003

# Contents



Welcome to another jam-packed issue of *Linux Format*, your guide to all things Linux!



## COVER HIGHLIGHTS

# Optimise your Linux box **52**

Server or desktop, we give you the top tips and tweaks for max efficiency

## 20 Xandros

Easy to install. Easy to use

## 58 Desktop dreams

Building a better desktop OS; how Open source gives you the freedom

## 64 WhatOnEarth is Helix

A universal streaming media platform

## 76 CORBA

Middleware demystified – cross platforms



**'The Internet has become the best way to chat at a distance as unmetered access and broadband has made it convenient'**

**40** RoundUp – Instant Messengers

**'Linux users will be familiar with some of the themes at play here: determined developers, who won't compromise'**

**58** Desktop dreams – OSS developments

## LINUXPRO

**Parallel computing** high performance from OpenMP **Mitch Kapor** Open Source and the next generation Agenda **Accessibility** Linux software for equality of access





**26 Cubit** – space efficient, energy efficient, and good looking, too

## 6 News

Analysis of the latest Linux stories

## 12 FOSDEM 2003

We interview the speakers

## 14 Mailserver

Your experiences shared

## 20 Xandros review

The Corel desktop Linux reborn

## 22 Realsoft 3D review

Advanced modelling and rendering

## 26 Cubit review

Small, quiet and definitely not beige

## 28 Kohan AG review

WineX version of the new prequel

## 30 Webmin review

Remote administration software

## 34 PHP Encoders reviews

Protect your code

## 38 Books reviews

From hackers to office workers

## 40 Roundup Instant Messengers

Chat around the world

## 46 Hot Picks

GStreamer, Anjuta, librash, Scribus

## 52 Optimisation

Leaner, meaner and fitter Linux

## 58 Desktop dreams

Building a better desktop OS

## 64 What on Earth is Helix?

The future of streaming media?

## 70 System Programming tutorial

Sockets – and their uses

## 74 Perl tutorial

File manipulation and dir searching

## 76 Corba tutorial

Middleware to cross platforms

## 82 Java tutorial

DIY bean building

## 84 OpenOffice.org tutorial

Better presentations with Impress

## 88 PHP tutorial

Media generation: jpegs and pngs

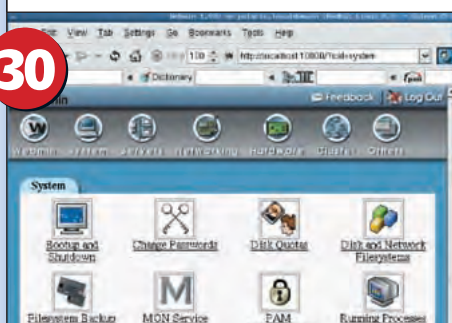
## 92 Answers

Your queries answered

## 110 User Groups

Local support and fun

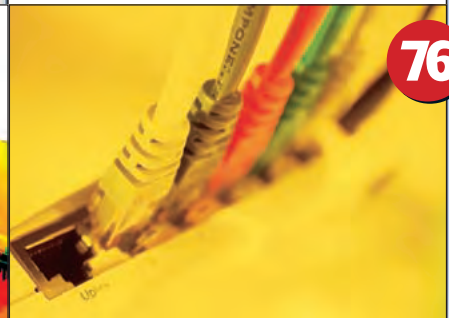
30



64



76



# COVERDISCS

A DVD or 3 CDs packed full of the latest Linux goodies **100**



## CDS A, B AND C

**Racer** 52 cars to choose from in this great 3D simulation; **Freeduc** Knoppix-based educational distro – runs from the CD; **Linux From Scratch** build your own Linux from source; **GNOME Meeting H.323** Video conferencing app; **Opera 6.1** latest version; **Film GIMP** give your home movies the edge; **phpOpenTracker** who's been visiting your website – and where did they go?; **Tuxpaint** for the kids



## DVD

**Evolution** latest version of the PIM/calendar/e-mail client; **Freevo** turn your PC into a personal video recorder; **OpenZaurus** open source improvements; **PilotLink** Palm sync

Please read the coverdisc instructions starting on page 100 before installing from coverdiscs!



# SAVE MONEY!

# SUBSCRIBE TO LINUX FORMAT

See page 98 or phone 0870 4448645





# Newsdesk

Governments looking to Free Software; Linux in the living room; UK govt reconsidering "fair use" rights; OOo scripting extensions; HP gains; Linux TapeCert updated; New GNOME board; Mandrake go parallel.

## FREE SOFTWARE STATES

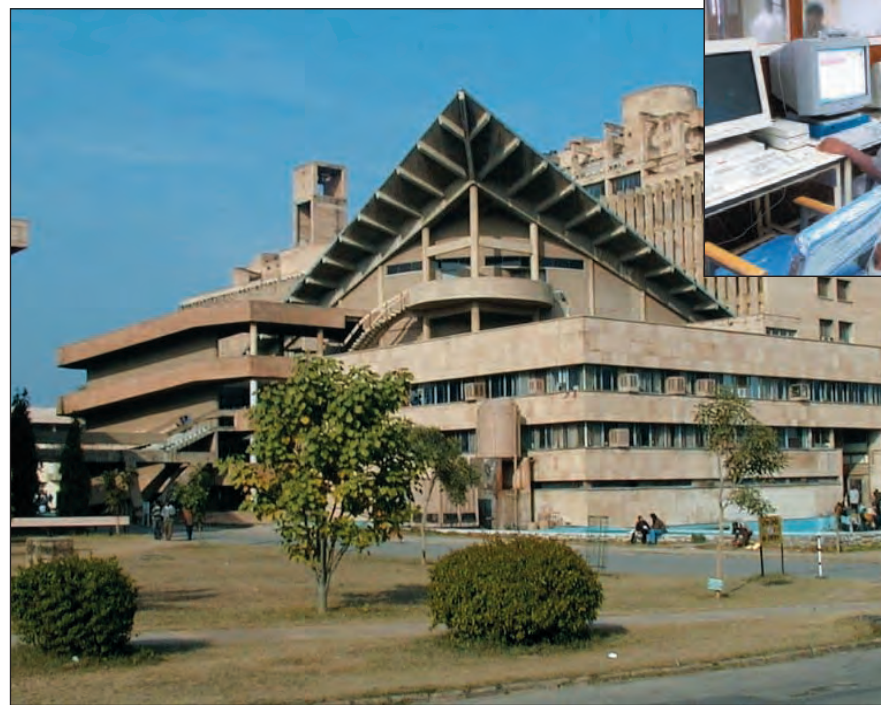
# More governments turn to open source software

**A**ttracted by lower costs and more robust security, more and more governments across the globe are turning to open source software and Linux for their computing needs.

The Japanese and Indian governments are the latest in a long line of national and regional institutions to begin investigating the benefits of free software. As any move away from the dominant MS operating systems and applications could potentially affect millions of desktop and server users, the corporations and organisations which have invested so much in their products regard winning this 'battle' as paramount.

The importance of India especially has been underscored by an three-year investment strategy by Microsoft – the company's biggest non-US investment – which will see \$400 million pumped into the nation's IT infrastructure. The Bill and Melinda Gates Foundation has also pledged a \$500 million donation to the country, though Gates insists it is unrelated to the current marketing push.

India's reliance on Windows is, like elsewhere, almost total with 95% of the country's computers using the OS, but changes in attitude and the need to stretch limited resources means that the government is now promoting Linux as the 'software of choice'.



**Students at the Indian Institute of Technology will get extensive Linux experience.**

The first sector to see real change is in the nation's Ivy League educational establishments such as the Indian Institute of Technology, Tata Institute of Fundamental Research and Bhabha Atomic Research Center, where Linux is set to become the standard OS. This is an important sector to 'break' as many of India's future developers (which make up an estimated 10% of the world's development pool) will pass through these institution.

In addition to the courts, various ministries, the regional government of West Bengal and the National Stock Exchange, India's Centre for Supercomputing – C-DAC – has also moved wholesale to Free Software solutions. The Department of Electronics are also helping to fund the OS localisation effort in concert with the IIT Kanpur Centre for Natural Language Processing at Hyderabad.

On a recent trip to India, Bill Gates shrugged off the threat from Linux, suggesting it is unlikely to gain mainstream acceptance. Richard Stallman was in the country at the same time as Gates in his attempt to convince the government of the merits of Free software, and said his discussions had been positive. 'I have met people in [the Indian states of] Kerala, Karnataka, Andhra Pradesh, and some people in the government

## LINUX APPLIANCE

## Dial-up video recorder

## Embedded Linux developer

MontaVista has inked a deal which will see its OS built into NEC's latest Home AV Server product. The AX10 is a TiVo style hard-disk based personal video recorder (PVR) that is also equipped with

networking features so that audio or video can be streamed to any PC (or other connected device) in the house. One of the AX10's more unusual features though is the ability to set up a recording via an i-mode compatible phone.

## It's not all one way

## Equal rights for proprietary software

### The Initiative for Software Choice is a lobbying group set up to dissuade governments from implementing OSS-only IT policies.

America's Department of Defence have come under fire from a group called the Initiative for Software Choice (ISC) for 'promoting' the use of Open Source Software. The ISC, who have the backing of Microsoft, Intel and Cisco, was responding to a report by defence contractor Mitre which said that Free and Open Source Software play a significant – though largely unrecognised – rôle in the DoD. The report (online at [www.egovos.org/pdf/dodfoss.pdf](http://www.egovos.org/pdf/dodfoss.pdf)) concluded that Free and Open Source Software (FOSS) applications are most important in the areas of Infrastructure Support, Software Development, Security, and Research. 'One unexpected result was the degree to which Security depends on FOSS. Banning FOSS would remove certain types of infrastructure components (e.g., OpenBSD) that

currently help support network security. It would also limit DoD access to – and overall expertise in – the use of powerful FOSS analysis and detection applications that hostile groups could use to help stage cyberattacks.'

Misinterpreting the recommendation to not ban 'FOSS' as a call to ban proprietary software, ISC said the DoD should not 'openly promote open source' citing the 'viral' nature of the GPL.

'It makes sense for companies to be highly risk-averse in this area, striking a more defensive posture when confronted with software development that may implicate GPL code,' the ISC report said. 'Commercial and hybrid software developers generally do not want to risk losing their investment.'

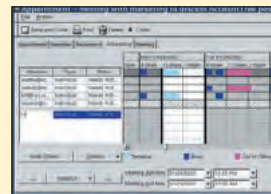
of India," he said. 'What they say sounded favourable.'

## Meanwhile in Japan

In Japan it is security rather than cost which inspiring a long, hard look at Linux and other open source solutions. The government has put together a focus group of experts – including members from Microsoft – in order to investigate the security and maintenance aspects of migrating e-

government services away from Windows. The group will have a budget of some \$410,000 to monitor other institutional switchers. Tatsuya Kawachi said the study would be finished in March 2004. 'This is not something we would undertake on hearsay.' The China Post claimed Japan was eager to catch up with their economic rivals – especially China – who are already well on the way to adopting Linux in an official capacity.

## NEWSBYTES



■ **Ximian** have spruced up their *Evolution* groupware client for its 1.2 release. The company seem to be on the hard sell with their *Connector* software at present so the latest *Evolution* publicity makes a great deal of the apps potential to work in tandem, or instead of, MS's Outlook.

■ Also on the upgrade cycle is **Codeweaver** who've released version 1.3.1 of *CrossOver Office*. Still no sign of *CrossOver* branded products to bring either *Photoshop* or *Dreamweaver* to Linux though!

■ **LinuxMAX** is a nascent web community in need of citizens. The site (<http://linuxmax.net/>) is handsome, efficient but the forums are a bit *Marie Celeste*-esque.

■ Guitar string maker **Eddie Ball** have completed a total move to Linux and are boasting their MS-free status. The move was the result of the company being fined for using unauthorised software after a disgruntled employee blew the whistle to the Business Software Alliance.

■ **Multi-threading** is flavour of the month at the moment with noted improvements in Linux's threading implementation, which theoretically allows the kernel to run hundreds or even thousands of threads, and the launch of Intel's first consumer targeted Hyperthread-enabled CPU. Linux has supported Hyperthreading (where the software 'sees' two CPUs instead of one) since early in the 2.4 cycle.

■ **IBM** are recycling and refurbishing old technology in Japan as part of the their 'PC Long-Life Service'. The machines, which are judged to be 'too old' for Windows will come with a choice of Linux distributions and should cost around \$66 each with a minimum purchase order of 10 PCs. Big Blue also have a decent tutorial on its DeveloperWorks site about making the best use of old hardware, hoping to buck the trend of treating last year's technology as through it were disposable. Check out <http://www-106.ibm.com/developerworks/Linux/>

■ After months of speculation, one of Australia's biggest companies is moving much of its IT infrastructure to a mixture of **Linux** and **Solaris**. While Linux will handle most of the tasks, the company are concerned that it won't scale enough to deal with their enormous billing systems.

## David Cartwright

David Cartwright is an IT consultant who specialises in providing Linux systems and solutions.



## COMMENT

## Be very vigilant

“ A friend of mine had his Web server hacked recently.

He thought he'd managed to eradicate the intruder and block the holes up, but a couple of weeks later the same thing happened again. A real pain, because the only real way forward was to reinstall the machine from scratch reformat, plonk a new version of SuSE on it, and rebuild the site.

The machine wouldn't even reboot in its hacked state, the root kit the intruder had installed had replaced too many key executables for it to stand any chance of starting up.

He had made reasonable efforts to install patches as they arrived, the passwords were chosen pretty sensibly, there weren't any stray user accounts lying around with null or default passwords, and so on.

The interesting thing is, though, that my friend noticed the attack because he looked for it. There were no tell-tale signs in day-to-day operation that anything was wrong – the root kit took care of that (The new version of *ps* tells you about every process except the rogue ones, for instance). Which is actually quite unusual even in some larger companies I've worked for.

It's all very well putting all manner of protection passwords, firewalls and such like in place, but what happens if someone gets in and you can't tell they're there? The moral of the story is: just because you've protected yourself, don't assume you're protected. With the best will in the world, there's always the chance you've misconfigured something, or there's a bug in the OS or server apps that you don't know about.

Monitor your servers closely and you may just thank yourself one day.



## COPYRIGHT CONSIDERATIONS

# Partial success for Euro DMCA campaigners

**A** concerted campaign to stop the UK ratifying the European Copyright Directive (EUCD), commonly known as the EuroDMCA, has succeeded in postponing its introduction until mid-2003. And as with the recent attempt to extend the RIP act, the government has admitted it has to go 'back to the drawing board'. However, sources suggest another attempt at getting the bill passed will be made in March.

Reporting the timetable slip, the UK Patent Office said almost 300 responses to the initial consultation document had been received from interested

parties by the October deadline, and it would take some time to analyse them.

"The sheer volume of replies, together with the fact that many present detailed arguments and suggestions for alternative drafting, mean that the present analysis will not be completed quickly. A full and proper consideration of all the responses is our principal concern, and the natural consequence of this is that the deadline set by the Directive for implementation is no longer attainable."

The deadline for implementing the EUCD was originally 22<sup>nd</sup> December 2002, but the Patent Office admits there is now no way of completing its work by then. "We do remain committed to the

earliest possible implementation of this important Directive. We will endeavour to implement the Directive by 31<sup>st</sup> March 2003 at the very latest"

The EUCD will extend current UK copyright law to 'harmonise it with the rest of Europe'. However, opponents such as the Campaign for Digital Rights (CDR) claim the directive uses such a broad brush traditional "fair use" rights will be lost and circumventing copy protection will be just as illegal as cracking a CD to share via *Kazaa* or *Morpheus*.

"By enacting Article 6 in local legislation, the governments in EU are signing away much of the power they

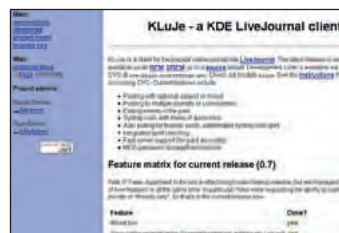
have to control the balance in copyright law, and giving it to the large companies that have the resources to develop these technologies and push them into the market place," The CDR said.

Article 6 of the Directive also has serious implications for cryptography and bug reporting as publishing the details of, say, a security hole in *IE* that gives a cracker access to 'protected works' could trigger a prosecution. Under the EUCD, posting details of the bug could lead to the poster being sued by the owner for facilitating the theft of their works. CDR says this will inevitably lead to more insecure software as 'ethical hackers' think twice about reporting their findings.

## Linux Web Watch/



LiveJournal – widely supported.



KluJe – client software for KDE.



Live Lizard – Mozilla-based.



Mambo Site Server – CMS.

## Publish and be damned

Media production has been wrestled from the hands of the establishment, and freedom of expression is but a download away.

WHEN CAXTON INTRODUCED BRITAIN to printing it was claimed that the technology would bring publishing to the masses, which was sort of the case. However, the means of publishing were still owned by those with a rather large pile of cash. Not so today, where the Internet was promoted with a similar claim to Caxton's: no longer would your thoughts have to be mediated through the tastes, prejudices and talents of others. Many to many publishing is with us. Oh, rejoice!

The easiest way to get your thoughts noticed is by signing up to one of the many 'Blog' sites, and one of the most widely supported is Live Journal ([www.livejournal.com](http://www.livejournal.com)). While you can create your entries online, a number of Linux projects have sprung up to bring Live Journal publishing to your desktop, meaning you can take more time and care composing your rants. There are versions of the client software for all platforms, but Linux is unusually blessed with them. If you're a KDE devotee, *KLUJe*

(<http://kluje.sourceforge.net>) is a decent – though poorly named – client that has proved to be stable and feature-rich in daily use. GNOME users will be better off with *Drivel* (<http://sourceforge.net/projects/drivel/>) which offers a very similar feature-set with footprints all over it.

Flying the flag for cross-platform goodness is *Live Lizard* (<http://livelizard.mozdev.org/>) which brings Live Journal functionality to a Mozilla-based app. *Live Lizard* has all the options for mood icons, music choices

etc and, as it's based on *Mozilla*, is about as OS promiscuous as an application can get.

If the thought of using someone else's layout and schemes appals you, the only option is to develop your own portal and, of course, you'll not be short of a Content Management System (CMS) or two. These systems are really worthy of a full round up, but if you're in a hurry both *EZ Publish* (<http://ez.no/>) and *Mambo Site Server* ([www.mamboserver.com](http://www.mamboserver.com)) are GPL solutions worthy of appraisal.

## NEWSBYTES

■ As well as trying to put together a world-class browser suite (latest release: 7.2), the folks at **Mozilla** have to come up with a new name for their *Phoenix* browser after a complaint from the BIOS maker. A revamped and renamed version 0.5 should be out as you read this.



■ No blockbuster distro releases this month (!) though **SuSE** are shipping the 'next generation Enterprise Server' based on the newly minted United Linux base. Designed for file, print and web serving SLES is available for a range of architectures including 32/64-bit Intel and IBM's eServer series.

■ The Register ([www.the-register.co.uk](http://www.the-register.co.uk)) recently carried a rumour that **Sun** may attempt to get Solaris LSB certified, and it wasn't even close to April 1<sup>st</sup>.

■ **Volvo** have jumped on the band wagon and migrated their design systems to Linux. While other auto manufacturers have utilised Linux for crash testing (See *Newsdesk*, LXF 35), Volvo appears to be the first to roll out the OS for its design processes. *Vorsprung Durch Penguin?*

■ **Sun's** Chief Technology Officer John Fowler has revealed that Sun are finally ready to commit to GNOME and make it the default desktop on the next version of Solaris, potentially bringing the GNOME experience to millions of users worldwide. In an interview on OSNews, Fowler said the company planned to create GTK+ bindings for Java which would lead to a consistent look across all Solaris apps.

■ Users wanting LinuxPPC without the expense of Apple hardware now have another option. **Terra Soft**, who also maintain the Yellow Dog Linux distribution have announced plans to offer both a Teron ATX PPC mother board and a complete machine tailored to YDL, as well as a range of genuine Apple Macs with Yellow Dog pre-installed. [www.yellowdoglinux.com](http://www.yellowdoglinux.com)

■ Though no stranger to Linux, **IBM** have so far yet to release a server capable of running the OS on its own, without having to rely on AIX. But that's all changed with the release of a modified p630 capable of booting SuSE Linux without aid from IBM's proprietary OS. Priced around the \$15,000 mark, the p630 is built around the 64-bit PowerPC processor with 2GB RAM and is expected to compete with Sun's UltraSparc and Hewlett Packard's Itanium solutions in the lower end.

## LINUX LEADERS

## HP claim the number one server spot

**Hewlett Packard are claiming to** have captured the biggest market share for Linux servers for the first three quarters of 2002. According to International Data Corporation (IDC), HP increased its share to 30.3%, shipping twice the number of servers as IBM and taking over Sun (in the UNIX sector) for the first time. HP also said revenue had increased by over 30%.

Senior Vice President Mary McDowell said the results showed a welcome upward trend in the server market. 'We are showing solid gains in unit share sequentially in EMEA, where we hold No. 1 market share in every country, and in Latin America as well.'

So does this mean CEO Carly Fiorina is earning her \$80,000 per day pay packet?



## BACKUP COMPATIBILITY

## Linux TapeCert – updated programme

**The Linux Tape Device Certification** Program website has received a major overhaul, making it even simpler to scout out information on a potential purchase. The TapeCert program, supported by the TOLIS group, sets out to test the Linux-abilities of tape drives and puts them into three compatibility groups: Basic, Enhanced and Extended.

Unusually for a certification programme, Linux TapeCert don't

actually make a change for the service, relying instead on the largess of the TOLIS group, developers of *BRU* backup software. 'Our intent is simply to contribute back to the Linux community by providing a mechanism that ensures the compatibility of various backup hardware manufacturers' products with the Linux operating system', the company's literature claims.

[www.linuxtapecert.org](http://www.linuxtapecert.org)

## CORPORATE ADVISORS

## GNOME board announced

Richard Stallman has once again failed to garner enough votes to make it onto the GNOME Board, despite a late rally. Stallman got 52 votes, placing him 19<sup>th</sup> in a field of 23. There were 11 places being contested and, some would say

inevitably, Ximian staff make up a substantial number in the final board. A ruling which forbids more than four individuals who share corporate affiliation sitting at the same time means Sun's Bill Haneman will replace Michel Meeks.

## Jono Bacon

The founder of UK Linux, *KDE* developer and all-round nice guy, Jono Bacon is studying at Wolverhampton University.



## COMMENT

## Let's get together

“ Let's face it – we all love the Internet. We love getting online to communicate and share info – we are a community after all. While some just use the Net for browsing and email, some people like to get a little more involved.

Distributed computing is a term that ranks high on the Buzzword scale. Projects such as *seti@home* have given techies food for thought in donating their cycles to various causes. This in itself is a fantastic concept, and one that solves a number of different problems that require either (a) lots of processing power or (b) lots of cheap hardware.

One initial thought that bears thinking about is the concept of decentralising package distribution. Whenever someone uses the fantastic *apt-get* to get a package in Debian for example, the program will get the packages from a costly-to-run server. Surely the thousands of connected Debian machines could share the load?

Not only is the community evolving as a technical hotbed to develop and maintain software and services, but as technology improves, I get the distinct impression that this will fundamentally change how we provide these services given the limitations and resources we have. This is one of the heartening things about the Linux community – the will to succeed usually outweighs the challenges – examples of this include filter development, Wine and other notable projects. As the Internet gets cheaper, fatter and faster, who knows where our distributed community will spread.

Free software has shown the world what collaborative, volunteer development can create when it is combined with the dedication of the people behind it. ”



## HIGH PERFORMANCE COMPUTING MADE EASY

## Mandrake clusters

**T**aking its eye off the desktop market, just for a minute, MandrakeSoft have unveiled a new clustering technology built in partnership with the publicly funded French Agency for New Technology. The new distro is called CLIC and features rapid deployment and configuration, MPICH, LAM and PVM support and a large number of mathematical libraries. One other interesting feature is *Netjuggler*, a parallelised virtual reality 3D engine which adds clustering support to the long running *VRJuggler* application.

Mandrake CEO Jacques Le Marois said CLIC was designed to be both

affordable and easy to use. 'This project should open the door to widespread adoption of high-performance systems in businesses and scientific laboratories throughout the world,' he said.

The application made its first public appearance on a 40 node AMD cluster at France's LPM2 Physics Research Laboratory (<http://lpm2c.polycnrs-gre.fr/>). Françoise Berthoud said he had serious concerns about managing a cluster before getting his hands on CLIC. 'The CLIC installation was a great relief because in just a few hours all of the essential administration tools, and



## Mandrake support parallel computing – with Le CLIC.

a large collection of development tools, were installed on every node of

the cluster... and with amazing ease. And it works!

## OFFICE MACROS

## StarOffice your way with Java-based scripting

Sun are developing a Java-based Software Developers Kit (SDK) for *StarOffice* which should improve the suite's attractiveness to hardcore Microsoft *Office* users. The kit, based on the old *OpenOffice Developer Kit* will

enable power users to develop plugins to extend the range of the productivity suite beyond what is capable with *StarOffice Basic*, the integrated scripting language.

An alpha version of the software,

including much improved documentation, is already available on the *OpenOffice.org* website ([www.openoffice.org](http://www.openoffice.org)), but the first official release is slated to appear in 'the next minor *StarOffice* upgrade.'

According to Computerwire, Sun are also in talks with major ISPs about tie ins with *StarOffice*, perhaps involving a monthly service charge, ranging from £2-£30, for browser-based access to the software.

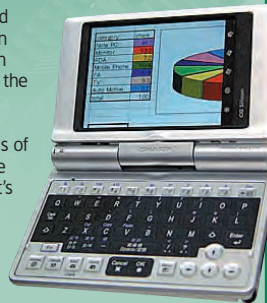
## Embedded Linux News



● Zarlink Semiconductor, based in Swindon, have announced a 'DVB-T on a chip' (left) which could see a dramatic reduction in the size of digital TV set-top boxes. The small form ZL10310 and ZL10311 systems should enable the next generation of designs to be 'slightly smaller than the average wallet.'

● Sharp's much rumoured clamshell Zaurus has been revealed and should be on sale 'soon'. A prototype of the device was shown in early November at a handheld tradeshow in Japan. Details of the exact spec were vague as we went to press, but it's almost certain that the colour VGA (640x480) screen, based on Sharp's proprietary CG Silicon display, spins and flips to cover the keyboard, like a baby tablet PC, offering the advantages of both stylus and keyboard input.

● IBM and Sharp are teaming up to promote a new version of the Zaurus PDA built on a selection of IBM software. The Enterprise Edition Zaurus, pencilled in for a mid-2003 release, will provide wireless access to IBM's range of enterprise applications including their DB2 database and *WebSphere Everyplace Connection Manager*.



## RESIST THE TAX

## A date for your diary

**23<sup>rd</sup> January 2003 is the second** annual Windows refund day, where those who object to Microsoft's license or just have no use for their pre-installed operating system attempt to get a refund from the company. This year the organisers at [www.windowsrefund.net](http://www.windowsrefund.net) are bypassing MS completely by attempting to use 'The Law' to achieve their ends. Cases are currently going through the courts of several countries around the world.

They are asking eligible protesters to visit their local courthouse to file a case in the small claims or civil court. The event coincides with the New York LinuxExpo (Jan 21-24) at the city's Javits Center.

## NEWSBYTES

■ Though not releasing full figures, American retailer **Wal-Mart** have said that their recently released \$199 Lindows- or Lycoris-based PCs have been hot sellers. A spokesman said sales had been brisk among both tech-savvy buyers and those looking for a basic email/web/word processing machine.

■ **SuSE's** Gregory Blepp has suggested that the United Linux project has been receiving interest from previously skeptical distribution vendors. Refusing to name names, Blepp told Computer Business Review that discussions were underway. "They now acknowledge what we have accomplished and are looking at it to see what can be done from there." The most high-profile refuseniks on UL's launch were Red Hat and Mandrake, though based on their previous statements, it's unlikely that either of them intend to join the United Linux family.

■ **Intel** has release a new iteration of its C++ and Fortran compilers for both Windows and Linux. The company is claiming that software compiled on their version 7 system will benefit from a 40 per cent speed boost on Intel hardware. The software would also allow developers to take advantage of the hyperthreading technology that is only now coming to desktop PC users. The C++ compiler costs \$399, while the Fortran is \$699.

■ **Linux server** shipments in 2002 have increased giving the upstart operating system just over a quarter of the market according to IDC.

■ Support charity **The Samaritans** have become the latest organisation to migrate their IT services to Linux. Citing cost, security and ease of use, The Samaritan's Information Systems Manager Mike Hermon told the BBC it was important for charities to follow the 'more for less' route that Linux exemplifies.

■ **FSMLabs** have inked a deal with India's Altosys Software Technologies which will see the latter selling and supporting RTLinux, FSMLab's real time Linux solution. In addition to India's burgeoning embedded market, the partnership will be targeting the country's manufacturing and medical sectors.

■ **Red Hat** are said to be prepping an Enterprise Workstation version of their successful Advanced Server distribution. It will be available in both 32- and 64-bit iterations, and will replace the current Advanced Workstation product line.

■ **FreeBSD** has taken a step closer to the next official release with the first candidate for v5. ISOs and FTP install files are currently available for i386, Alpha, Sparc64 and IA64 architectures. [www.freebsd.org](http://www.freebsd.org).



## PROTECTING MS OUTLOOK

## Protection for Samba

**RAV is a new anti-virus solution** from GeCAD targeting *Samba* users which is designed to 'protect against viruses and other malware regardless of the systems targeted'. While the virus threat to Linux is still minor, GeCAD hope to convince those in mixed OS environments to offload

virus checking onto a Linux server, checking files as they are opened or closed on *Samba* shares.

It is almost entirely command line driven, fluent in over 74,000 viruses and is available for x86, Sparc and s/390.

[www.ravantivirus.com/](http://www.ravantivirus.com/)

## NEW KERNEL SOON

## The joy of 2.6

**The next big release grows ever** nearer and it looks increasingly likely that the focus of 2.6 will be improvements in scalability. Linus Torvalds told eWeek the improvements would be quite noticeable in high end applications. '2.6.x will bring quite a big improvement in the ability to handle huge amounts of memory gracefully and having lots of I/O active at the same time,' he said. The feature-freeze, he said, had gone smoothly and it was time to start thinking about a date for a code freeze.

'I have a soft spot for January 5<sup>th</sup>. It will be exactly 12 years since I

got the PC that was to become the first Linux PC.'

One thing that has failed to materialise – along with the intention to jump to version 3 – is vital clustering technologies according to Oracle's Wim Coekaerts. 'While network failover and some I/O failover capabilities will be in the 2.6 kernel, disk volume management is something enterprise customers want,' he says.

If you still haven't had a flutter on exactly when 2.6 will hit the servers, visit [www.tummy.com/kernelpool/index.html](http://www.tummy.com/kernelpool/index.html) and put your money (or reputation) where your mouth is.



## Hoyt Duff

The author is one of 800 Hoyts living in the USA and runs a little fishing pier when he's not dabbling with his computers.



## COMMENT

## Why bother?

“ At a recent meeting there was ample opportunity for me to chat up the other attendees ranging in age from 18 to 80, many of whom (even the older crowd) were computer savvy. While the predominant use of computers was email and web browsing, there were several who used computers in their daily business, one even rendering 3D architectural models on topographically correct sites to check the “fit” of the building to the locale.

Without exception, Microsoft products were used. While they were intrigued by Linux during our conversations, they were not interested in it as an alternative to what they were using. In their minds, Linux failed to satisfy at least one of three criteria for making that choice: Will it save me money? Will it make me money? Will it solve my problem?

The first criteria is the easiest to deal with. Certainly Linux saves money, but not if you already “own” Windows.

Making money is nebulous at best since the OS and apps of choice are merely tools; any appropriate tool used effectively will enable the making of money if that is the goal.

Solving problems is where Linux really shines because of its overabundance of flexibility (often cited as a detriment). But when the MS software solves the problem too, the choice comes down to a matter of inertia, *i.e.*, the resistance to change.

The anti-MS sentiment seems to be the impetus for many current Linux users, but we can no longer expect this anger to drive Linux adoption and must work to satisfy one or more of those three needs before any more significant market growth occurs. ”



# FREE THINKING...

With an incredibly diverse range of speakers lined up for 2003's FOSDEM, Alain Buret caught up with a few of them on behalf of Linux Format.

## interview

### HAVOC PENNINGTON

**LINUX FORMAT** – What is your current job at Red Hat?

**HAVOC PENNINGTON** – I'm a technical lead for desktop/UI development. I spend a lot of time with *Bugzilla*, and hacking on new desktop features.

**LXF** – You are working on many projects at the same time: The GNOME Foundation, Metacity, GConf, FreeDesktop, ... What is the project you prefer working on? How are you doing to work on so many projects at the same time?

**HP** – My favorite project is probably *Metacity*; it's relaxing, at least when there isn't a flamewar about it. :-)

I definitely have too many projects right now, I'm trying to cut back.

**LXF** – There are several Desktops available for GNU/Linux: KDE, GNOME, *GNUSTep*, ... Interoperability is important (your talk will focus on it at FOSDEM), and that is why you are working hard on it. Why do you think it is so important? Are all desktop environments reacting well, and collaborating well to the elaboration of such standards?

**HP** – We need to have a single desktop platform. What I mean by that is not that all apps need to use the same libs and dev tools; but rather that an app must not be tied to only GNOME or only KDE. Apps have to work (and work well, and properly, and be integrated) regardless of the desktop UI a user is running.

If we have multiple desktop platforms, then the existence of multiple desktop projects is a problem, and will make things difficult for free software.

Multiple desktop UIs are really not an issue; it just adds to the choices users have available. Users should not have to choose between two big bundles of applications; they should be able to

choose the best apps out there. Similarly, app authors should not have to choose between two big bundles of users. So the basic goal is to break any mandatory links between apps and the desktop runtime environment, by ensuring those links are implemented via well-documented, widely-adopted specs.

Part of the reason for this is that *Mozilla's* XUL framework, *OpenOffice's* VCL framework, and *Wine* need to work well with the desktop as well. We aren't dealing only with *GTK+* and *Qt*.

Another part of the reason is that documented, stable specifications are easier to support and rely upon over the long term.

**LXF** – How many people are working on those desktop standards? How is it related to the Free Standards?

**HP** – It's hard to say; a good number of people participate on *xdg-list* and *wm-spec-list*. *freedesktop.org* is less formal than the LSB and the Free Standards Group; it's more of a discussion forum.

But once we have a *de facto* spec worked out with the major desktops, the LSB may document it as a standard.

**LXF** – If interoperability is achieved between the different Desktop Environments at the basic level, what do you think should still be done? Do you think that interoperability on the design and of the look of the GUIs is important?

**HP** – We do have basic interoperability, but there's a lot more we could do. The MIME system, the help system, these kinds of things. I do think having a similar design and look is important. In part this is fairly easy, as the major GUI frameworks are already cross-platform and support themes.

**LXF** – What are the standards that are currently defined to facilitate the interoperability on the desktop?

**HP** – We have lots of things finished or in progress. People can find a list at [www.freedesktop.org/standards](http://www.freedesktop.org/standards), including



Havoc Pennington – busy standardising the desktop.

the interaction between the window manager and the desktop, drag-and-drop, cut-and-paste, widget embedding, icon themes, file thumbnails, and more.

**LXF** – When do you estimate that interoperability could be achieved? Is the way to full interoperability still long?

**HP** – I think it's an ongoing process; we already have basic interoperability, but it can always be better, and as new features appear and become mature, we have to look at writing down the specification for them.

## interview

### BRUCE MOMJIAN

**LXF** – First, the traditional question for the FOSDEM (and Linux Format) audience. Please introduce yourself, Bruce Momjian...

**BRUCE MOMJIAN** – You can pick this up from my home page <http://candle.pha.pa.us>, particularly the Resume page. Text at bottom is:

"Bruce Momjian is a co-founder of the *PostgreSQL* Global Development Group, and has worked on *PostgreSQL* since 1996. He is the author of *PostgreSQL: Introduction and Concepts*,

published by Addison-Wesley. Bruce is employed by SRA (Tokyo, Japan) in their *PostgreSQL* support division. Previously, he was vice-president of Database Development at Great Bridge LLC, another *PostgreSQL* support company. He has spoken at many international open source conferences. Prior to his involvement with *PostgreSQL*, Bruce worked as a consultant, developing custom database applications for some of the world's largest law firms. Prior to this, he was a high school computer science teacher and holds a Masters in Education."

**LXF** – Can you give a little history about your *PostgreSQL* history? I mean, you did not wake up a morning and said: "I'll be a *PostgreSQL* core developer"?

**BM** – In 1996, I had been using SQL databases for 7 years, and wanted an SQL database on my Unix machine. I tried *mSQL*, but that didn't have the same features I was used to in commercial databases. *Postgres95* had just recently been released as *Berkeley Postgres* with an SQL interface. I tried that and it worked pretty well, but there were lots of bugs, and there wasn't a group collecting

## interview

## MICHAEL MEEKS

**LXF – For those who couldn't attend to FOSDEM last year, please present yourself**

**MICHAEL MEEKS** – I'm Michael Meeks, I have the privilege of working for Ximian, where I'm paid to work on GNOME (and other interesting Desktop related projects), I'm English, 25, 1m tall, hairy all over, and like bananas.

**LXF – What did you work on since last year's FOSDEM ?**

**MM** – Lots of disparate things; auditing *ORBit2*, fixing *Nautilus* bugs for GNOME 2.0, performance and efficiency improvements all over the place, and more recently playing with *OpenOffice.org* (*OO.o*)

**LXF – What is your main job at Ximian ?**

**MM** – Research and development of interesting bits of code I suppose. At the moment that means *OpenOffice.org*, maintaining the bits of GNOME I'm interested in in parallel, and poking my nose into other people's work on *Ximian Desktop 2* (*XD2*).

**LXF – Many applications are already ported to GNOME 2, but do you think that most people already switched to GNOME 2.**

**MM** – Well – of course, since our platforms are parallel developable, some people are using GNOME 2.0 for it's improved speed, stability and

beauty, and hacking on GNOME 1.4 stuff on the same prefix. Then again many of our users are not hackers, but workers, so they went straight to 2.0, with many people from non-GNOME systems. Other people are just testing the water with a toe and waiting for *XD2* to make it an even more beautiful, integrated experience before diving in.

**LXF – GNOME 2 has been released a few months ago, and GNOME 2.2 will be released soon, what are the major improvements of GNOME 2.2 compared to GNOME 2?**

**MM** – There are lots – but most people have been concentrating on producing the new GNOME for release, rather than adding new features. Thus to a large extent GNOME 2.2 is a great yardstick for purely time based, six-monthly releases. Having said that, there are plenty of nice new things in *Nautilus*, and some great underlying infrastructural improvements, and additions to the platform.

**LXF – What are the next steps to bring a full desktop to the Unix community? What do you think is still missing?**

**MM** – Integration, coherency, standardisation, lockdown, management, ... lots of interesting things. Ultimately I think GNOME 2

provides an excellent environment to work in now – we have to focus on ensuring every application integrates with the others intelligently, that we have a coherent look and feel etc. (I applaud the Red Hat initiatives there). As for what is missing – watch this space.

**LXF – People often complains that GNU/Linux is not ready yet for the desktop among others due to the lack of support of multimedia applications. Do you think they are wrong?**

**MM** – Yes – I'm fairly convinced they're wrong; but then I judge what other people need by my own workflow where *Sibelius* via *XMMS* is idyllic luxury while I type. That coupled with the fact that I think projects like *GStreamer* are beginning to provide GNOME with a powerful multi-media platform. I'm not convinced GNOME is (or ever will be) ready for the Home / Personal desktop market in terms of the functionality that people expect today.

**LXF – OpenOffice.org has been chosen as the Office suite of the GNOME desktop. Does it mean it will be better integrated to the GNOME desktop in the future releases (GTK integration, Nautilus integration, ...)**

**MM** – Well – of course, if your GNOME distribution is correctly setup, *Nautilus* integrates nicely with *OO.o* in terms of



**Michael Meeks – GNOME hacker for Ximian, fixing Nautilus and taking an interest in XD2.**

point and click document editing. *OO.o*'s GUI toolkit has the merit of acute simplicity and ease of portability, it has lots of downsides in layout and RAD terms – I'm not that convinced that people should care what widget toolkit is used under the hood personally. As for future *OO.o* releases – I'm sure that they will integrate far more closely with GNOME.

**LXF – Last year, the room where you made you talk was full ... so what do you expect this year ?**

**MM** – A bigger room – there wasn't much space in that broom cupboard last year :-)

I'm hoping that lots of hackers, dripping creativity will come and be inspired to hack on GNOME and/or *OpenOffice.org*.

them to generate new releases. Merc Fournier and I got that group together, and that's how *PostgreSQL* started.

**LXF – You're not the only developer and there are plenty things to do around PostgreSQL. Can you present us the other people and aspects of this project, like documentation, ...**

**BM** – Well, we are very distributed, in the sense we don't even have predefined roles for people. We have a core committee, but they are mostly there for conflict resolution. All our work is done in the open, on mailing

lists, where people ask questions, submit bugs, and post patches.

We do have a webpage listing some of the major contributors: <http://developer.postgresql.org/bios.php>.

**LXF – How does the databases world evolve ? What are the main improvements made these five years in general, and particularly for PostgreSQL?**

**BM** – Well, that's a big question. I would say our first few years involved fixing server crashes and fixing things that clearly were broken.

In more recent years, we have focused on improving functionality, standards compliance, and performance.

**LXF – You are the vice-president of Great Bridge, a PostgreSQL support company : what kind of clients have you ? Have you been hit by the "dot-com dot-gone" crash ?**

**BM** – Well, Great Bridge is already dot-gone. I work for SRA/Japan now. See my biography listed above. SRA/Japan supports *PostgreSQL* users in Japan. We have very good saturation in that country.

**LXF – PostgreSQL is always compared to MySQL.**

**I will not ask you which one is the best, instead can you tell me what are the main differences between them ?**

**BM** – In a few words, *PostgreSQL* is more like *Oracle*, and *MySQL* is more like *Microsoft Access* – at least that's how it was put by several other people. *PostgreSQL* focuses on standards compliance and on the needs of enterprise sites, while *MySQL* focuses on performance and is more concentrated on web-based applications. [LXF](#)



Mailserver

# Mailserver

Share your opinions, right wrongs and demand justice by writing to *Linux Format*. Drop us a line at: **Linux Format**, Future Publishing, 30 Monmouth Street, Bath BA1 2BW or email: [lxformat@futurenet.co.uk](mailto:lxformat@futurenet.co.uk)

## FlightGear?

I have had three copies of the *Linux Format* magazine since I placed a regular order with my local newsagent. I was quite interested in the self contained Linux setup on the DVD. The, Knoppix Distro ran well.

Being a flight sim fanatic since the F55 days (I wrote several airports for Western Australia, which will work in the latest versions) I was most interested in the "Full playable version without install, of *FlightGear*," having looked at this program in its infancy I was not really impressed. So, I went into the shell command and, after ploughing through umpteen directories I tracked down the exec file (It was certainly well buried in the system) and coming across many "files not found" and with a pounding heart hit the final enter



button. (incidentally the command **sh FlightGear.sh** does not work, in the directory the file is written as *Flightgear.sh* I was led to believe that when writing files in Linux one normally kept them all in lower

case except in certain circumstances, let alone throwing caps in the middle of filenames. I eventually found the correct combination by trying various formats of the filename.

I was not disappointed...I got what I expected would happen ..... "Unable to execute program, missing libglut.so.3". Welcome to Linux. So far I have tried virtually all the various formats of Linux: Red Hat 8.0, Lycoris Amethyst, Mandrake, and Debian Woody.

I digress. The LXF DVD was run on a Pentium 1.1GHz with a Gforce 444 MMX card and 512MB RAM. So far I have not been able to install any of the major installations from the LXF magazine DVD's as they all appear to be missing many files and libs - OK, so this is Linux and there is no such thing as install and run. It's designed to use all your monthly allocation online time searching for missing files. Whilst I enjoy cruising the 'Net I would rather spend the majority of my time running and experimenting with programs, etc.

## ★ Letter of the month

This month's winner receives a copy of Sun's StarOffice

### HTML email

The article finishes with the question, "can anyone think of a good case for emails that should be sent in HTML?" For me, it's simple, we live in a colour world, well at least I do. How many ASCII supporters still use monochrome monitors - or have they all moved to colour displays? How many of them still use *Gopher* or *Lynx* to display web content.

At the end of the day whether it be email, webpages, etc., it's all about information - when you use colour the message is emphasised. Plain text, unless with plenty of white space doesn't have the same impact.

Whether HTML is the correct

medium for maximising the content of emails, I don't know, but is something we all have. If you chose to use mail clients that don't support HTML, that's your right. I chose to use mail clients that support HTML as I live in a COLOUR world.

I suppose email is also about personal expression, and for some HTML delivered email achieves this and for others plain text. As technology and bandwidth improves, we no doubt will receive even more content rich emails whether we like it or not this is called "progress". The whole reason the WWW took off is because of HTML, if colour and content rich information wasn't important we would all be using *Lynx* and *Gopher*

or even simple browser based technologies.

On another matter, keep up the good work, I always eagerly check the newsagents for the latest copy of LXF. We are usually about two months behind in receiving the DVD version downunder. An excellent mag, that covers Linux objectively and keeps me informed even if we do get it two months later.

Paris, Australia

Thanks for putting your views across. I suppose that for some people such things are important. Looking through my inbox, which currently has about 450 mails in it, I can't see one that to my mind would be improved by being in colour or including different fonts, styles and graphics. But I guess it depends what information



we are trying to communicate.

The problem is though that at least 1/3 of the mails in my inbox are HTML. Not only do they take up more space, they very often include links to all sorts of graphics and whatnot that take time to load, and probably evily register my mail address for spam.

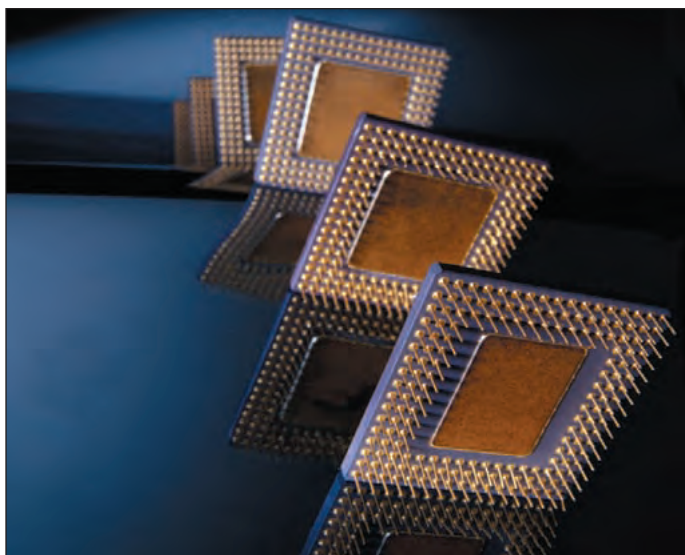
The original debate was about whether mail clients that didn't recognise HTML should be classed as 'broken'. Any more opinions?

In the meantime, I hope you enjoy your copy of *StarOffice*, this month's prize for our star Letter. *StarOffice* is currently available for £45.99 from [amazon.co.uk](http://amazon.co.uk) and other retailers.



I think that at the present stage Linux still appears to be in a wilderness of confusion and inconsistency. Debian appears to be the most complete archive available, but Linux has a long way to go if it is to be an alternative to MS. (Some will say its not supposed to be an alternative for the general user.) I doubt if there will ever be a "complete" archive available into which one can run other programs because of the massive number of "Cooks in the Kitchen" and we all know what that causes to be done to the broth! I may return in five years time to see how the system is going. (If I am still living, maybe I am getting too old for all this – LOL)

Meanwhile I will hastily despatch this before a warning sign comes up on my screen, "The application you are running has performed an illegal function and your system will now be shut down." Oh thanks William ... Incidentally the latest "bug" in XP seems to be that the DUN/Network settings for no reason vanish and cannot be restored. Happened to me and a



Don't forget the **nopentium** option for booting an Athlon chip!

from any obvious differences in performance, how significant are compatibility differences, and are there any specific application areas where one type of processor is likely to offer a significant advantage?

Keep up the excellent work – I always look forward to LXF hitting the doormat, and the very

manufacturers introduce optimised feature sets for processors to extend it's feature vocabulary (such as MMX and AMD's own 3DNow!), but for the most part, in terms of instructions, the chips are largely the same.

There is no prerequisite for Linux to run only on Intel chips, it performs just as well on AMD technology. One

proviso is that modern Athlon chips may cause problems if you don't include the **nopentium** option to the kernel, or compile a custom kernel for them on distros with optimised kernels (e.g. Mandrake).

## Simply Linux

This week I attempted to buy a new PC for my boys at 'Simply' recently. They sell cheap PCs specified by customers. I said I didn't want any software because I was going to install Linux on it. The salesman said that he couldn't sell me a PC without MS Windows on it, because to do so would be illegal! I laughed out loud. I said that this was ridiculous, and explained that Microsoft don't make the law in this country, but he was insistent that he couldn't sell me a PC.

So, I asked to speak to his manager. A manager was duly sought and said the same to me as his salesman has said: "It would be illegal for us to sell you a PC without MS Windows on it". I asked the manager whether he cared that he was about to lose the sale of a



**"The salesman said that he couldn't sell me a PC without Windows, as it would be illegal! I explained that Microsoft don't make the law in the UK"**

local friend over the weekend..

David L Smithdale, *Western Australia*  
Sorry if you had problems with *FlightGear*. From your brief explanation, it seems the problem is that you don't have any 3D support installed for your graphics card. The reason for this is that nVIDIA produce closed source drivers for their hardware which aren't included in many distributions – you will have to download and install the driver from nVIDIA themselves. Don't tell me you've never had to do something like that on Windows!

## nopentium

Further to Trevor Cushen's Mailserver query in September issue 'Lintel only', I'd be grateful if you can clarify just how important it is to use an Intel processor rather than an AMD for example? Apart

interesting features in *Linux Pro* too.

Just a suggestion for a future article – could you do a beginners' guide to updating different distros. Like some other Linux users I've got my system (Red Hat 7.2) working well but I'm scared to try updating it in case I lose some critical configuration or other. I know it's really important from a security point of view, but keeping my system going is even more important!

I've heard a bit about Debian *APT* being especially good, but how do all the various update methods used by the different distros compare, including cost. Should I use *APT* on my RH7.2 for example? Mike Hudson *via email*

Thanks for the kind words. From an instruction set point of view, the AMD chips and Intel's own processors are very similar. Occasionally chip





# Mailserver

« PC, but he didn't seem bothered.

A colleague suggested writing to the office of fair trading. Does this horrify you too?

Please publish adverts for PC-makers who will supply software-less Linux-compatible PCs, or PCs with Linux already installed.

David Legg, *via email*

We're grateful to you for sharing your experience. A quick google search will bring up a few companies in the UK who will sell you a computer without Windows. It may interest you to know that even 'mainstream' companies are selling Linux desktop PCs now – check out Evesham!

## Libraries

I wonder if it is possible to include all required libs for the apps you so generously include on your CDs/DVD? I know, I can always download them from the Web somewhere, but with an old-fashioned modem

connection I sort of rely on your CDs to keep me up to date and provide me with interesting new apps. Also, sitting down happily to install a new interesting app from your CDs/DVD only to find out I need a number of libs that I don't have, is like unwrapping the Xmas toy so hotly desired only to find that batteries were not included...

Olof Liungman, *Sweden*

This subject often comes up, and for the full answer, I would refer you back to the letters pages of almost every issue of the magazine. This is not a problem that can be easily solved by putting on 'all' the libraries. Because of interdependencies, we'd end up including a complete new distro every time, just to support one application. Common libraries are kept updated and can be found in the Essentials directory of the disk, but we can't include every dependency for every program.



Putting every library on the coverdiscs would result in a complete distro.

## Libraries

I would like to suggest a topic for your programming tutorials.

Anyone who does any C or C++ programming quickly learns how to use libraries. However I haven't found much information on how to create libraries. It would be useful to be able to create shared or static libraries of C++ classes that I have developed to make reuse easier.

KDevelop supports the creation of many types of application but not of libraries. KDE however

comes with many libraries. How are these developed?

David Manley, *via email*

KDevelop doesn't explicitly support the creation of libraries only because it doesn't have a 'project profile' for them, but I'm pretty sure you could use KDevelop to create one.

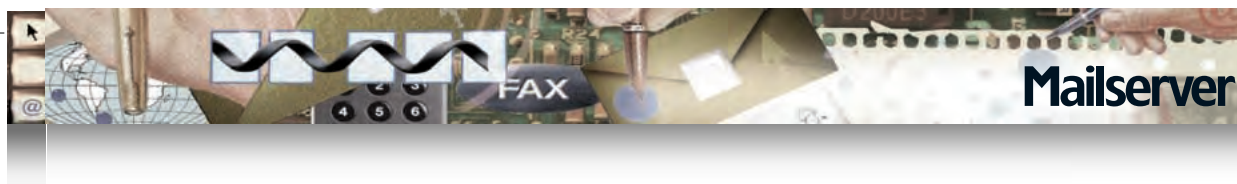
Essentially, a library is just the same as normal code, except it only contains functions and lacks a `main()` structure. To be used as a standard Linux shared object library, it needs to be compiled with the gcc options `-fPIC` (to make the code non-position dependent)

**“Jigdo currently only works on Linux, Windows and Solaris. If anyone has a better way of enabling people to burn the images and boot the DVD – tell us”**

# Helpdex

shane\_collinge@yahoo.com





and **-shared** (to produce the shared object file), which can presumably be invoked by changing the compiler flags in *KDevelop*.

## Mandrake thanks

Thank you for the Mandrake 9.0 with your Christmas edition, in spite of earlier experiences I tried it and I'm (moderately) happy.

First the good news. For the very first time on any machine I've actually got a desktop that runs (as opposed to crawling) and doesn't fall over at the first mouse click. Hell, this is actually up to the standard of Windows 3.11! (Red Hat 7.2 crawls on a 1600MHz Pentium 4 but this one does OK on a 200MHz P2. Not being a kernel maven I haven't the foggiest idea why.)

Now for the unkind comments. Do the guys at Mandrake know that when you do shutdown the sequence "ends" (actually it doesn't, in under an hour) with an infinite stream of gibberish? Cute.

Also it would be nice to be able to access a CDRom, because *Nautilus* deems all files and directories on CDRoms to be strange and terrible and blanks them from the listing if you click on them. This includes those coverdiscs too.

Incidentally, when I went into a rootmode console and manually mounted the CDRom there was no problem accessing it – except the desktop no longer showed the icon.

Finally, for now, where's *The GIMP*? It doesn't seem to be present, nor does there seem to be any option to load it. (Unless the installer assumes that only Web designers need it.)

So I'll award points for improvement (Mandrake 7.0 tried to fry my monitor) but I'm still not going to recommend this to my friends to replace Windows 98, yet. Tom Groves, *Ashford, Kent*

There are a few 'issues' with Mandrake 9 (as referenced in our review last issue), which as you'll appreciate is a fairly complex piece of software! The CD problem is almost certainly to do with *supermount*. The easiest things here is to remove the *supermount* reference. You can do this manually, or use the Mandrake control centre and just uncheck the 'Supermount' option.

The shutting down problem is usually related to certain BIOSes and power management features. The



**Mandrake 9.0 – an improvement, but supermount is a gotcha.**

good news is, it's safe to turn off the power to the box after the disks have been unmounted.

## IDEs

I thoroughly enjoyed Maurice Kelly's C++ IDE review in December's *LXF*. However, three significant IDEs were not covered in the review which I would have liked to get a "second opinion" about, namely IBM *et al.*'s *Eclipse* project (using the CDT environment), Sun's *NetBeans* project (with the C++ module) and, although mentioned in passing, Metroworks *CodeWarrior*. I realise that it's impossible to cover all existing IDEs in a single article but these three are notably more significant than *QIDE*.

I have used *Eclipse* recently and found it to be a relatively solid IDE. Being open source and supported by some major players, it would have been a natural fit for the article. *NetBeans* is fantastic for Java development but still nascent for C++ development. However, it's CVS integration is unmatched by anything I have used to date. As for *CodeWarrior*, I cannot bring myself to part with \$149 for a product that I might end up never using, but I would have liked to know if others feel it is \$149 well spent.

That said, I strongly agree with Maurice that *KDevelop* is the strongest in terms of features and usability. The most important aspects of an IDE for myself personally are revision control system integration, integration of build/debug, code highlighting, and polish of the editor in decreasing order of significance.

Marc Lavergne, *Florida*

Glad you found the roundup useful. The *NetBeans* C++ support is still very much a work in progress so we didn't

feel there was much point in including it here. *Eclipse* would have been more relevant. We have covered it in *LinuxPro*, and I'm sure we'll be revisiting it soon.

## Jig-don't

Oh joy. You've deemed the Debian experiment with *jigdo* a success. So much so that you've repeated it with Mandrake. Well I can't get it to work. I really want to try the Debian setup and . . . wait, you say that *jigdo* on the latest DVD is now available to all users. Hurray!

So, grab the box with the DVD player in it, boot up Mac OS X, grab the DVD and – hang on though – you've got Linux and Windows versions. Nothing for me to easily create my CDs on my Mac box. Sigh. Back to the floppy disks and a trying a network install ...

Rod Furey, *The Netherlands*

Sorry for the misinformation. *Jigdo* currently only works on Windows, Linux and Solaris. If there is a better way of enabling people to burn the images to disc, and boot from the DVD though, we're willing to try it, so please let us have your ideas. [LXF](mailto:lx@futurenet.co.uk)

## SUBMISSION ADVICE

### WHAT WE WANT:

- Letters about the magazine or Linux in general
- Constructive criticism
- Your opinions
- Concise points about relevant subjects

### WHAT WE DON'T WANT:

- Technical question – direct those to our Q&A pages!
- Random abuse
- Nonsense rants
- 200 pages of meandering diatribe

### WRITE TO US AT:

**Linux Format**, Future Publishing, 30 Monmouth Street, Bath BA1 2BW or email: [lx@futurenet.co.uk](mailto:lx@futurenet.co.uk)



**Developers Solution Provider**

## Reseller Package

Ideal for Web Designers, Web Developers and Entrepreneurs

5GB Fast UK Web Space,  
Each VS £20 p.a.  
10 Virtual Servers included.  
Expert telephone & email technical support, regular tape backups, 24/7 server monitoring, same day setup.

**Assigned Technical Manager**  
**99.7% uptime guarantee**

### Each Virtual Server includes:

Supreme Control Panel,  
Unlimited Mail Boxes,  
Announcement Lists & autoresponders, secure server,  
3GB Bandwidth per month,  
Graphical Statistics, Real Audio & Video, Password Protected Directories, Spam Filtering, 24/7 FTP & SSH Telnet access.

**Powered by Zeus Web Server**

**Advanced features include:**  
CGI, ASP, PHP4, JSP, Java Servlets, ODBC, MySQL Databases.

**£45 per month £450 p.a**

Hosting your web site with CWCS lets you benefit from our state of the art data centre, speedy network, expert technical support and friendly customer service.

**Our Services are your Business Solution**

tel 0870 703 1000

[info@cwcs.co.uk](mailto:info@cwcs.co.uk)

[www.cwcs.co.uk](http://www.cwcs.co.uk)



prices exclude VAT.



# Reviews

All the latest software and hardware reviewed and rated by our experts

## LXF verdict explained

Each review is accompanied by a Linux Format Verdict to help you to assess the product at a glance (it's no substitute for actually reading the review, though). We award scores out of ten in the following categories:

**Features:** Does it provide the functions you need? Is it innovative?

**Performance:** How well does it do its job? Is it fast and reliable?

**Ease-of-use:** Is the interface well designed? Is the documentation well written, helpful?

**Value for money:** Does it have a competitive price?

For those who like numbers, the Linux Format Rating is a score out of 10 summing up the overall excellence of a product. It will usually, but need not be, an average of the above categories. We award scores as follows:



**10** The close to perfect product.



**8-9** Good, but has a few niggles.



**6-7** Does the job, but needs work.



**5-4** Average.



**1-3** An utter disaster. Back to the drawing board.

## The Top Stuff Award

If we really, really like something — we really think that a particular piece of software, hardware or any other sort of ware is the best stuff around — then we'll give it our Top Stuff Award. Only the very best will be chosen. It's not guaranteed to all products that score highly.



## THIS MONTH...

### Xandros

Debian-based desktop distro with *Crossover plugin* and *Office* **p20**

### Realsoft 3D

Powerful modelling and rendering software available under generous licence **p22**

### Cubit >>

Strikingly good-looking in a world of beige boxes, but we feel you'll also be interested in its size and low power consumption **p26**

### Kohan AG

Thanks to WineX we get to play the Fantasy role-player prequel **p28**

### Webmin

Version 1.0 of the ubiquitous remote admin tool **p30**

### PHP Encoders

ioncube PHP Encoder and Zend Safeguard Suite compared **p34**

### Books

Linux in the work place and two rather different hacker biographies **p38**



## COMING UP SOON...

### Arkeia 5.0

New version of the popular backup solution — includes a completely redesigned interface

### ARCServe

New kid on the backup block, from Computer Associates

### Jool's Kwartz Server

Ready to go webserver in a shuttle-size box. Includes radical, new admin tools

### Perl bookshelf

Six of the best Perl reference books in a drinks-mat size package

### United Linux

Teamwork brings forth the first full release of this enterprise class distro

### UT 2003

Receiving many votes in our awards poll, but some of you may care to read the review first

## DESKTOP DISTRO

# Xandros Desktop 1.0

**Jono Bacon** appraises a relative newcomer to the desktop arena.

**Easy to install distro, which integrates well into a Windows-dominated World. Compare Lycoris and Lindows.**

- **DEVELOPER** Xandros
- **WEB** [www.xandros.com](http://www.xandros.com)
- **PRICE** \$99

**X**andros is a new brand to many people, but not a new distribution. Xandros has licensed the Corel Linux distribution from a few years back and breathed new life into it with new features, a new brand and new concepts.

Given the rebirth of this Corel based distribution, many have taken a keen interest in seeing just what this new Xandros Linux is all about, and I was no different.

## Installation

The first element to test in reviewing any Operating System is how easy it is to install; this is particularly important for an OS such as Xandros which is aimed at new users with little or no computer knowledge.

Xandros like many distros is bootable from CD, so I changed my BIOS to boot from CD, popped the disc in and rebooted my machine. After a minute a screen with an animated Xandros logo popped up

and started booting the installer. After a few more minutes I was presented with a pretty graphical installer.

Installation can take two routes – express or custom. As I was mainly evaluating this distro for users with little knowledge of computers, I first tested the Express routine. The installer asked me a few questions for setting the root and user accounts up, and then started installing packages. After 20 minutes and five clicks Xandros was installed and it instructed me to reboot. This installation was impressive in it only asked a minimal amount of questions which were simple to understand and answer. At this point I was quite skeptical if any of it would work, but luckily it all worked well.

After rebooting, Xandros presented me with a graphical boot selection screen (it also detected my other OSs and added them to the menu); I pressed **Enter** and Xandros booted and presented me with a graphical login screen. I found this quite ingenious as it asked no questions about XFree86 configuration and it detected both my network cards and got my cable modem online at boot with no problems whatsoever. The ease of installation should not be underestimated. I did install Xandros using the Custom routine and it offered the usual array of settings,

but the Express routine was unparalleled for ease of use and other distros should take lessons from Xandros.

## Getting to work

OK, so Xandros was installed, now it was time to use it. I logged into the machine with the user account I added in the installation, and it booted me into a modified KDE environment.

Being part of the KDE effort, I have usually taken quite a dim view of modified KDE installations, but I found that the effort Xandros made was well structured and easy to use. KDE had been modified mainly in the panel and *Control Centre* and the additions to the environment add value in the user interface while still preserving the concepts that make KDE what it is. Not only has KDE been modified with additional wizards, interface tweaks etc., but Xandros comes with a rather impressive collection of applications.



The default desktop presented when you log into Xandros.



Xandros's tools, and the *Crossover* apps, make this distro worth the price.



The choice of software in Xandros is overall well made. *OpenOffice.org* was good to see, as was the popular *Mozilla*. The usual array of utilities was present, and also a decent array of text editors is included. One unfortunate absence was *The GIMP*. It seemed to me that many people would find *The GIMP* useful in conjunction with *OpenOffice.org*, and although it can be downloaded using the *Xandros Networks* application, it was a shame it was not part of the normal installation. Despite this minor setback, I did find myself not needing to download the usual array of additions post installation, and the base Xandros install is fine for web browsing, office productivity, email, instant messaging and other typical desktop tasks.

## System config

Configuring Xandros is in general a simple process. Common things such as adding shortcuts, adding printers and setting up the environment are simple with wizards added to handle most situations. It was nice to see that Xandros have catered for networks well with configuration panels for setting up Windows/NFS shares, Internet connections and other requirements. Although the purists will still use command line tools for many of these processes, it is positive to see easy to use wizards that handle these issues. Something that should be noted also is that the wizards are actually easy to use and do indeed work; claims that cannot be said for some other Linux distributions.

It seems that Xandros have been hellbent on trying to provide a GUI

## The Magnet test

### Cherchez la femme

**The ultimate test to run Xandros through was to sit a one-woman computer destroying machine in front of it. This woman affectionately known as The Magnet is my girlfriend, and this is how she got on...**

**Susan first sat down and installed Xandros. The Express route was chosen and she found the questions simple to understand and answer. She booted into the machine and found the modified KDE interface easy to use. Susan welcomed the My Linux and My Documents directories, and the use of**

**the automounter was much cheered after the hell she went through accessing the floppy drive on my Debian machine. She used OpenOffice.org for the first time and wrote her research project in it and found Xandros "much better" for the things she needed to do (which also included gossiping online with Instant Messaging).**

**All in all the Magnet test proved successful, and it gives Xandros the thumbs up from both a resident computer shredder, and from a freelancing cynic.**

interface for pretty much most things, and the good news is that it works well. For the first time in a Linux distribution review, I did not immediately load up a command line terminal as soon as I was logged into the GUI. I found that for most things I needed to do in Xandros, there was a graphical wizard or front-end that would work well for the task. Much of this is down to Xandros, and much of it is down to the Open Source applications that are included, but Xandros deserve the credit for smoothly interfacing the separate pieces together. The feeling of confidence with Xandros is quite satisfying, and the ominous feeling that something will break with the GUI is rarely encountered.

## Additional features

Like with any distribution, you often get vendor written applications that are included to help you use the system. Xandros have included some very

useful applications that add real value to the OS. The first is *XFM*, the *Xandros File Manager*. The file manager works in a similar way to *Windows Explorer* or *XTree Gold* from the old DOS days. *XFM* works well at presenting a simple access point to your system and devices such as floppy and CDROM drives, which are all automatically mountable within the file manager. Xandros have presented the system as My Linux and you also have some directories created for you such as My Documents and My Pictures which can help new users store their files easier. Although these are only minor changes, they make the process of learning the system easier for new users.

There are another two main apps included with Xandros called *Xandros Networks* and *Xandros Update*. *Xandros Networks* is touted as a web-based software catalogue, based on the Xandros network, to provide a simple front-end for downloading updates and new software. It was clearly stated that this application was a pre-release and hence may have some bugs, but I found it quite stable. When the user clicks on an application, it is downloaded and installed automatically with dependencies. This works very well, and future updates to the service should prove interesting. The second application, *Xandros Update*, is a front end to installing packages from the Xandros archive. The underlying package manager is the popular tool *APT* (Xandros itself is Debian based) and this application is merely an interface to *APT* – you can also use *RPMs*, *Tarballs*, *Debs*, and a Debian *apt* source as a mirror to download your software from. The only thing that was a little confusing was how to

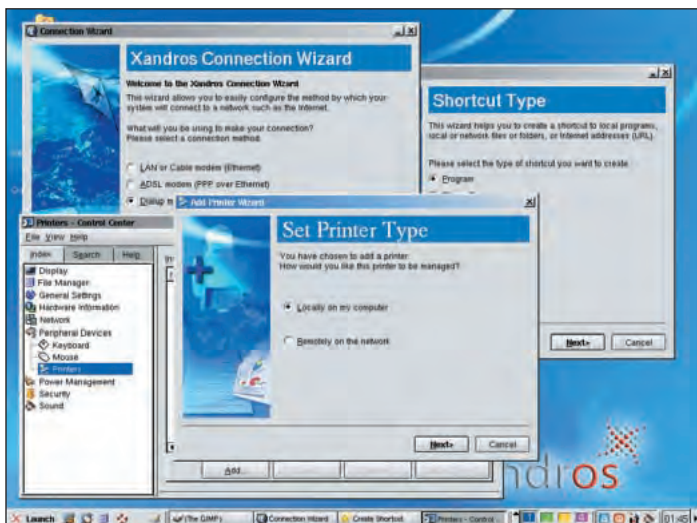
differentiate between the two applications, and it may be wise in the future for Xandros to consolidate them both into one application.

As well as these custom applications, Xandros have also included the *Crossover Plugin* and *Office* products (these products alone justify the price tag of Xandros Desktop). The inclusion of these products has been a wise move and lets the distro interface with Windows apps much more easily than other distros. Xandros has made the decision to include these products for you, and it seems a wise move. I tested both of the *Crossover* additions out, and both worked extremely well – particularly the *Crossover Plugins* product which allowed *Realplayer* and *Shockwave* to work well within the web browsers on the system, as well as other plugins.

## Summary

Xandros is a great, great distribution. Although you may think I am on commission for writing such a glowing review, I am not, but I am simply very impressed at what they have managed to do. Xandros has successfully managed to take a solid base distribution (Debian) and make it simple to install and use for the masses. I am usually quite despondent about shelling out the £££ for a Linux distribution, but I would be happy to pay for Xandros Desktop – they have managed to implement a feeling that you are getting real value for money with the product.

Xandros have set a standard with this distribution. If you want an easy to install, configure and use Linux distro and don't mind paying for it, I highly recommend this product. If Xandros put the marketing budget behind their distribution and ensure that they keep the updates frequent and high quality, I foresee great things for them. [LXF](#)



Wizards are provided to set up printers, shortcuts and more.

## VERDICT

Ease of use	9/10
Features	8/10
Performance	7/10
Value for money	7/10

Xandros is a well designed, built and implemented distribution. If you need ease of use and the power of Debian, it is highly recommended.

**LINUX FORMAT RATING**  
**9/10**

## 3D MODELLING

# Realsoft3D v4.5

**Nick Veitch** gets his polygons in a twist and discovers that he can talk a lot about metaballs.

**Realsoft3D** is a 3D renderer/modeller. It compares to software like *Maya*, *Lightwave* and others.

- **DEVELOPERS** Realsoft Graphics Oy
- **PRICE** €300
- **WEB** [www.realsoft3d.com](http://www.realsoft3d.com)

**R**ealsoft3D has quite a history. Like a lot of 3D software, it began its life, many seasons ago, on the Amiga platform, once the de facto 3D platform until it got left behind in the processor power stakes. The original *Real3D* was a little quirky, but nevertheless had some powerful features and a refreshingly easy to use interface. *Realsoft3D* is a completely different product, but it's easy to see how those years of experience have contributed to this product.

Unlike, for example, a spreadsheet, there is no universally accepted paradigm for 3D software, mainly because of the different combinations of tools and the ways they can be

used. Generally this means that any 3D software involves a significant learning curve, even for those experienced with other systems. Realsoft have made things significantly easier with their multipanelled interface which is easy to navigate, and can be completely customised when you are more familiar with the tools.

## Modelling

Render engines for Linux aren't as scarce as good modellers. *Blender*, *POVray* and commercial solutions like *Maya* (and Newtek's *Screamernet* is soon to be available for Linux) produce good results, but modellers are another matter. *Realsoft3D* has a full range of modelling methods, including nowadays standard features such as metaballs, NURBs, boolean functions, fractal subdivides, meshes and pretty much every common modelling tool. The functions available in the toolbar react to selected objects, editing modes (you can easily switch between points, edges, faces and complete objects)



reorienting the view. You can customise the interface yourself, or a number of tweaked environments are available from the menu.

The crucial thing is that the modelling system didn't take that long to get used to. The user interface is easy to navigate and there are plenty of excellent tutorials in the printed or online manual, as well as dozens of useful examples. The work put into the

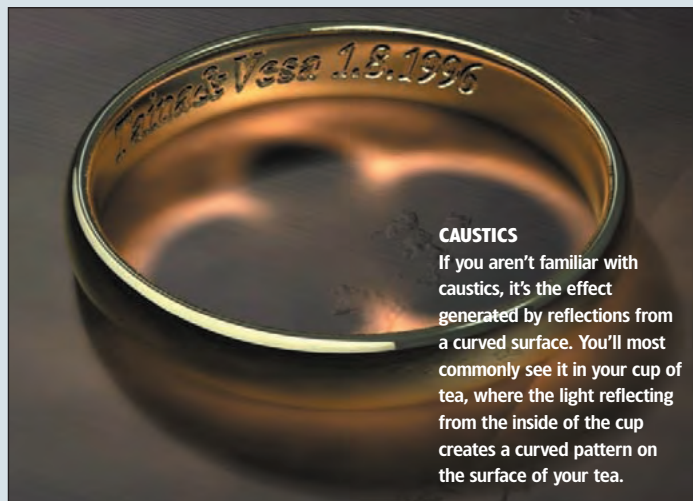
and which modelling system you are currently using.

Only one viewpane is displayed by default with a side bar controlling the orientation of the view and various tools for panning, zooming and

## Cool effects and features

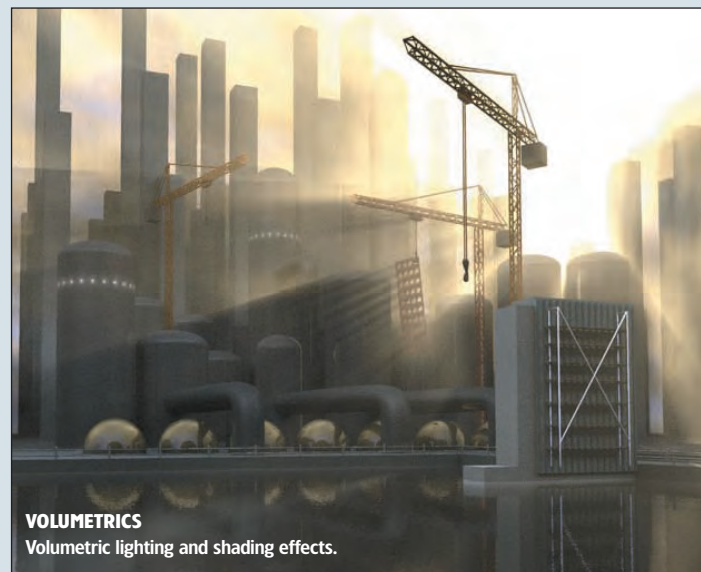
Advanced rendering demonstrated.

The best way to highlight some of the advanced rendering techniques and effects in *Realsoft3D* is to show you. Here are some of our own, and some of the demo images.



### CAUSTICS

If you aren't familiar with caustics, it's the effect generated by reflections from a curved surface. You'll most commonly see it in your cup of tea, where the light reflecting from the inside of the cup creates a curved pattern on the surface of your tea.



### VOLUMETRICS

Volumetric lighting and shading effects.

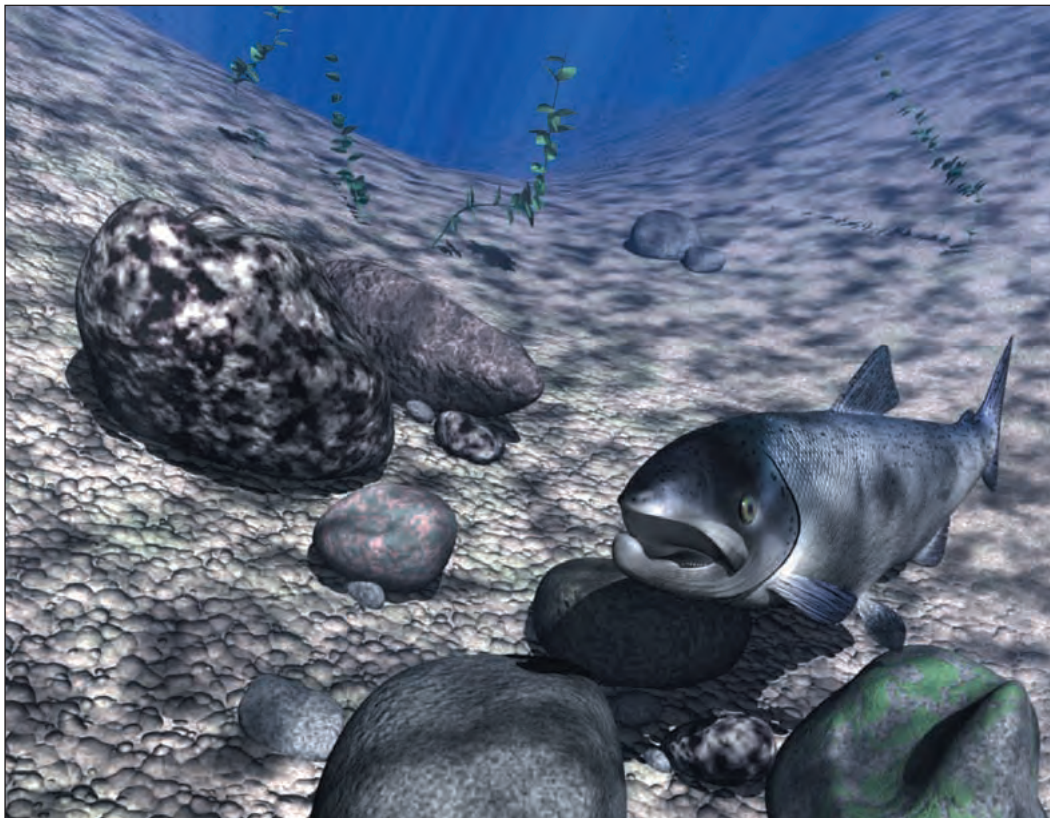


documentation has certainly paid off in a slightly gentler learning curve.

## Animation

With path animation, morphing, particles, skeletal systems, inverse and forward kinematics, muscles and muscle morphing, there is a whole range of tools here for almost any animation project – it's just a case of choosing the right one for the job. Many of the more cunning ideas come from *Realsoft3D*'s 'Choreography' features. The basic concept behind this is that objects can be set to react depending on a number of attributes, rather than just keyframing them. Some of the examples given help to illustrate the power of this technique – a sphere that deforms according to acceleration, a face that morphs according to the amount of light that falls on it. Rather than micromanaging the scene, this leaves the animator free to define the effects that need to be achieved and then concentrating on the global picture. The muscle morphing feature is a good example of this. Set up an object to deform when a skeletal limb is flexed and you have muscles. Now to animate the object, you merely have to decide when the limb is bent, the rest takes care of itself.

To help you there is a complete physics system included, which can handle the usual gravity effects, magnetism plus some fluid effects too. Collision detection is also included, and



**Sadly current print technology can't show you how fluidly this fish swims thanks to Choreography...**

some of the demo scenes show how this can be used to good effect – a wall exploding when hit by a heavy object, or some skittles bouncing out of the way as they are hit by a bowling ball.

## Post processing

It's now pretty much the norm for 3D software to include its own

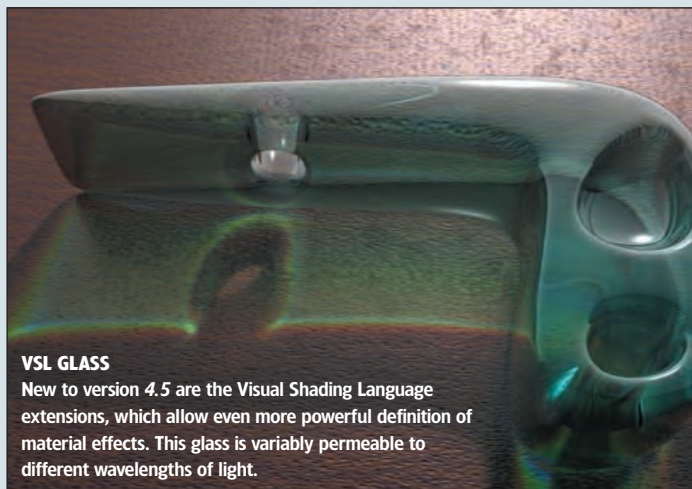
postprocessors, and *Realsoft3D* is no exception to this. In fact, it has a powerful battery of post-processing options, from the standard glow and lensflare effects to very precise compositing effects. A lot of this functionality is down to the use of custom rendering channels. It's possible, for example, to create a

custom light which is invisible to the rendering process. The light cast by this illumination source can be captured in a data channel to control a post-processing effect such as a glow or in fact, anything else you care to define. This not only opens up lots of possibilities for unique and customisable effects, but can



### POST EFFECT PARTICLES

Particles can be used for all sorts of post rendering effects, from lens-flares to glow or smoke effects.



### VSL GLASS

New to version 4.5 are the Visual Shading Language extensions, which allow even more powerful definition of material effects. This glass is variably permeable to different wavelengths of light.



# ReviewsRealsoft3D



This scene has only one light, and demonstrates flexible global illumination.

« significantly reduce the amount of work required to turn raw rendered work into a completed project, particularly if you are combining with video footage.

## Interface

Applications running with *Motif* on Linux always seem to be somewhat problematic. *Motif* just seems to be clunky and slow, although this does vary between the different implementations of the motif libraries you use (*ICS Motif*, *Lesstif*, *OpenMotif*, etc). Running *Motif* on X generates a significant overhead in itself, and seems unresponsive even on a well specced machine. From a cross-platform point of view, *Motif* may have its benefits, but there are other



Realsoft3d really is a Swiss Army knife of modeling and rendering.

solutions which certainly work better under Linux (e.g. Qt), and might be a better bet in the long run.

## Distributed rendering

### Sharing the load

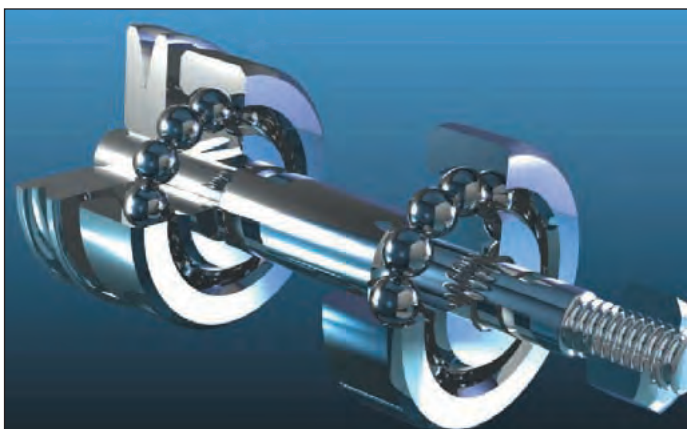
Realsoft have been very generous with the licensing model for *Realsoft3D*. Site licences cover simultaneous users, but there's no restriction on the rendering clients. So, a single user can still use a render farm of as many network boxes as they like, in contrast to most of the other proprietary 3D software.

The rendering setup is quite simple. As well as the option to run rendering as multiple threads (for SMP workstations), you can also identify a number of hosts.

The software uses its own socket at 10050 to communicate between

rendering clients. All that is required is a standard TCP/IP network connection. In the 'hosts' box, of the render>distrib menu add the IP address or hostnames of the boxes to be used for distributed rendering and off you go.

Individual frames will be split into 'boxes', which are rendered independently by any available processor on the network. The optimal number of boxes depends on the complexity and size of the scene as well as the capabilities of the rendering hardware, so some experimentation will be required.



There are many different rendering settings, and shiny objects like this can take minutes or hours to render depending on how realistic you would like the reflections to be.

Realsoft have also taken the interesting step of scripting the interface. This enables it to be completely customised, though you may be a trifle alarmed when the window starts resizing all the elements on startup.

## Conclusion

This software is currently being sold as a commercial beta – i.e. there are known problems with it and it isn't fully stabilised. The intention is that people who buy it now and contribute to the testing process will receive the full release version free when finished. This is certainly reflected in the price, which is a lot lower than similar software on Linux or other platforms.

While it should work on any i386 based Linux, it seems that the software is happier on the likes of Red Hat. While relatively stable there were occasional seg-fault problems which may dissuade you from trusting the software with your life's work, but most

of the problems are acknowledged and being worked on.

Parts of the 'drag and drop' interface seem to be a bit hit and miss. Using it to allocate materials to objects often picked up the wrong material, and sometimes didn't seem to apply it at all. These are small annoyances in the main though, and the full release is eagerly awaited here. The rendering quality is fantastic, as these images show. **LXF**

## Cool effects and features



### DISPLACEMENT MAPPING

To get across the idea of how useful VSL surfaces can be, here's a bear. The fur (which is self shadowing) is not an object, it's a procedural texture using displacement mapping.

## VERDICT

Features	9/10
Ease of use	7/10
Performance	7/10
Value for money	9/10

Affordable and very powerful. Hopefully minor niggles will be sorted out soon.

## LINUX FORMAT RATING

8/10



## MINI-ITX SYSTEM

# Netbox Cubit

Is your PC too large, noisy and power hungry? If so, **Richard Drummond** examines a system that may solve your problems.

**Small, quiet, attractive PC. See also the OpenBrick E, Compaq Evo D500, and the Shuttle SB51G.**

■ **MANUFACTURER** Netbox  
 ■ **WEB** [www.netbox.co.uk](http://www.netbox.co.uk)  
 ■ **PHONE**  
 +44 (0)1453 840171  
 ■ **PRICE** From £375 + VAT

**L**et's face it. The average desktop PC is an ugly, lumbering beast that consumes desk space and power, and spews out heat and noise. If you want a general purpose PC that is less obtrusive, perhaps one that wouldn't look out of place in a living room, then Netbox's Cubit may be just what you are looking for.

The Cubit is based on VIA's Epia 800 motherboard, a mini-ITX unit powered by an 800MHz C3 processor. This device offers x86 compatibility (including MMX and 3D-Now) with an unmatched performance to power ratio. While it won't break any speed records, it puts out a respectable 1600 bogomips and is more than capable for desktop use or for powering a file, print, or mail server for an office or home network. The Epia 800 board features on-board audio, video and LAN, but upgradability is sacrificed for the small form factor. It has only one PCI slot, and the processor is surface-mounted. Upping the processor power of your Cubit requires replacing the whole motherboard, but at these prices that's not such a wrench. It does have two DIMM slots for memory, and Netbox ship units with either 128MB or 512MB installed.

## Down with beige!

The most immediately striking aspect of the Cubit is its custom case design. This is cut from 5mm-thick anodised aluminium (a white plexiglas case is also an option) and comes in a choice of colours: red, blue, green, black or silver. With a footprint of 200mm by 200mm, it requires a quarter of



the living space of a standard desktop, and around half that of the average tower. As well as being stylish, this case is incredibly sturdy too. You could drive a truck over it with making a dent.

Shoe-horned into this attractive case are the mini-ITX board, a 150W power supply, a 3.5" hard drive (a whisper-quiet 60GB Seagate Barracuda on our test model) and a laptop-format CD-RW/DVD combo drive (a cheaper CD-ROM drive is an option). And there's still enough room for a half-height PCI slot (populated with a PCI ADSL modem in our test unit) and an optional Compact Flash reader.

The modem supports bridged Ethernet, PPPoE and PPPoA connections and is a efficient solution for those that require ADSL access, since it frees up an Ethernet or USB port. The compact flash unit is one that attaches to the ATA bus and appears to the system like an ATAPI floppy, so is transparent to use. The BIOS supports booting from a Flash

card, too, so this handy extra opens up the possibility of turning the Cubit into a disk-less thin client.

The front of the Cubit is a sleek affair, with only the CD drive, power button and two of the four USB slots on show. Around the back, it boasts all the usual ports of a desktop machine (see photograph, *opposite*).

In operation the Cubit isn't quite silent, but it's a marked improvement over the noise pollution caused by many desktop boxes. The low power requirements of the Epia 800 board means that the Cubit can get away with smaller, quieter fans than usual – plus the hard drive and CD drive are both very quiet devices. Air flow through the case is good via the vents in each side and on the underside.

By the way, if you want to personalise your Cubit, for an extra

fee, Netbox will machine the side-vents with any design or logo you request.

The drawbacks of the compact case design are a lack of expandability and difficulty of access. OK, so there's only one PCI slot, anyway, so maybe the first isn't much of an issue, but the Cubit's case is a real pain to take apart. The CD-drive is fixed to the lid, so you have to unscrew the lid and either the front or back panels to be able to gain access. The hard drive is buried underneath the motherboard itself, so replacing the hard drive will be a real chore. Still, you can't have everything. Besides, how many times do you really need to take your machine apart, anyway?

## Life, Linux and Cubit

The Cubit can be shipped with Windows XP for an extra fee or with Red Hat 8.0 for free (guess which one I would pick). Red Hat 8.0, to my mind, is Red Hat's most impressive release to date, and will satisfy most needs whether you want to use the

Cubit as a desktop or server. If you prefer a different distro, you should encounter no installation problems with the current release of any of the major distros.

The VIA motherboard is mostly well supported by Linux. The kernel IDE driver copes with the UDMA modes of the on-board ATA-100 adaptor, and in combination with the 7,200 rpm drive supplied, puts out a good head of steam. Testing with *hdparm* reported a raw read speed of just a shade over 40MB/s. (My main work machine, a 1200MHz Athlon box, manages about 32MB/s with an older 40GB drive and a similar VIA chipset.)

The integrated chipset on the Epia does have its prices, though, and *hdparm* reported access to the disk buffer cache as 70MB/s (whereas my Athlon box gets around 205MB/s).

The on-board graphics, the Trident CyberBlade/i1, is supported by *XFree86* 4.2.0, and most distros will detect and configure this device with no fuss. Performance-wise, it's not stunning, but is capable enough for desktop use. The shared-memory architecture means that you need to steal main memory to acts a frame buffer for this device, and performance does suffer under high

CPU load. Try running any software-driven GL application, and try to move windows around on screen, and you'll see what I mean. The most annoying aspect of the CyberBlade processor, however, are the unsupported features. It has on-board 3D acceleration, which is completely unusable on Linux due to a complete lack of drivers, and TV-out support is scarce too. You can get modified *XFree86* drivers from VIA that will enable the TV-out, but this has yet to be integrated into the main *XFree86* tree, so you'll have to download and build them yourself. (There's also an experimental project to develop a kernel framebuffer driver that supports TV-out display. See <http://epiafb.sf.net/>.)

The final niggle is if you opt for the PCI ADSL modem option. This is a really neat third-height card, based on the Globespan Pulsar chipset, but Linux support is only available as a binary-only driver, downloadable from the manufacturers, Traverse Technology (see [www.traverse.com.au/downloads/drivers](http://www.traverse.com.au/downloads/drivers)). You'll have to download and install these manually, since they are not provided with any current distro. Traverse provide generic drivers for 2.2.x kernels and drivers for the 2.4.18 in Red Hat 7.3.

If you have another 2.4-based distro, then you have to compile your own driver by submitting your kernel headers to an online compilation engine. Yuck! If they must supply closed-source drivers, can't they at least do it in the way nVIDIA and others do it: provide a binary-only core and ship the kernel glue as source code which you can build against any kernel yourself.

On the plus side, the Traverse drivers make the ADSL modem appear to the system as an Ethernet device, so use is transparent once it's configured. Living in Bath, however, where ADSL is virtually non-existent, we were unable to test this device beyond loading up the drivers.

## Reclaim your desk

The Cubit is an impressive piece of kit. It's stylish, cheap, and packs more than enough processing power for most people's needs. You do have to make sacrifices for the small footprint and low power design, most notably in terms of expansion capabilities, but for many this will be a trade-off they are willing to make. If your requirements are for a modestly-spec'd PC, that won't domineer your office space or living room, or cause your ears to bleed with the din of



**For the personal touch, Netbox will machine your logo into the case.**

constantly whirring fans, then the Cubit will be an excellent choice. The fact that it is a thing of beauty rather than a dully, putty-coloured box is an added bonus. **LXF**

## VERDICT

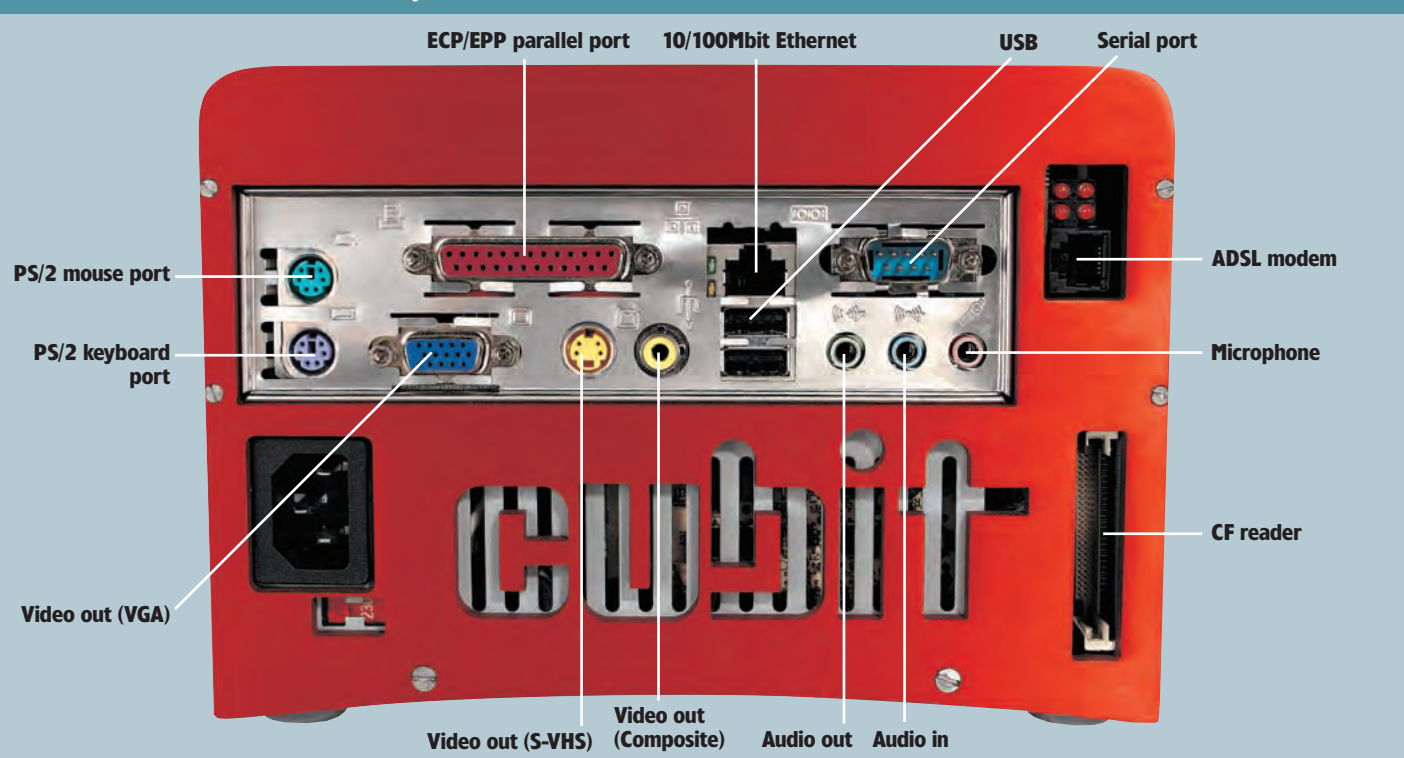
Ease of use	7/10
Performance	8/10
Features	9/10
Value for money	9/10

Proving PCs don't have to be dull, the Cubit packs an amazing punch in a compact, low power design.

**LINUX FORMAT RATING**  
 **8/10**

## Cubit ports

This little box boasts a lot of connectivity.





## ReviewsKohanAG



The three different campaigns do offer better value than the original.



The editor allows you to create custom, pre-populated maps.

## FANTASY RTS GAME

## Kohan Ahriman's Gift

Nick Veitch revisits the land of the Kohan to find out what's changed.

**Predictable but diverting RTS fare. Consider also the original Kohan, and Heroes of Might and Magic.**

- **DEVELOPERS** Timegate / Transgaming
- **PRICE** \$29.99
- **WEB** [www.transgaming.com](http://www.transgaming.com)

**W**hen we reviewed the original *Kohan* game back in *Linux Format* 33, it gained average scores for being a somewhat dated, but playable RTS game. So can the postdated prequel improve on that?

Firstly, we should outline that the main game engine is almost exactly the same, as are the graphics. Like most real-time strategy games, this is a mix of resource management (build quarries, mines), and objective-based strategy (make units and capture cities, build towns or whatever). *Kohan* forces are organised in units of six and one leader, with some flexibility. You can create mixed units – for example, four infantry with two archers for support – or support basic types with a variety of magic-wielding auxiliaries.

The units fight with their own AI, and much of the micro-management

of other RTS games is thankfully absent, leaving you to direct strategy rather than make sure each individual combatant is lined up properly.

Most of the scenarios follow the by now familiar formula – extend your base of resources, amass a large army and then pummel the enemy into non-existence. A few time-based ones have been thrown in, and there are some unusual conditions. Usually this means playing the mission once or twice just to find out what is going to happen before you can actually play to win.

### Campaigning

There are three different campaign threads to follow, which sort of co-exist with the original story. Although the game is billed as being a prequel, some of the campaigns take place as part of the original story. The main difference here is that you get to play some different factions, and there get to fight with some of the units which were only ever your enemies in the original game.

Playing the different campaigns requires slightly different tactics. The Slaanri lizard people for example are quite tough, but slow moving and quick to flee combat. You really need to mass more of them together and try to surround an opponent rather than slug it out. The undead armies

have good morale and are pretty resilient, but don't do a lot of damage, and need support of missile or magic-wielding allies to be really effective.

### Flaws

Most of the same flaws are still present from the previous game. Although your units can increase in experience, all improvements, and units are lost at the end of the scenario. This takes away any feeling of connection with your troops. You can't have a favourite squad of crack troops, because at the beginning of each level you start afresh.

The enemy AI seems to be more intelligent in this game, but that might just be down to better prepared scenarios. Some of the missions are certainly pretty challenging, though these are mixed up quite a bit. You may complete a scenario that has taken four or five hours and been a very close thing only to find the next one is a walkover.

### New additions

The new story gives the opportunity to introduce a few new unit types into the mix. Some don't really add greatly to the gameplay – The 'Ice Drake' is pretty much the same as the dragons from the original, except they live in cold climates and use ice as a weapon instead of fire. Some of the units do

### Transgaming

The *Kohan* game runs under Transgaming's WineX system. It is essentially exactly the same as the Windows version of the game with a few tweaks. The game is available directly from transgaming's website, but you'll need a fast connection as you're looking at 150MB of downloads.

offer the ability to alter tactics – The Crossbowmen for example are slower than normal archers, but do more damage. Most of the factions in the game have at least one new unit type.

It's also still possible to create your own scenarios using all the new units, and to play multiplayer (though everyone must have this version of the game, rather than the original).

The different unit types and new campaigns probably make this a better value proposition than the original game (especially as they are the same price), but if you weren't that impressed by the original, not much has changed here. **LXF**

### VERDICT

Graphics	5/10
Playability	7/10
Features	6/10
Value for money	7/10

Better value than the original, but fundamentally the same game with different campaigns.

**LINUX FORMAT RATING**  
**7/10**

## SYSTEM ADMINISTRATION

# Webmin 1.0

**Marco Fioretti** reports on a configuration system with something for everybody, from clusters wizards to home surfers.

**Remote admin solution**  
rivaling Plesk Server  
Administrator, NetOp Remote  
and Volution at the lower end.

■ **DEVELOPER** Jamie Cameron  
■ **WEB** [www.webmin.com](http://www.webmin.com)  
■ **PRICE** Free

**W**ebmin is a Unix administration tool suitable both for professional system administrators of company servers and for SOHO desktops. It can be accessed through any web browser, locally or remotely, and provides one common front-end to many configuration and maintenance tasks.

*Webmin* can be easily extended with custom modules, and offers the same interface on almost every distribution of Unix, BSD and Linux. This means that the administrators of heterogeneous networks can, at least in certain cases, configure different platforms always in the same way, without remembering ten different locations of the same configuration file, or ten different names for the same parameter.

*Webmin* is written in Perl: internally, it consists of a simple web server calling CGI scripts which display and allow to modify in a more readable way the content of the standard config files. Once it is installed, to manage the local computer simply point the browser to <http://localhost/10000> and log in as root. When this is too risky (i.e.,

## Mastering Webmin

The ultimate reference

*Webmin* users now have a comprehensive resource available, including many tutorials: it is *The Book of Webmin ...or How I Learned to Stop Worrying and Love Unix*, by Joe Cooper, published by No Starch Press: *The Book of Webmin*, December 2002, 304pp. ISBN 1-886411-92-1 [www.nostarchpress.com](http://www.nostarchpress.com)



Figure 1: The system window of *Webmin 1.0*.

across the Internet) the box could be set up to accept *Webmin* connections only if local, and not start it at boot. When needed, both *Webmin* and a web browser could be started on the remote machine, with the output displayed on your PC, through SSH, and without worrying about bandwidth.

As already mentioned, the HTML code generated by *Webmin* is viewable with any browser, including the console ones: the only exception are those modules, like the Telnet client, which require Java or JavaScript.

*Webmin* can make almost every aspect of system administration easier. It knows about disks, filesystems, users, boot and *cron* processes, as shown in **fig 1**. Networking, Hardware and Clusters have dedicated windows too. Custom commands can be run, and their output logged from *Webmin*. All the standard intranet and Internet services, from mailing lists to MySQL, including CVS, Apache, Jabber, SSH and many more,

can be configured from the "Servers" window. It is not by chance that a *Webmin* control panel is offered by many hosting providers as a free, Open Source alternative to more expensive solutions like *Cpanel* and *Plesk*.

## Try this at home!

So far, it may seem that *Webmin* is only useful to network professionals, but this is not the case. First of all, several of the tasks I mentioned should really be familiar also to responsible single users. In the second place, *Webmin* can help a lot also in the everyday use of a stand alone desktop. Personally, I restored my MySQL databases from text dumps in a few seconds, and was able to define and schedule very quickly a regular hard disk backup procedure. Such procedures can go from the simple generation of a tar file from a predefined file list, to mirroring whole directory trees on a CD, with the click

of one button. The "CD Burner" module in the Hardware window supports this and several other use cases, including CD to CD copies.

I also like a lot the Perl and *GnuPG* modules. The former lists installed Perl modules, and can fetch and install new ones from CPAN or local disk. The latter is now helping me to create and manage all the private and public keys my PC must know about. Users with mixed home LANs will also appreciate the interface to *Samba* file sharing. Email administration is another fundamental need shared, obviously at different levels, by both home and network sysadmins. Apart from mailing list setup, *Webmin* makes it possible, out of the box, to manage *fetchmail*, *procmail*, and three different SMTP servers: *sendmail*, *qmail*, and *postfix*.

## Test ride summary

To write this review, I installed *Webmin 1.030* on a K6-2 350MHz, with 128MB of RAM, running Red Hat 8.0, using the RPM linked from the home page. The install script recognised the distribution and, in general, worked perfectly: since all *Webmin* needs are Perl and a shell, I have no reason to expect a different behaviour from the tar file, or in any other distribution. In a few seconds I was testing the tool with *Galeon* and *w3m*: both browsers had no problem in rendering *Webmin* pages, and the tool was fast enough, even on that relatively slow machine.

In my opinion, there is only one small glitch in the install procedure of *Webmin* (and of many other applications, from *Emacs* to *gkrellm*): the package contains code for all the supported platforms and languages. Since it does recognise the operating system, and could do the same with the system language, why doesn't it put on disk only the files needed by that combination? This may help whenever old machines with very small disks are used as thin servers.

The documentation on the website is almost always exhaustive, but there is no link in the *Webmin* start page. The single modules, however, do have links to documentation, including hyperlinked man pages, and sometimes also entry forms for Google searches.

## New In Webmin 1.0?

Version 1.0 of *Webmin* was released on September 12<sup>th</sup>, 2002. It includes a lot of bug and security fixes, and



## Usermin

Reduced system administration for users

*Usermin* can be defined as a reduced version of *Webmin*: it offers to all users of a Unix system services like webmail, SSH setup, and file management. It has the same graphical interface and architecture of *Webmin*, and is available from the *Webmin* site under the BSD license. Not surprisingly, it can (and should) be entirely configured from *Webmin*.

several performance improvements. Predictably, it also includes support for new distributions, for example Gentoo, and for the latest releases of other ones, like Red Hat 8.0 and SuSE 8.1. As far as features are concerned, the main new ones are:

- Linux *iptables* support
- Voicemail Server
- Logfile Analysis
- Improved SSL key management
- Some additions to the *Webmin* API
- File manager improvements

The *iptables* module (see **fig 2**) was a much needed addition. While sufficiently flexible and functional, it shares with other *iptables* front-ends the limit described some time ago in *A Comparison of iptables Automation Tools* (<http://online.securityfocus.com/infocus/1410>):

"The idea ... still has not found its implementation, particularly one [for] people who do not understand the technical details of packet filtering ... its

Linux implementation [and] some of the *iptables* internals." People who do have this background, of course, will be able to make custom firewalls quicker. This is obviously not a limit in *Webmin*: packet filtering is inherently not trivial, and the responsibility of delivering a safe starting point, *i.e.* an initial installation that doesn't let *anybody* in, belongs to the distro packagers. Not to whatever takes charge later.

Another major novelty is the Voicemail Server (see **fig 3**), which provides answering machine capabilities to the system by configuring and starting *vgetty*.

Received messages, as shown in the figure, are listed, played back or deleted in a separate window (note that, at least by default, only root can do that). Other options that can be set up in *Webmin* include the automatic forwarding of all recorded calls as WAV email attachments (which practically solves the root access problem), or custom commands to run on each incoming message. Even the greeting message can be unique, or randomly chosen at each call from a given list.

Don't forget that *vgetty*, once started in any way, will cling to the chosen modem and, until explicitly stopped, intercept all the calls. Including those made from the PC to connect to the Internet, as I found, with some embarrassment, in the frenzy of testing. Theoretically, the Voicemail server could be improved to check and issue a warning if the serial

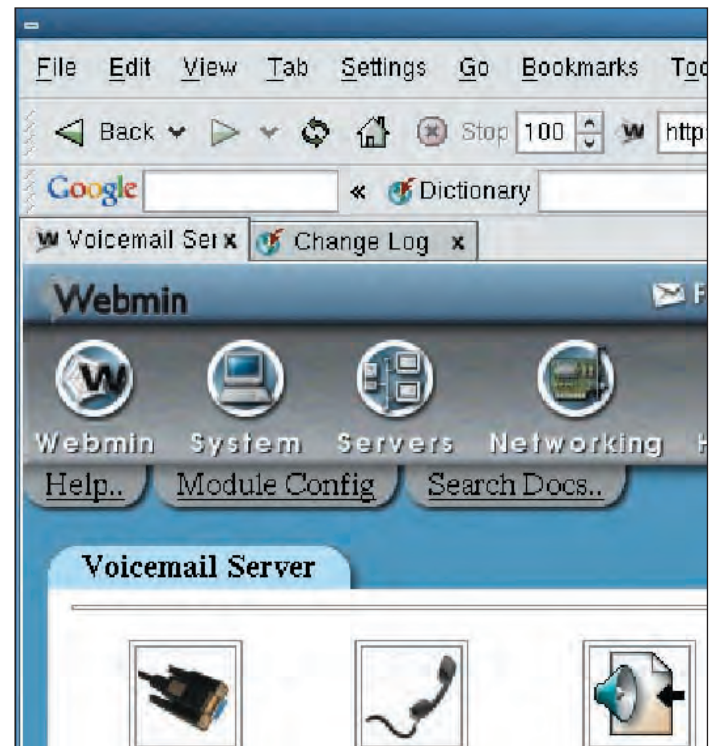


Figure 3: The *Webmin* answering machine.

port is already linked to */dev/modem* or used in *pppd* scripts. However, I have no idea of how complex it would be to make this function really cross platform, and not even if it would be really worth the effort.

Back to Internet services now. From this point of view, as listed above, the main new entries are *Webalizer* and *SSL*. *Webalizer* generates graphic reports from HTTP, FTP and proxy access logs. Its *Webmin* module also supports *Squid* 2.5 directives, and *Squid* delay pools. Several filters can be defined to show or hide daily or hourly usage, as well as sort accesses by Client, Referer, Countries or other criteria.

SSL support in this release has improved: first of all, *Webmin* does not use a built-in key anymore, but generates a unique one at install time. In the second place, some bugs and weaknesses in the key generation mechanism have been fixed in Release 1.03. Please check the mailing list archives and the documentation of the latest version for more updated information.

Since *Webmin* modules are basically CGI programs, custom ones can be easily written: the website has a whole section listing all those contributed by *Webmin* users. So far, most modules have been written in Perl, but nothing prevents the use of other languages. Version 1.0 includes

some additions to the existing API which increase this interoperability.

Last but not least, the File Manager, which resembles the GNOME or Windows ones and requires Java support: the module in *Webmin* 1.0 is finally capable of sorting the file list of any column, by clicking on the heading.

## Conclusion

Remote administration over the Internet must be allowed only when the box has been carefully configured and is constantly updated and monitored: desktop PCs connected to the 'Net should simply not allow *Webmin* access from the outside. That said, there is no doubt that *Webmin* does a good job, on any box you want to use it, and in a less intrusive way than *Linuxconf*. **LXF**

## VERDICT

Ease of use	9/10
Features	8/10
Performance	8/10
Value for money	10/10

*Webmin* is a nice and easily customisable way to give professional care to any Unix box. Try it!

**LINUX FORMAT RATING**  
**9/10**

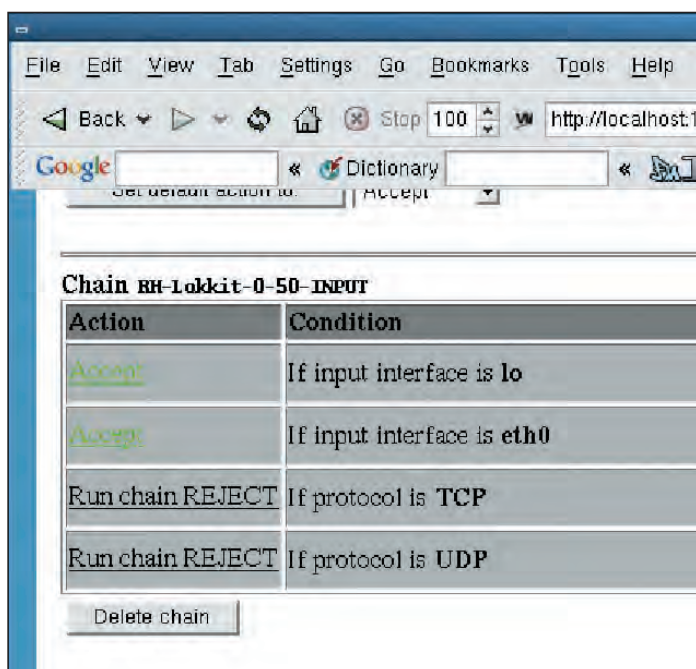


Figure 2: Linux Firewall configuration.

## PHP ENCODING

# Protecting your PHP

**Paul Hudson** looks at the latest wave of PHP encoding solutions, designed to protect your code in the wild.

**U**nlike languages such as C and Pascal, which are compiled to unreadable machine code, PHP is interpreted direct from source code and this means that when you distribute an application you have made, you are supplying your users with an entire copy of your source code.

In the open-source world, this of course is not a problem – a product installed on your machine would feel a little naked if the source code wasn't available! However, in the commercial world where everyone wants some return on their investment, handing out source code isn't always viable. In order to give programmers some options in this regard, Zend once again took the lead and created the very first PHP encoder soon after PHP 4 was

released. This took PHP code, and essentially mashed it up into unreadable nonsense that was “un-mashed” and run normally when executed by PHP. Of course, that was some time ago – things have moved on a great deal since then. Known for their excellent *PHP Accelerator* product, ionCube have released their own, competing product, and Zend have also recently released their much-anticipated *SafeGuard Suite*. I've put each of these two products through their paces using a variety of tests – read on to see how they compare...

## Encoding?

The process of encoding is made up of several smaller, clearly defined stages. At the core of the procedure, surprisingly enough, lies the Zend

engine itself. If you read that and didn't blink, go back and read it again – both Zend *SafeGuard Suite* and the ionCube *Encoder* both make extensive use of the Zend Engine internally as part of their encoding process.

You see, the key stage in the encoding process is to run the target PHP script through the Zend parser itself, stopping the process at a crucial point where the script is sitting in memory ready to be executed. This is then stored directly as a binary file on your hard drive, and therefore quite meaningless to the human eye.

Technically speaking, it's possible to reverse-engineer the binary code, however, because it is essentially just a direct representation of the Zend intermediate code generated from the script, and it's possible, albeit difficult,

to extrapolate the approximate PHP source code that would generate given intermediate code.

In order to push the envelope, however, there is now the possibility to pre-process your PHP scripts in order to both improve performance and add obfuscation – the code in your scripts is juggled in unusual and unlikely ways in order to make reverse engineering nightmarishly difficult, if not approaching the impossible.

So, given that the Zend Engine itself makes up perhaps the most important component in both products, what is there to separate the two? You'd be surprised! Note that all encoding tests were performed with *SafeGuard's* optimisations set to **0** so that it encodes as fast as possible, and ionCube's *Encoder* was set to the – fastest option, which trades some run-time performance for encoding speed.

## ionCube PHP Encoder

- **DEVELOPER** ionCube
- **WEBSITE**  
[www.ioncube.com/encoder](http://www.ioncube.com/encoder)
- **PRICE** \$349

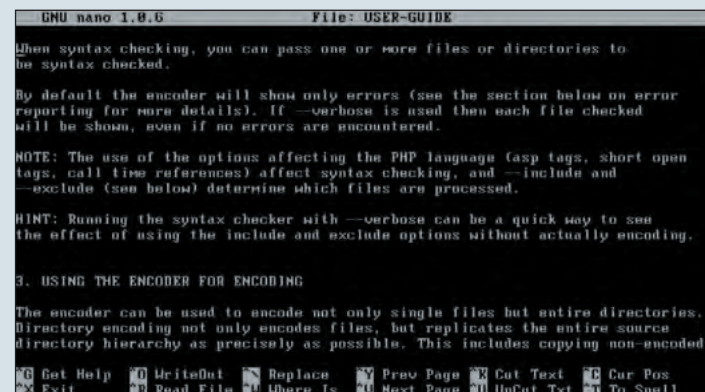
**Having been much impressed with their *PHP Accelerator*, I was eager to give ionCube's *PHP Encoder* a try. If you're familiar with ionCube, it'll come as no surprise that *Encoder* is entirely command line based. While this does make the learning curve higher for many users, the choice is mostly justified by the ethos of the product – it's designed to be implemented directly into the build cycle of your product. As it stands, though, the ionCube *Encoder* lacks even a simple GUI, and I feel that one offering even the most basic options would give users extra choice, which is never a bad thing. I was glad to see that the documentation supplied (my favourite area to pick on when reviewing a product) included a complete user guide that, despite missing a table of contents, was fairly easy to get information from due to its friendly**

tone. Also supplied was a thankfully succinct quick start guide that got me encoding files immediately.

## Using the encoder

The ionCube *Encoder* takes a variety of command-line parameters, starting with simple but helpful options like “syntax check only”, on through customising the speed and efficiency of the encoding process (you can select whether to encode faster or smaller), and finally up to more advanced options like restricting code usage to a particular IP address. Each of these options are helpfully-named, which means that you can run with parameters **--allow-asp-tags** as opposed to the usual cryptic single-letter options.

When encoding, the program by default automatically re-writes each file with a special “pre-amble” that is unencoded PHP designed to check for the existence of the ionCube *Loader* – the PHP module to decode scripts for execution. If for some reason the loader isn't available, a friendly error



**A first-class user guide from ionCube, something I'm always glad to see.**

message is produced, providing instructions to your end users as to how to fix the problem. This can be disabled for those wishing to maximise speed while minimising execution time.

With regards to selecting your input and output, you can choose to input one file or a whole directory. This functionality allows you to encode entire projects from one command – you can automate the entire process into a makefile, then read through the errors all at once at a later date. This

is further enhanced by *PHP Encoder* outputting errors in standard *Emacs* format, meaning that *Emacs* users can cycle through encoding errors directly from their preferred environment. Whilst encoding, *PHP Encoder* automatically syntax checks documents as it encodes them, and outputs helpful error messages as it goes. The error messages aren't quite what PHP itself would output, but they're close enough to be recognisable to most of us. Given that the messages are almost certainly

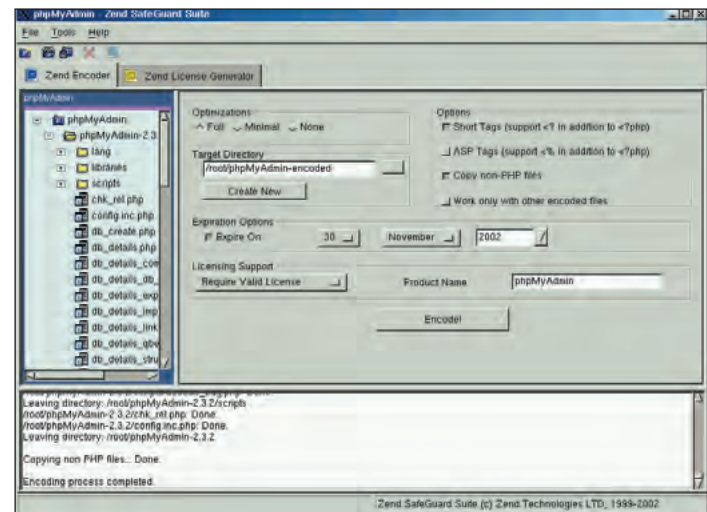


## Zend SafeGuard Suite

- **DEVELOPER** Zend
- **WEBSITE** [www.zend.com/store/products/zend-safeguard-suite.php](http://www.zend.com/store/products/zend-safeguard-suite.php)
- **PRICE** Starting at \$960 (includes Zend Studio)

I was a little surprised by Zend's *SafeGuard Suite* at first simply because I experienced a few installation problems on my laptop – something unheard of from a company that makes installation as easy as sneezing! However, the Zend support team were quick, helpful, and up-to-speed on Linux, so the problems I encountered, rare as they probably are, were soon resolved. The docs provided are far better than I have come to expect from Zend, however they still let themselves down with various clumsy errors that should have been picked up before the product was released. While I freely admit that I love to pick on docs, I think anyone would have problems with documentation claiming

the product works on “Windows NT, Windows 2000, or Windows XPY... Windows 98/NT 4.0/2000/XP” (yes, it mentions the Windows versions twice, including the as-yet-unheard-of Windows XPY) and has peculiar notes written either to or from the mysterious “ZDE Team” with suggestions on how to improve the docs. Come on, Zend – you should be better than this by now! Once the documentation is behind you – and, trust me, it's better left behind you – the product starts to shine, and it more than makes up for early problems. The GUI itself, pictured *right*, takes all the hard work out of securing your programming investment. Zend have clearly expended a great deal of effort making the interface at once easy to learn and use whilst also being fast to get around once you're keyed in to how it works. Even though it's anecdotal, I had the feeling that encoding took a little longer from the encoder GUI than from the command prompt, although the difference was minimal, and more than made up for by the ease of use factor.



Good for veterans and perfect for beginners, the *SafeGuard Suite* GUI is compact, well thought out, and a match for its prompt-based sibling.

One curious point to note, though, is the checkbox marked, “Copy non-PHP files”. This is a very helpful feature in itself, but points to a peculiar line of discrepancies in the feature set of the product. To start with, nowhere could I find documentation on how to make use

of this very important feature from the command line – quite a frustrating problem in that without this option any non-PHP files that make up a project (such as help files or images) aren't copied to the output folder. Following on from this peculiar omission are a

generated by the Zend Engine itself, it seems odd that the ionCube *Encoder* trades in *Emacs*-style reporting for less descriptive error messages, although this may be attributable to a recent change in the internal PHP system itself. The encoder can be optionally configured to *only* syntax check (that is, not encode) which is a neat addition, if somewhat redundant given the **-I** option in the PHP CGI.

### Speed

With regards to how long it takes to encode files, I ran ionCube's *Encoder* through a series of encoding tests including encoding *PHP Nuke*, *SquirrelMail*, and *phpGroupWare*, and it performed excellently, taking just six seconds to encode *SquirrelMail* on its fastest settings. To compare, the same test took upwards of 15 seconds using the command line version of Zend

*SafeGuard Suite*. Similarly, ionCube *Encoder* took just 27 seconds to encode *PHP-Nuke* as compared to 61 seconds using Zend *SafeGuard* – certainly quite a difference! It's arguable, though, that the speed difference really isn't all that important, however. I say this because, as mentioned already, the focus of the product as given in the accompanying user guide is to help encoding fit into the project build cycle, and more often than not building a product is a rare, and slow thing to do. I'm sure everyone reading this has hit **Enter** on a *make* command and gone off to watch TV for twenty minutes or so, so waiting a little longer for encoding to complete is probably not a major issue.

### Pricing and features

Despite the lack of a GUI, ionCube *PHP Encoder* does fairly well for itself

on the features front, with a lot more to come in the imminent v2. At the time of writing, however, only v1 is available, which lacks much of the licensing functionality included in *SafeGuard Suite*. A four-machine licence of the ionCube *Encoder* weighs in at \$349, which is certainly very affordable, however, as I've said, this doesn't include any licence management.

At the time of writing, the ionCube website states that a package of *Encoder v2* and a licence manager, when released, will weigh in “well under \$1000”, which may well be a dig at the old Zend *Encoder* licence fee of \$960 for one year of encoding. Currently, licensing options are restricted to limiting execution to a certain IP address, and also forcing scripts to expire on a certain date.

One potentially serious drawback to the ionCube *Encoder*, however, is that it doesn't perform any code obfuscation at all, which makes the output of the encoder easier to reverse engineer by people with the right tools. That's not to say reverse engineering the code is easy – far from it!

However, suffice to say that the ionCube *Encoder* doesn't go to the same lengths as Zend *SafeGuard Suite* does in order to protect your code.

### VERDICT

Ease of use	8/10
Features	7/10
Performance	8/10
Value for money	8/10

The fastest encoder available, but achieves this by skimping on obfuscation. Great value at this price, and a solid prompt-based encoding solution

**LINUX FORMAT RATING**  
 ////////////// 8/10

**“I want to bring encoding to the masses, not just to the elite few with big budgets”** Nick Lindridge, ionCube

# ReviewsProtectingPHP

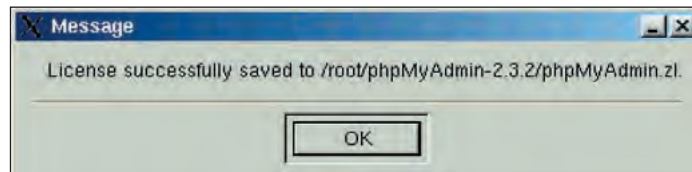
« collection of other unusual quirks which don't appear to have been ironed out quite yet. For example, one test I did was to have one file, "test.php", with a symlink, "symlink-to-test.php". When this was encoded, my output directory contained two standard files, test.php and symlink-to-test.php. That is, symlink-to-test.php was no longer a symlink – *SafeGuard* had converted it to a file in its own right. While these are potentially minor points, they do add up into a more serious issue that I hope Zend will be addressing soon. However, these glitches only take a small amount away from Zend's achievement with this release. The new *Zend Licence Generator*, for instance, allows you to customise the encoded project your produce so that it's registered to a particular user, locked to a particular host ID or IP address, or you can even make it expire on a given date.

## Speed and features

The *SafeGuard Suite* drags a little: encoding large projects take a very long time, and I even had one test – that of encoding the gigantic *phpGroupware* project – fail to encode. Much of the time spent encoding can be attributed to the extensive optimisation and obfuscation that the product performs,

## "The need for intellectual property protection and licensing is a must. We go the extra mile to fulfil that demand"

Zeev Suraski, Zend



**I must admit I went through the licensing procedure several times in my first test, simply because I refused to believe it could be this easy**

and I think most people will be more than happy to accept longer encoding time in exchange for faster execution and harder to decode output files. The features of *SafeGuard Suite* are a long way ahead of anything else that is available right now, which brings the price tag into perspective – you get what you pay for. Perhaps my favourite piece of functionality is the ability to force scripts to cease working after a certain date. With features such as that just a mouse click away, Zend certainly have a winner on their hands.

The command line version of the product is a fairly complete replica of what you see in the GUI, and enables you

to accomplish many of the same tasks using scripts and *cron* jobs – perfect for integration into a build environment. One particular high point of *Zend SafeGuard Suite* is the effort the Zend coders have gone to in order to make your PHP scripts as safe as possible – not only does it optimise your source for fastest execution, obfuscate the code to make it harder to understand, and then save the compiled course as an unintelligible file, but it also internally generates its own intermediate code, which means that even if a team of hackers manage to figure out the file format used to save the file, what they're left with isn't even standard Zend Engine intermediate code!

I think it's fair to say that Zend have taken code protection to a new plateau of excellence with *SafeGuard Suite*. The only things holding them back from runaway success are lack of config options for the command line version, the few small bugs scattered here and there, and the quirky docs. If you're willing to live with these minor annoyances till Zend fix them, you'll find *SafeGuard Suite* to be a lot more than just an incremental upgrade.

## VERDICT

Ease of use	10/10
Features	10/10
Performance	7/10
Value for money	7/10

Offers an impressive array of features, way ahead of the competition, but still has kinks that need resolving

**LINUX FORMAT RATING**  
 9/10

## Conclusion

This is one of the few times I wish I could give out scores out of twenty, because the two products come very close together. However, at the end of the day, *Zend SafeGuard Suite* slides smoothly into the top spot owing largely to its dazzling and well thought-out collection of features and its friendly-yet-powerful interface. With regards to performance, the *ionCube Encoder* takes the lead – often by quite a substantial margin, which is an interesting result to see given Zend's inherent experience in using their own engine. Whether or not the *ionCube Encoder* achieves the speed increase through faster or more efficient code is hard to say, but it's quite possible that a large amount of the speed increase comes through *ionCube Encoder* not actually doing any code obfuscation at all.

Furthermore, your need for the extra speed given will govern how heavily the performance aspect of each product weighs with you. On the basis of features alone, the *ionCube Encoder* is just left standing in Zend's dust, and not just

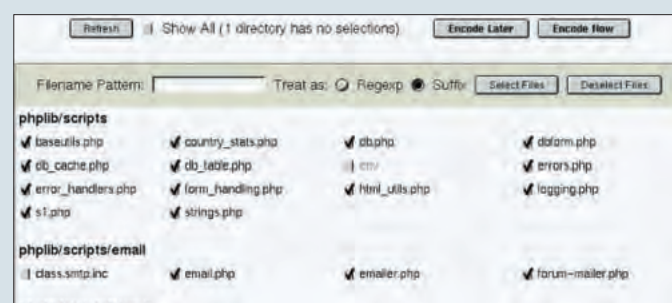
## An alternative to traditional encoding?

Go on, dip your toe in the water...

If you are interested in seeing how encoding works without splashing out to buy either *SafeGuard* or *PHP Encoder*, you have two choices.

First, both products have a free-to-try downloadable edition that allows you to get started to see how things work in your own environment. Note that both versions are hindered in some way, so your performance is likely to be substantially lower than when you purchase the full product.

The other option is to give *ionCube's* online encoding service a try. This service allows you to upload PHP scripts to *ionCube's* secure server, then download the encoded results – you only pay for what you use. For example, using the



**When you upload an archive of files, the online encoder automatically extracts the files and recommends likely PHP scripts.**

online service to encode *Squirrelmail* 1.2.7 would cost approximately £20 – not bad considering it has 177 PHP files to encode!

This bureau-style service provides

a clever alternative to the normal standalone approach, and is perfect for users who are unsure about committing to an investment in PHP encoding.

because it lacks a GUI. The simple fact is that *Zend SafeGuard Suite* is just that – a suite of software, and as such it does a lot more than just encoding files. Of course, that's not to say that the *Zend GUI* counts for nothing – on the contrary, I was very impressed to see a Linux

product have a front-end that was as useful to me as its command line equivalent. Users who have no use for the large feature set offered in *SafeGuard Suite* will find themselves preferring *ionCube's* product – it does all anyone could need, and excels as an

encoding-only solution. However *Zend SafeGuard Suite* features, such as limiting file execution based on date or host machine ID, put it in a league of its own, and, in my opinion, it's a league that's well worth paying extra to get into. [LXF](http://www.linuxformat.co.uk)



# The Art of Deception

Nick Veitch isn't frightened, he's very, very afraid.

■ **AUTHOR** Kevin D. Mitnick  
■ **ISBN** 0-471-23712-4

**K**evin Mitnick is probably the world's most notorious cracker, phone phreak and general scourge of the sysadmin. His dubious talents were put to use at an early age, and before long he was on the FBI's 'Most Wanted' list. He still managed to evade capture for several years by er, using a false name and moving house often. Finally caught and harshly dealt with by the authorities, Mitnick is, in the long tradition of caught culprits, turning over a new leaf and publishing a book on security. Unlike many others though, this one is worth reading.

Technical exploits come and go – there is always some new vulnerability being found in whatever web service you care to mention – but the most damaging security vulnerabilities always seem to involve people.

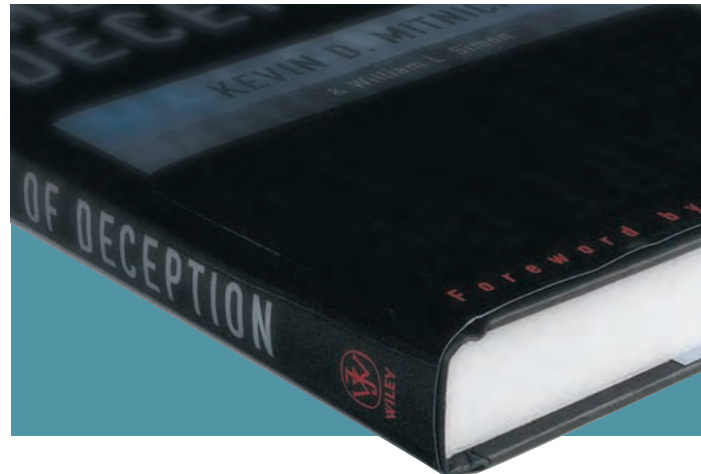
There is a delightful and highly illustrative tale in the book about his early cracking days. He went with a

friend to a computer fair where one company were demonstrating a security product they were so confident in, they were offering a cash prize to anyone who could break into their system with administrator privileges. They provided terminals at the stand, but the terminals were linked to ports on the server to which the software would never give privileged access. The system seemed foolproof.

Foolproof maybe, but not Mitnick-proof. He distracted the attention of the person running the booth while everyone else was at lunch, while his chum picked the lock on the server cabinet and plugged the terminals into the remaining, non-protected ports. Then it was child's play to break into the system and win the prize.

This may not seem in the spirit of the competition, but it illustrates the point that seems to trip up security measures again and again – you can't rely solely on technology to keep the bad guys out.

Most of the book is devoted to social engineering attacks, each highlighting common areas of failure. You'll be amazed at how employees were



duped into allowing unauthorised access to sensitive areas, giving their passwords to someone they didn't know, divulging private customer info over the phone and even faking the source to their most important project to someone who basically called up and asked for it. Each tale makes compelling reading in itself. Of course, no real details are given, and it seems likely that some of the stories have been embellished slightly, but you still get the sense that these things have actually happened, and that Mitnick has himself participated in more than he admits to.

The concluding pages of the book outline an exhaustive security policy to reduce the impact of social engineering attacks. Some of them simply aren't

practical in many companies though – calling back to check on identities isn't going to happen in any busy company.

The book is quite scary. Having read the first few chapters, you begin to imagine what sort of conversations could easily take place in your own company, and how much damage a social engineer could do. You'll never trust anyone on the 'phone again.

## VERDICT

A riveting (and disturbing) read.

**LINUX FORMAT RATING**  
■■■■■■■■■■ 9/10

# Linux In The Workplace

Neil Lucock discovers a practical, well written book for Linux beginners.

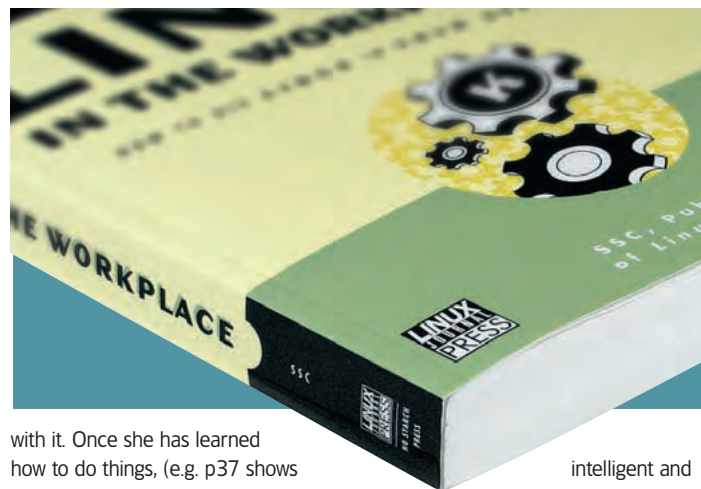
■ **AUTHOR** SSC, Publishers of The Linux Journal  
■ **PUBLISHER** No Starch Press  
■ **ISBN** 1-886411-86-7  
■ **PRICE** \$29.95

**A**lthough this book's title suggests that it is intended for office use, it's also a very useful book to buy if you are new to Linux. It shows the user how to do various tasks and demonstrates what the various programs are capable of. For example, you get 17 pages on *The GIMP*. Not enough to qualify as a reference book on the subject, but enough to get you started. I was particularly pleased to see a whole chapter on the *OpenOffice.org* suite. The chapter after it covers *Abiword*

and the *KOffice* suite. There are details on using the Internet, email, CD burning tools and, of course, they include a detailed guide to the user interface.

That's the KDE interface. GNOME is mentioned briefly as an alternative, but they obviously would rather write about KDE. The index does not mention *Nautilus*, *Galeon*, *Evolution* or other key GNOME tools. They all work under KDE so they should be included. If the book has a weakness, this is it.

This book is ideal for the Office Manager of a company that intends to abandon Windows. She is not a Linux enthusiast, she isn't interested in GPL or computer issues. Someone else configured the computer for her, all she wants to know is how to make it work. To her, it's a tool to do the job, nothing more. Buy her the book and let her get on



with it. Once she has learned how to do things, (e.g. p37 shows her how to make *OpenOffice.org Writer* automatically open MS *Word* files) she teaches the others, and the book is left on the desk as a handy reference.

The book does not delve far into the technical side of Linux, although Chapter 13 covers shell commands. There are over 300 pages with clear b&w pictures on nearly every page. The index and program reference (that tells you what each program does), combined with

intelligent and jargon-free text, make it a good choice for all beginners.

## VERDICT

Well written and worth buying. Ought to be included in the boxed Linux distributions as a reference book.

**LINUX FORMAT RATING**  
■■■■■■■■■■ 9/10

# Just for Fun

**Jonathan Wright** finds that Linux creator Linus Torvalds is as idiosyncratic in prose as he so often is in person.

■ **PUBLISHER** Texere  
 ■ **ISBN** 1-587-99-151-9  
 ■ **AUTHORS** Linus Torvalds and David Diamond  
 ■ **PRICE** £9.99

One of the more enigmatic characters in the Linux scene is Linus Torvalds himself. He rarely gives interviews, and doesn't operate a personality cult like some other Linux luminaries, so the arrival of his biography in 2001 certainly caused a stir of interest.

Co-authored by Red Herring journalist David Diamond this volume outlines the remarkable Linux story from its head honcho's own unique perspective, and has now been re-issued in paperback format, hence its coverage here.

Part autobiography and part sounding board for Linus to bang on about everything from intellectual property and patent laws to the meaning of life (honestly), this is a rare opportunity to get to know one of the world's ultimate geeks, one of the few people who can legitimately claim to have changed the world. Thus we learn that, from an early age, Linus was fascinated by computers and technology. Hardly surprising, especially when you read Torvalds's portrayal of his parents, Nils and Anna, as left-wing idealists caught up in a heady atmosphere of hippydom.

The early days of Linux are covered in the second part of the book, *Birth of an Operating System*, and, while many parts of the story are familiar, it's intriguing to get Torvalds's perspective on events. Nevertheless, it's worth bearing in mind that this is a personal view of what happened and if, for



example, you're looking for a balanced account of Torvalds's public spat with Minix creator Andrew Tannenbaum, this is not the place to get it. (This you will find in *Open Sources: Voices from the Open Source Revolution*, from O'Reilly press.)

The book concludes with Linus sounding off about subjects dear to his heart, and it's here that *Just For Fun* gets, quite frankly, a bit irritating. The problem is that, while many of Torvalds's sallies against corporate secrecy are well directed, his faith in free markets and consumer power

(suggesting that people simply won't buy things that don't work well) seems astonishingly naïve. Perhaps his second book of memoirs, whenever it may be penned, will have a more rounded perspective. **LXF**

## VERDICT

Well it is Linus on Linux, so you probably want to take a look. However you may be a little disappointed.

## LINUX FORMAT RATING

7/10

From the makers of PC Plus, PC Format, .net, PC Answers and Internetworks

# maxpc

Making the most of your PC

The ultimate PC website. Now with a fresh new look, it's easier to navigate, with speedier downloads and a host of other great features:

- 100s of Reviews covering everything PC
- Expert tutorials to help you get the most from your PC and the internet
- Helpdesk – all your PC questions answered
- Free PC downloads
- Compare prices on PC equipment
- Great offers on books and software

**PLUS the latest news, features, competitions and all Future's PC, internet and Creative magazines on the web.**



Take your PC to the max! **www.maxpc.co.uk**

The screenshot shows the maxpc website with a blue and white color scheme. The top navigation bar includes links for Home, Reviews, Tutorials, Helpdesk, News, Latest download, Business, Compare Prices, Special features, Downloads, Magazines on the web, and Subscription offers. The main content area is divided into several sections, each with a featured article or image. The 'Reviews' section highlights a '512MB PC133' memory module. The 'Tutorials' section features 'Portable advice' on how to download Linux. The 'Helpdesk' section offers a search function for PC problems. The 'News' section includes a 'New Office for 98' article. The 'Latest download' section promotes a 'Voice of the Future' software. The 'Business' section discusses 'The future of the office'. The 'Compare Prices' section lists various PC components. The 'Special features' section mentions 'New PC Hardware'. The 'Downloads' section lists various software packages. The 'Magazines on the web' section promotes 'PC Plus', 'PC Format', and 'Internetworks'. The 'Subscription offers' section provides details on how to subscribe to these magazines.



# Roundup

Every month we compare tons of software, so you don't have to!



## OUR SELECTION AT A GLANCE

- AOL Instant Messenger
- Amsn
- Gabber
- Yahoo Messenger
- GAIM
- Kinkatta
- IMCom
- Everybuddy
- Licq

# Instant messengers

Donning his Conversation Hat™, **Mike Saunders** gets all chatty with a roundup of the best instant messaging software for Linux.

**B**ashing a large wooden club on the cave floor and grunting such things as “woman” and “food” was once the main form of human communication. In those days, single-word utterances and attacking strangers were sufficient ways to get your message across – there simply wasn't very much to say.

Time went on and innovations such as paper, telegrams and SClub7

ringtones added to the rich tapestry of messaging. Now, in the modern world, simply pointing at an object and screaming coarsely won't always achieve the desired result.

Jobs, relationships, cars, money, journeys, friends – with so much going on, everyone wants to keep firmly in touch with matters, and the vast proliferation of mobile phones has helped this. Pretty much anyone can now be reached at any time,

anywhere; whether that's a good or bad thing is for another discussion, but for getting out of hairy situations, phones, pagers and the like can be very handy indeed.

Still, phone calls aren't cheap, telegrams are almost dead, letters are too slow, semaphore isn't hip and telepathy is still being worked on. In the meantime, the Internet has become perhaps the best way to chat at distance, as the adoption

of unmetered access and broadband services has made it fast, inexpensive and convenient. And while email, IRC and Web-based chat rooms have all helped to create quick and simple methods for communicating, instant messaging tools are still the best for instant one-to-one chatting.

## It's good to talk

If you're not familiar with the concept of IM software, it's essentially a real-time discussion system where programs on computers Internet-wide communicate with one another, sending text messages (and optionally other files and data as well). Each user fires up his or her choice of IM client, and can start talking to friends

**‘The Internet has become perhaps the best way to chat at a distance, as the adoption of unmetered access and broadband services has made it inexpensive and convenient’**

## Protocols

Like chat rooms on the Web, IM services tend to be associated with particular subscription services, ISPs and established websites. This has led to an assortment of incompatible protocols used in instant messaging – here's a few of the more notable ones...

■ **AOL** – The American ISP behemoth gives us AIM (AOL Instant Messenger). Not just restricted to users of its dialup service, and members of Netscape's webmail can use it too.

■ **ICQ** – From Mirabilis, a startup based in Israel, "I Seek You" was the first major IM system of any significance. Bought out by AOL in 1998.

■ **MSN** – Microsoft Network Messenger. Enormously popular thanks to the Redmond giant's Hotmail and Passport services.

■ **Yahoo!** – Ties in with their email and other related services. Offers native Linux and FreeBSD versions of their IM software, along with a Java client.

■ **Jabber** – Community-driven open source system based on XML – clients available for all popular operating systems.

on the same service – see the box on protocols, above.

In this way, it's more personal than other chat services such as Yahoo!, and if you just want some general banter with unknowns or help on a particular subject, IRC and Web-based chat systems are more suitable. IM software is the best choice when you know exactly who you want to talk to – you can set up contact lists to monitor someone's online presence, leave messages when they're not around, and (with some services) organise group meetings or "conferences". Other features to look out for include voice chatting, security and privacy, logging and file exchanging.

As can be expected, most IM programs make their appearance on Windows first. However, some popular services have offered proprietary Linux versions of their software, while teams of open source programmers have worked on clients to fill any gaps – possibly the best example of the "scratching an itch" development philosophy. Now there's an excellent range of Linux IM software available for almost all of the popular protocols. So without further ado, let's take a look and see which rules the roost...

# AOL Instant Messenger

From the land of unwanted CDs...

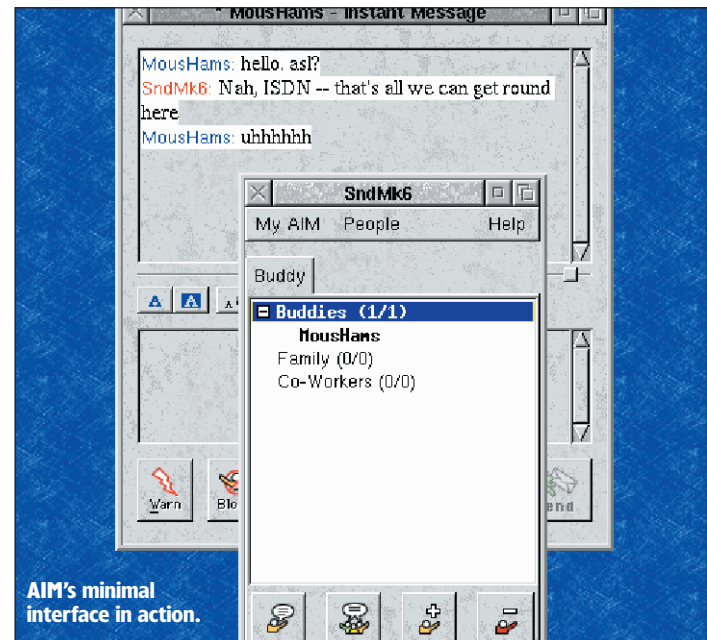
■ **VERSION** 1.5.234  
■ **WEB** [www.aim.com](http://www.aim.com)

### Facing growing competition

from a number of open source AIM-compatible clients, AOL hasn't stood still, and has been developing its own version for Linux. The ISP's position in the world of Free Software is an interesting one – it owns Netscape, and thus many of its developers are working on the *Mozilla* codebase, but it continues legal battles with the free IM software coders.

*AIM Linux* is only available in binary form for Intel compatible boxes, and has been officially tested on Red Hat 6.0, SuSE 6.4 and Mandrake 7.0. Debian and Slackware packages are also available for download, along with a generic tarball (essentially the Slack package with a different installation procedure). The client includes a modified *GTK* (specifically the *GTKtext* component).

When first started, AIM's login prompts for a "screen name." New users are forwarded to a webpage to create an account, before logging in (Oscar only). Aside from the usual large logos, AIM presents a very bare interface *sans* adverts found in the Windows client. Along with a tree list of contacts, there are a few tooltipped buttons – all basic and easy to grasp,



and very much suitable for newcomers.

The chat window permits font changes, emoticons and link insertions, and also includes warning, blocking and info buttons. Sound alerts abound during conversation; sadly, while these can be disabled for different actions, there is no way to change them without manually replacing the WAV files by hand. Other options available in the prefs box include blocking lists, proxies, invisibility and away messages, but no logging or file-transfer functions are supported.

In all, AOL's official offering is a very unvarnished app. Most of the essential features are provided, but little can be customised and it won't satisfy a power-

chatter. Various bugs and glitches need to be ironed-out (the adequate online-help content is ruined by poor-handling of the hardcoded Netscape dependency).

### VERDICT

Features	4/10
Documentation	6/10
Performance	7/10
Stability	5/10

Limited and glitchy – could be of use to Windows converts though.

### LINUX FORMAT RATING

5/10

# Amsn

Talking to the Microsofties...

■ **VERSION** 0.61 ■ **WEB** <http://amsn.sourceforge.net/>

**Microsoft has yet to release an** MSN client for our favourite "un-American", *Pacman*-like OS, so in the meantime, groups of skilled developers the world over have been writing free apps to stimulate competition and provide real innovation. *Amsn*, a continuation of CCMSN, is a TCL/Tk-based lookalike available in a beefy range of languages.

*Amsn's* interface is a quiet, light-

purple box, with all operations being performed through the menus. File transfers, sound alerts and new mail notification are all available, as are group chats and emoticons. Fonts, colours, proxies, text encoding and default mail/browser apps can be changed – but there's little else in the way of config.

Although it could be a simplistic client for Windows refugees thanks to its visual similarities to MS's client, *Amsn* is

still relatively immature when put against its other free siblings. Error messages are unhelpful, the widget set may be off-putting, and the online help is basic.

### VERDICT

Features	4/10
Documentation	5/10
Performance	7/10
Stability	5/10

It shares much of the Windows client's look, but is uncomfortable in use. Help is extremely basic.

### LINUX FORMAT RATING

5/10



# RoundupInstantMessengers

## Gabber

Helping you to Jabber away.

■ **VERSION** 0.8.7 ■ **WEB** <http://gabber.sourceforge.net/>

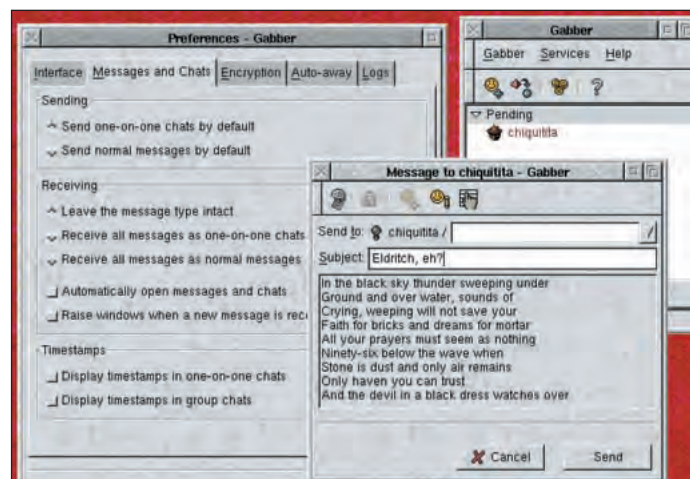
Unlike AOL, MSN, ICQ et al., Jabber was developed from the start as an open source XML-based messaging system. Since the original code was put together in early 1998, the project has acquired sponsorship and now clients exist for all major platforms – the JabberCentral site even lists versions which run through Flash, Java and Mozilla. Current estimates for the userbase stand around a million – not a lot compared to the commercial efforts, but still impressive for a purely word-of-mouth project.

*Gabber*, then, is a GNOME-based Jabber client, released under the GPL. Being developed in C++, it requires the GNOME and GTK libraries to run – but unfortunately the links to the RPMs of these on Gabber's homepage were broken. Compiling from source is the best option right now, providing you have the necessary GNOME

1.2+/*Glade* development files and libraries mentioned above.

On first use, *Gabber* pops up a smart Wizard-style setup dialog which progresses the user through some crucial steps: setting up an account, choosing a server and opting to be included in the general contacts database. Very polished and good to see. A welcome message from Jabber is then ready to be viewed on the small and uncluttered main window.

In use, *Gabber* is intuitive and hassle-free – adding new contacts, joining conference chats (and IRC servers) and updating personal info is a cinch through the well-thought-out dialogs. Support for encryption (through *GnuPG*), spellchecking and logging is available, although there are no native file-transfer methods or sound alerts. The client is reasonably configurable, with message types and timestamping,



Chat window, buddy list and prefs screen busy at work.

XHTML or XML logging (and auto-purging), encryption and interface details all tweakable. The supplied online documentation is lacking, though.

While it won't be the first choice for long-time users of the commercial IM services, the Jabber protocol is strongly suited to developers and users in the open source community, and the *Gabber* client performed very well in our tests. It's robust, fast and pleasant to work with, and has enormous potential for the future. Give it a try.

### VERDICT

Features	7/10
Documentation	5/10
Performance	9/10
Stability	8/10

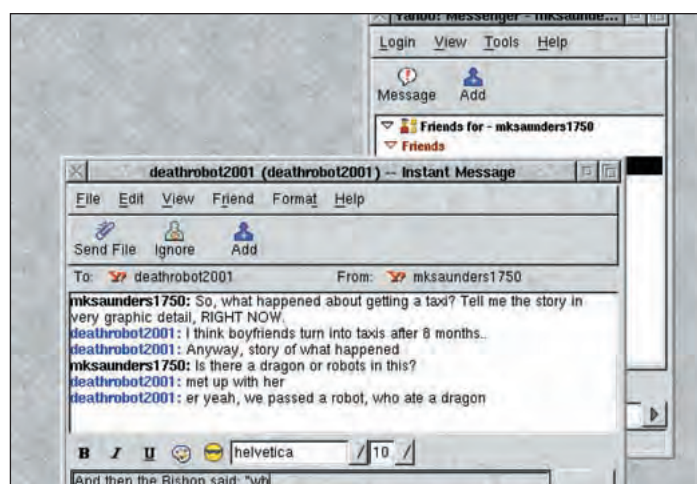
Polished and friendly, *Gabber* is a top way to get involved with the Jabber world.

**LINUX FORMAT RATING**  
 8/10

## Yahoo! Messenger

The! Official! Yahoo! Messenger! Client!

■ **VERSION** 0.99.19 ■ **WEB** <http://messenger.yahoo.com/>



Agony Aunt Mike dishes out yet more CarmoCapsules to a distressed chum.

Few people will be unaware of Yahoo!'s presence on the Web. The enormously popular search engine and portal site offers email accounts, chat rooms, online games and a variety of

other services paid for by mainly by advertising, and it's no surprise that they have an IM system for members as well. Numerous YM-compatible clients have sprung up in the open source world

through reverse engineering, but here we're looking at the official version.

Unix clients are available in binary form for Linux and FreeBSD (good to see the latter supported, seeing as most of Yahoo! runs on that OS). The latest release is a 580K download – this initially seems quite big for a simplistic client, but it includes a modified version of the *GTKHTML* library. Desktop links for KDE and GNOME are provided, and the SuSE, Red Hat and Mandrake RPMs install in /opt with a symlink to the binary in /usr/bin. A Debian package is also available.

YM's easy-going *GTK* interface is pleasingly short of pointless visual frills; the main window contains a couple of icons for adding friends and starting new chats, with the lower pane listing currently-online friends. Helpfully, YM's initial login screen has a button for creating a new account – simply by pointing your chosen browser at the relevant Yahoo! webpage.

As far as features go, Y!'s Linux client is rather lacking when compared with its Windows counterparts. Conferencing and file exchange are present, as are varying fonts and emoticons, along with a "buzz" command which sends your co-

chatter's window into an alarming fit.

The *Netscape*-esque Preferences box sports a decent array of options to tweak: fonts, window popups, keyboard focus, sound alerts, proxies, ignore lists and logging can all be altered. Sadly, no documentation is provided with the client package itself, so clicking the Help button to fathom out any settings requires that you're online at the time.

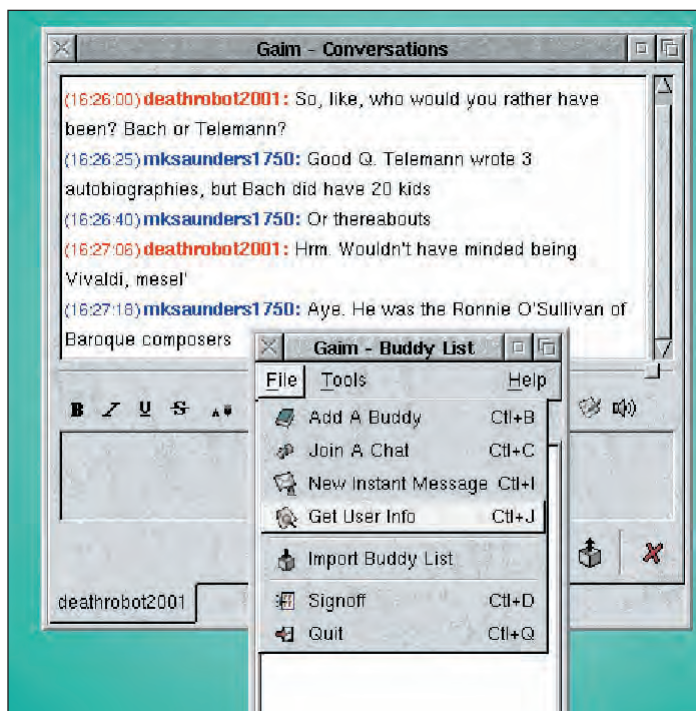
Ultimately, YM is an adequate Linux client with some friendly touches. At the same time, it's somewhat short on the feature front and generated far too many error dialogs and messages for comfortable use on our test machines. Hopefully by 1.0 it'll be more polished.

### VERDICT

Features	6/10
Documentation	3/10
Performance	8/10
Stability	4/10

Does the job, but needs work on the feature set and stability. Simple to use – but lacking documentation.

**LINUX FORMAT RATING**  
 6/10



GAIM's main buddy list window and a highly cerebral conversation.

# GAIM

Multiple protocol support from AIM clone.

■ **VERSION** 0.591 ■ **WEB** <http://gaim.sourceforge.net/>

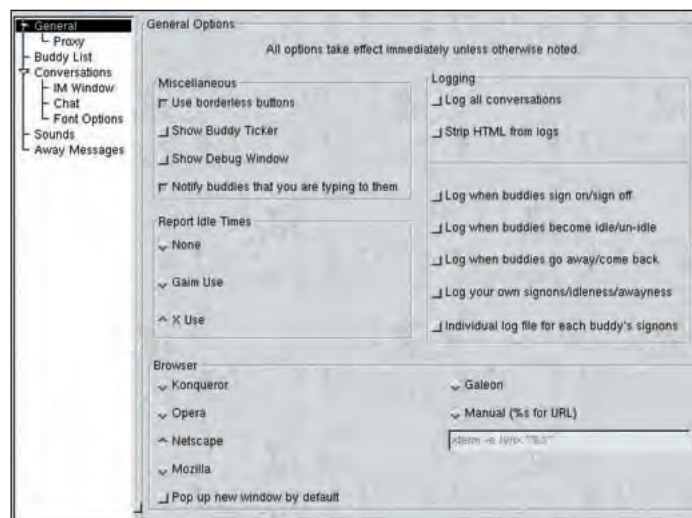
## As the name suggests, GAIM

started out as a GTK-based implementation of AOL Instant Messenger. Beginning its life in 1998, original lead coder Mark Spencer originally used the project as a way to learn programming with the GTK toolkit; additionally, he wanted to keep in touch with his AIM-using chums, and writing GAIM appeared to be a perfect two-bird one-stone kill scenario.

In its current development stage, though, GAIM is worthy of the most attention in this roundup for one simple reason: multiple protocol support. While most of the other

clients on test here focus on one particular IM service, GAIM supports a hugely impressive range of protocols – AIM, MSN, IRC, Jabber, Yahoo!, Napster and Zephyr – all through a well-crafted plug-in architecture. Naturally, AIM (TOC and Oscar) is still the default system.

Almost all distros include a version of GAIM, but compiling from source is generally straightforward – also, building from the tarball provides options to statically-include your choice of supported protocols if you'd rather avoid plug-ins. According to the developers, the 0.5x releases will be the last to use



Options aplenty in GAIM's packed Preferences box.

GTK1.2; from 0.6 onwards, it'll be designed around GTK 2.

## Name of the GAIM

When first started, GAIM's main window acts as a hub for the configuration steps which need to take place: first of all, the required plug-ins need to be loaded to support your chosen protocols. All this is achieved through a hassle-free dialog system, and each plug-in offers some help text describing its usage. In the default installation, plug-ins are available for ICQ, IRC, Jabber, MSN, Napster, Yahoo!, Zephyr and Gadu-Gadu, with some extras to implement spellchecking, auto-reconnection and visual-notification.

Multiple accounts for different services are catered for, along with options for automatically logging-in, alerting when new mail has arrived (if the service also does email), and aliases. Again, accounts are managed through the pleasingly easy dialogs – GAIM is ideal for Linux virgins who've only used the Windows and Mac clients.

GAIM's tree-structured options window contains a vast array of settings to please even the most

demanding chatter – appearance, fonts, window dimensions, logging, key controls, sounds and away-messages can all be tweaked aplenty, although tooltips for some of the more esoteric options would be a welcome inclusion in future releases.

Other features which stand out are tabbed-chatting (supremely useful for saving on screen space), encryption (only between GAIM clients at the moment) and username tab-completion. Also of note is the "Buddy Pounce" option, which provides a small amount of automation – you can set a sound to be played, message to be sent or even command to be executed when a friend appears online or returns from being away.

Overall, GAIM is undoubtedly the most complete and versatile IM client on show here. During our testing with several different protocols, we didn't encounter any stability issues and the friendly front-end makes it a joy to use. A couple of things are still missing – a file-transfer system being the most prominent – but the general quality and effort put into GAIM makes it our pick of the bunch. Highly recommended.

## Connie says: "You've got legal threats!"

When you're reverse-engineering protocols and writing Free Software to interoperate with a large company's products, life is never going to be easy. The GAIM coders have had a few run-ins with AOL's legal bods – and when a bunch of spare-time hackers find themselves faced with razor-blade-eating lawyers, it's no surprise that they have to back down occasionally.

The first event occurred in July 1999: unhappy with the use of their logos and trademarks, AOL contacted the GAIM crew and asked them to make some changes.

These were quickly followed through; two years later, though, AOL sent out some stern letters to various open source IM client coders – they preferred that such programs avoided using similar names to the commercial one. KDE client

KAIM promptly renamed itself *Kinkatta*, and a few others followed suit.

But GAIM decided to stand up for itself, and the project started gathering donations to prepare for a legal battle against the ISP – apparently receiving more than three times the original cash goal. The battle continues from here, so keep an eye on the project's homepage for updates.

## VERDICT

Features	9/10
Documentation	7/10
Performance	8/10
Stability	9/10

Fast, flexible, fun and free – God's gift to the IM client world.

**LINUX FORMAT RATING**  

 9/10



# RoundupInstantMessengers

## Kinkatta

The former outlaw comes good.

■ **VERSION** 1.01 ■ **WEB** <http://kinkatta.sourceforge.net/>

**Developed in response to problems** with *GAIM*'s early releases, *Kinkatta* has become a fully-fledged AIM (TOC and Oscar) client for KDE/Qt. Its original name was *KAIM*, but AOL's lawyers soon put an end to that. The current stable tree, which we compiled, is built around Qt 2.x, but a CVS branch exists for those wishing to try and newer Qt 3 version.

*Kinkatta*'s UI is attractive, with the usual tree view of friends in the main window and separate chat boxes. Logging, pounces (automatic operations when someone comes online), emoticons and chat rooms are supported, and the interface is easy to navigate. There's a plethora of changeables in the prefs screen – all toolbars can be reconfigured, sounds modified, fonts and coloured altered and much more.

In all, *Kinkatta* is a solid AIM client with some nice visual touches. It's satisfying to use and does the job well, but could do with a few more features and proper online help. With KDE 3.x being rolled out on *en masse* now, we hope to see a full release for that desktop very soon.

### VERDICT

Features	8/10
Documentation	4/10
Performance	8/10
Stability	9/10

A commendable AIM client for KDE.

**LINUX FORMAT RATING**  
 **8/10**

## IMCom

Liberate yourself from the GUI.

■ **VERSION** 0.95 ■ **WEB** <http://imcom.floobin.cx/>

Our roundup this month has been dominated by graphical clients, so it's only fair that we should check out a text-based tool for those on older hardware or running over a *Telnet/SSH* session. *IMCom* is a maturing Jabber client written in Python (2.x), with an IRC-style command interface, and requires the *pyexpat* module (should be installed with Python or in the *PyXML* package).

Supporting conferencing, file transfers, secure SSL connections and multiple profiles, *IMCom* is an accomplished client which stands well amongst the graphical competition. Operating the program will be fairly straightforward to IRC-heads – /MSG, /AWAY, /INFO and the like are used along with many other well-documented commands.

Hardly accessible to the first-time chatter, but experienced Jabberians will find a stable and unintrusive client that's ideal for use on a shell account. With a healthy range of supported features and ability to run almost anywhere, it's worth keeping around.

### VERDICT

Features	7/10
Documentation	7/10
Performance	8/10
Stability	9/10

Small and speedy, but not the easiest client to get started with.

**LINUX FORMAT RATING**  
 **7/10**

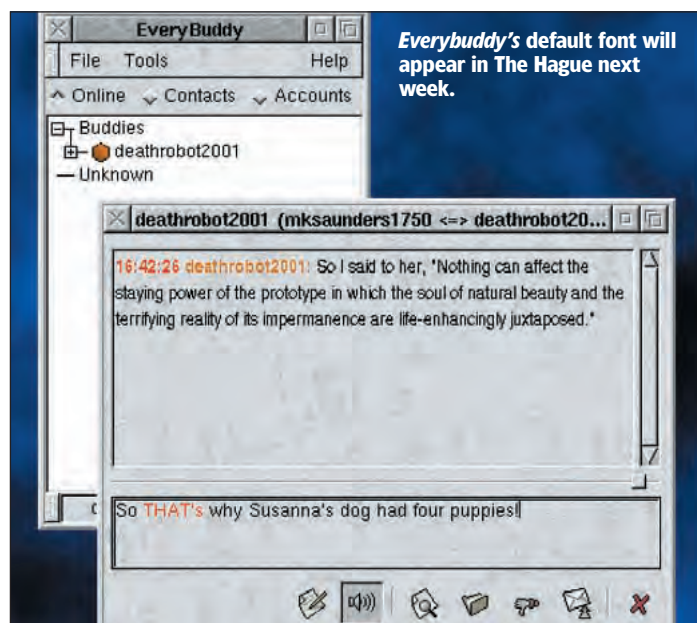
## Everybuddy

Popular, multi-protocol client.

■ **VERSION** 0.4.3 ■ **WEB** [www.everybuddy.com](http://www.everybuddy.com)

**Like GAIM, Everybuddy attempts** to be a one-size fits-all client which supports a number of protocols. And although the authors claim that it's still

in an early stage of development, it already supports an impressive number of services – AIM, MSN, Yahoo!, ICQ, Jabber and IRC – and



sports some neat features. It's built around the *GTK* toolkit, and is GPL'd.

The project's site offers RPMs of both a regular CVS snapshot and the stable release, and there's even a .deb for *Fink* users on MacOS X. We opted to build from source on our main test box (a stock RH 7.3 installation) – this went smoothly enough, and with the source tarball being a couple of megs smaller than the RPM, it's quicker to grab from the 'Net.

### Interface

*Everybuddy*'s minimal interface is peppered with a few icons and is generally easy to understand. The account editor is slightly more intuitive than *GAIM*'s; protocol modules don't need to be loaded as the service can be selected from a drop-down list, although new-account creation isn't present.

Together with the range of supported protocols, *Everybuddy* also offers file transferring, sound alerts, importing of *GAIM* and *LICQ* contact lists, logging and tabbed chatting. One innovation is the use of *Babelfish* for on-the-fly translation – the accepted deficiencies of instant translation methods aside, it could still prove to be a handy feature for those chatting with foreign friends.

Almost every aspect of the client can be tuned and modified through the tabbed preferences window: sound events and files, logging (and HTML stripping), colours, fonts, spellchecking and additional modules can all be altered, along with instant character set re-encoding. The online help link points to a well-written website which explains the client thoroughly with the help of screenshots.

As it stands, *Everybuddy* is a worthy competitor to *GAIM* and will please all but the most fickle chatters. With support for the popular protocols, a decent range of options and some adventurous features like the *Babelfish* translation, it's a close runner-up in our tests. Right now, it just needs some more work on the reliability and ease-of-use fronts.

### VERDICT

Features	9/10
Documentation	10/10
Performance	7/10
Stability	7/10

A versatile client with plenty of neat features – well worth a look.

**LINUX FORMAT RATING**  
 **8/10**

# Licq

The best-known ICQ client for Linux.

■ **VERSION** 1.2.0a ■ **WEB** [www.licq.org](http://www.licq.org)

**ICQ was the first notably** successful IM system, and has proved to be particularly popular in Asian countries. Various Linux clients exist, but *Licq* is the best known – and for good reason. Not only can it use plug-

ins to provide different interfaces (Qt, GTK+ and text), but the default Qt is also skinnable.

We built the latest release from source, along with the Qt front-end. *Licq*'s default skin looks a tad plain

compared to its new-fangled GNOME and KDE friends – there's little in the way of icons or other eye-candy. Random group chats (*i.e.* games, romance etc.), sound events and multiple away messages are supported, along with the usual font, proxy server and UI options.

If you only need ICQ and no other service, *Licq* is a decent choice and the UI plug-ins will help it fit in, whatever your desktop. It's adequately configurable and very stable, so for no-nonsense chatting, give it a go.

## VERDICT

Features	5/10
Documentation	7/10
Performance	8/10
Stability	8/10

Plain and simple, and does the job for basic ICQing. Can be built with Qt or GTK interfaces,

## LINUX FORMAT RATING

7/10

# INSTANT MESSENGERS THE VERDICT

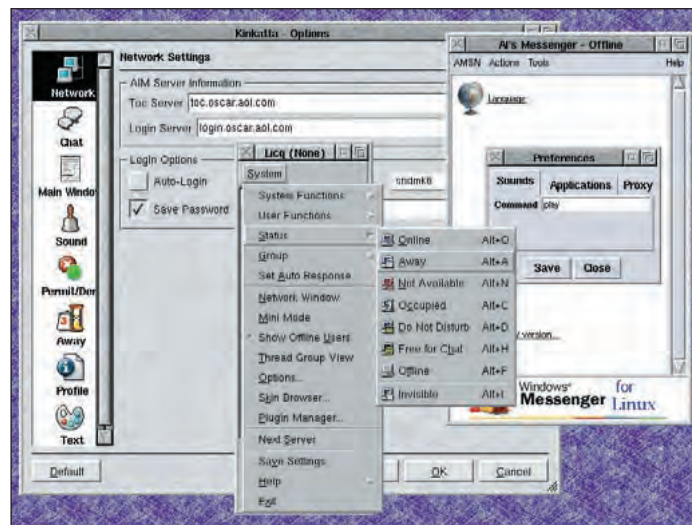
So, what's the word on the street, IM-client-wise? One of the most interesting findings from looking at the software here is that the commercial ad proprietary choices lag way behind their Free Software alternatives. A common gripe of end-users in the last few years has been with the interfaces of open source apps – typically, commercial vendors put a little more polish into their front-ends.

Not here, though. The level of spit-shine in clients like *GAIM* and *Gabber* go far beyond the limited and glitchy interfaces that AIM and Yahoo! present in their clients – the proprietary tools appear to be a quick way to keep the small (but vocal) Linux userbase happy temporarily. Of course, the advantage of the commercial programs is protocol support – should the system change, they'll be updated quickly while the Free Software developers spend their time reverse-engineering.

Right now, *GAIM* and *Everybuddy* are clearly the best choices for users dependent upon multiple IM services. Being able to communicate with people on different networks from the same tool is invaluable – both in terms of consistency and saving RAM and screenspace. *GAIM* is our number one choice thanks to its superb protocol support, excellent interface, ease of configuration and general stability.

For Jabber users, though, *Gabber* puts up a good fight – the attention paid to newcomers with the wizard-esque interface deserves much applause – and it maintains its friendly and graphical style without becoming "facehuggerware." The others we've looked at are worth considering too if they suit your chosen protocol, but for the time being, *GAIM* is the bees' knees-laden, cat's pyjamas.

Finally, what happens from here? Most crucially, the hackers behind the free applications will need to



*Kinkatta, Licq and Amsn* strutting their stuff, yesterday.

keep on top of protocol changes – the large commercial services can change the back-end workings on a whim (not just to break compatibility with Free Software clients, but also in battle with each other), and push

out updates to their customers. With this in mind, we hope to see the completely open Jabber network continue to grow in popularity, and the Linux clients continue to improve. **LXF**

## Table of features

Our main test box for the IM clients was a 800MHz PIII PC with 320MB RAM, running Red Hat 7.3. Note: the package sizes and memory statistics below should only be used as a rough guide. Memory usage varies from system to system, and is affected by the program's configuration and various other factors.

Name	License	Protocol	Interface	Pkg size	Avg mem use	Logging	File transfer	Encryption	Alternatives
AOL IM	Proprietary	AIM	GTK	910k	6M	No	No	No	Kinkatta, GAIM
Gabber	GPL	Jabber	GTK	1.7M	8M	Yes	No	Yes	IMCom, Jarl, Pybber
GAIM	GPL	See review	GTK	2.2M	4M	Yes	No	Yes	Everybuddy, Imici
Yahoo!	Proprietary	Yahoo!	GTK	583k	5M	Semi (hist)	Yes	No	Curphoo, GTKYahoo
Everybuddy	GPL	See review	GTK	1.1M	4M	Yes	Yes	No	GAIM, Imici
Amsn	GPL	MSN	TCL/TK	165k	3M	Yes	Yes	No	Centericq, Gtkmsn
IMCom	Distributable	Jabber	Text	240k	4M	Yes	Yes	Yes	Jabber.el, Sjabber
Kinkatta	GPL/LGPL	AIM	Qt 2.x	976k	9M	Yes	No	No	AIM, nTAIM, TiK
Licq	GPL	ICQ	Qt/others	1.7M	6M	Semi	Yes	No	GICQ, Kicq



# HotPicks

The best new open source software on the planet!



**Richard Drummond**

As well as writing our Java series, Rich finds time to try new Linux apps.

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 which are released or updated, and select the newest, most inventive and best for your perusal. Most of the Hot Picks are available on our coverdiscs, but we've provided web links if you want to make sure you have the very latest version.

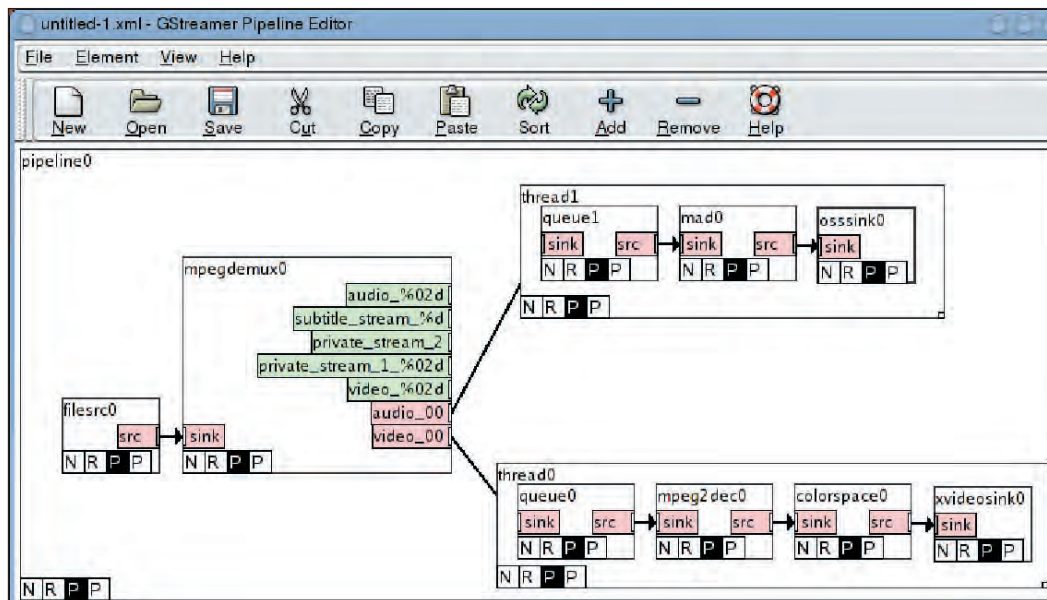
If you have any suggestions for things that we should cover, email us at [linuxformat@futurenet.co.uk](mailto:linuxformat@futurenet.co.uk)

## HotPicks at a glance

GStreamer	46
Anjuta	47
libtrash	48
Disk ARchiver	48
Synaptic	49
Scribus	50

## HOTPICKS AWARD

Everything covered in our Hot Picks section is unmissable, but every month we'll be singling out one project for outstanding brilliance. Only the very best will be chosen!



**GST-Editor** lets you build arbitrary multimedia processing pipelines and serialise them as XML.

## MULTIMEDIA FRAMEWORK

# GStreamer

■ VERSION 0.4.2 ■ WEB [www.gstreamer.net](http://www.gstreamer.net)



**G**Streamer is a bold and beautiful piece of programming. The aim of the project is to create a framework that allows arbitrary media-handling components to be hooked up as a pipeline. It successfully applies the ideas of component-based development to the multimedia world. Using *GStreamer* developer can build any kind of media processing into their applications, by employing a pipeline of re-usable components.

*GStreamer* is based on the objected-oriented foundations of the *GLib* library and most of its functionality is created via plug-ins. The core library aims for a small footprint and high performance.

Plug-ins are divided into classes. There are sources and sinks, which are the beginning and end points of your multimedia streams, respectively. Other classes – codecs, multiplexers, demultiplexers, and filters – operate

on streams between sources and sinks. Support for threading, scheduling and synchronisation is built-in.

Any file – including any that can be reached via a transport method supported by the GNOME virtual filesystem – can be used as a stream source, while OSS or ALSA soundcards can be the source of an audio stream, and a video4linux capture device the source of video stream. Codecs are supplied for decoding MPEG, Ogg Vorbis and FLAC streams, while demultiplexing of AVI and *Quicktime* streams is supported. Flash decoding is implemented via the open-source *libswf* library. A wide variety of output devices are also supported, including files, X and SDL for video streams, and an ALSA or OSS sound-card or the *Arts* and *ESD* sound daemons for audio streams. Dozens more plug-ins are provided, including audio effects such as reverberation, and visualisation effects.

The number of projects using *GStreamer* is currently small, but serves to give a taste of what *GStreamer* is capable of. The *GST-Player*, bundled with *GStreamer*, is a simple media player that supports playing any kind of media for which a plug-in is available. Independent projects include a DVD viewer and an audio mixing application, while work is underway to build non-linear audio and video editing suite on top of *GStreamer*.

*GStreamer* also includes a number of developer and debugging tools. *GST-Editor* is designed to let developers build *GStreamer* pipelines by visually hooking up components on a canvas. Even if you are not a developer, if you enjoyed Lego as a kid, then I'm sure you'll enjoy tinkering with it. But it has a serious and very practical side, too. *GStreamer* includes the ability to serialise pipelines as XML, and pipelines can be designed in *GST-Editor*, saved out as an XML file, and then loaded into your applications at run-time to build the pipeline. This really underlines the power and flexibility of the system. From a developer's point of view, the ability to include any kind of multimedia processing into an application by doing little more than piecing the desired components together by point-and-click is little short of amazing.

## C/C++ IDE Anjuta

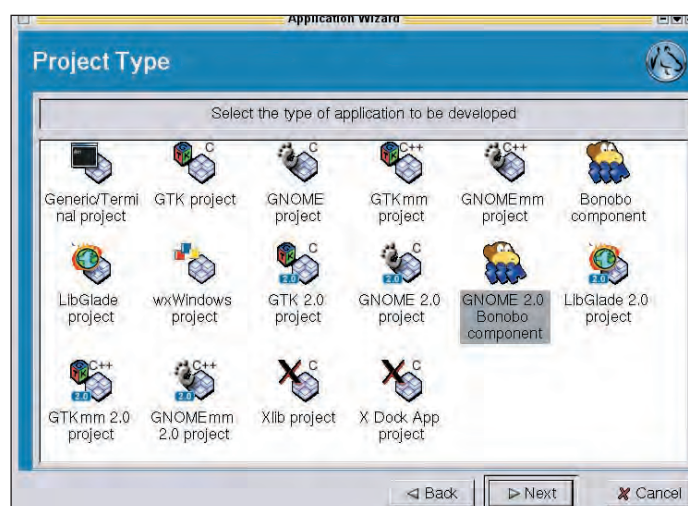
■ **VERSION** 1.0.0 ■ **WEB** <http://anjuta.sourceforge.net/>

**D**eveloping software on Linux doesn't have to be hard work. Of course a text editor such as *Vi* or *Emacs* and the GNU tool chain is all you really need, but these days people expect a more comfortable environment to work in. *Anjuta* is one of the many graphical IDEs available for Linux that can provide this.

*Anjuta* is C/C++ IDE, and is built on GNOME. Version 1.0 builds against GNOME 1.4, but work is progressing on a port to GNOME 2.0. You can use *Anjuta* to build projects for either release of GNOME, however. *Anjuta* isn't really a visual IDE, but it can integrate with *Glade*, the GNOME/GTK+ GUI editor. *Anjuta's* 'New project' wizard can generate the skeleton of a GNOME or GTK+ project for you, supporting development in either C, C++ or a mixture of the two. This wizard also give options to generate a selection of other types of modules and programs, including pure *Xlib* projects, *wxWindows* projects and Bonobo components. No support for Qt or KDE is included, though.

The main window of *Anjuta* will be familiar to anybody who has used an IDE before. It is split into three areas. On the left you the Project Window, a tab pane containing the project browser, which lists the files in your project; the symbols browser, which lists classes, functions, variables, etc. in your source code; and the file manger. At the bottom is another tab pane, the Message Window. This contains various output logs, such as output from the build process or debugging output, and also features a shell terminal. The rest of the main window is given over to the editor. The Project and Message Windows can be hidden on or torn-off as floating windows if you want to dedicate more space to the editor.

*Anjuta's* editor is pleasant to use. It has all the usual niceties, such as syntax highlighting, code-folding, auto-formatting, and auto-completion. The editor offers some configurability, but doesn't include editor 'personalities' like some do. There doesn't seem to be any way to modify the keyboard mapping, and tab control is basic.



The New Project wizard can generate skeleton projects of various types.

Auto-completion works via the ctags system. *Anjuta* transparently uses ctags to build a database of identifiers in your code. Hitting **Ctrl+Return** will try to complete the current keyword or identifier and pop-up a list of possible completions. The tags database is also used in the symbols list in the Project Window. This works well in the main, but gets confused if you over-use preprocessor macros. Although it understands them enough to be able to list the macros you have defined, it won't expand them when used in function or variable declarations. This rather cuts down on its usefulness.

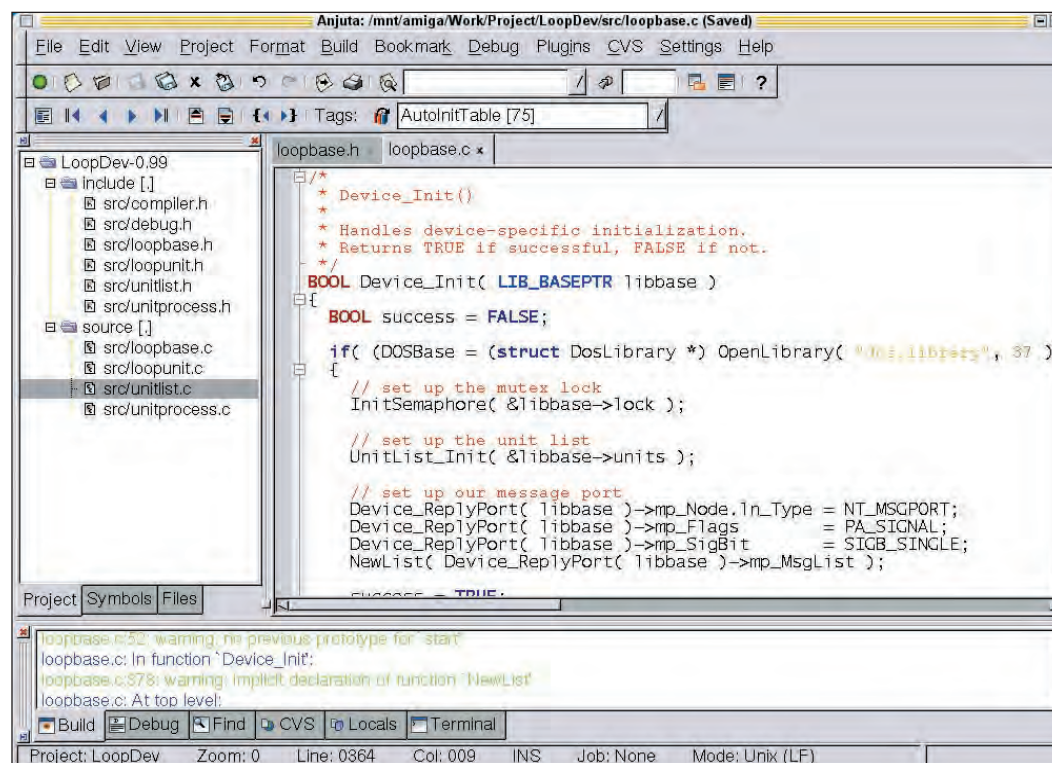
*Anjuta's* auto-completion system

cuts down on typos, but is otherwise fairly basic. It just list words. It can't show any documentation associated with a function or hint at a functions parameters. *Anjuta* can use the *DevHelp* system for referencing documentation though. Simply highlight a word in the editor, select Context Help, and *DevHelp* will be called to look up the word. Handy.

### Auto pilot

When it comes to building and debugging, *Anjuta* supports the usual GNU tools such as the *autotools*, *make*, *gcc* and *gdb*. If you chose to generate your project base with the project wizard, then *Anjuta* provides all the glue and scripts you need. Otherwise, you can use *Anjuta* to generate a configure script, using the project options to control the process, or you can manually roll your own Makefile. *Anjuta* really only works well if your project has a single target, though, and there isn't any support for multi-level *make* scripts. Various compile time settings can be configured, such as the paths to the compiler and linker, library and include paths, compilation options, etc., but annoyingly these are saved out as a user session rather than associated with a particular project.

*Anjuta* is fairly simple IDE, but this could be any advantage if you simple needs. It's a very easy program to pick up and learn, so will be great for beginners or those that just need to get hacking with a minimum of fuss. Attention to detail is lacking in some areas, though, and stability can be a problem if you try to make it do things it doesn't like. It also has the annoying habit of forgetting your settings and preferences, rather than saving them to disk.



*Anjuta's* main window makes good use of space and is easy to navigate.



## LIFE-SAVER

## libtrash

■ **VERSION** 1.5 ■ **WEB** [www.m-arriaga.net/software/libtrash](http://www.m-arriaga.net/software/libtrash)

**W**e've all made the mistake, at one time or another I suspect, of accidentally deleting files that we didn't intend to – and hours, days or even weeks of work were made to vanish in the blink of drive activity LED. Of course, the safest solution is to always make back-ups, but a neat way of avoiding such erroneous destructiveness is effected by *libtrash*.

*libtrash* creates a transparent filesystem trashcan by hijacking

certain functions of the standard C library, namely those functions that can delete files. Instead of blasting deleted files into oblivion, *libtrash* moves them into a sub-directory of your home directory called Trash. This process is completely invisible to the user and works with all software.

The contents of the folder Trash reflects the hierarchy of your home directory. So, deleting ~/foo/bar.txt would move the file to ~/Trash/foo/bar.txt. What's more

*libtrash* automatically generates unique filenames when collisions occur in the trashcan. *libtrash* has yet more tricks up its sleeve. It can be configured to ignore certain types of file by filename extension – say, log files, or back-ups – it can be instructed to ignore files on removable media, and it includes a script to automatically clean the oldest files from your trashcan when it reaches a certain specified size.

Installing *libtrash* is fairly straightforward. The library needs to be preloaded, and this can be done globally, if you have root access, by adding **libtrash.so** to the /etc/ld.so.preload config file. Alternatively you can define the variable **LD\_PRELOAD** in your

environment to load *libtrash* and so enable *libtrash* on a per-user basis. The **LD\_PRELOAD** method has the disadvantage that if you *su* into another user account which doesn't have *libtrash* set up, without preserving your environment, then *libtrash* will be disabled. You can also turn *libtrash* temporarily off and on without totally disabling it by setting the variable **TRASH\_OFF** to **YES** or **NO**, respectively. *libtrash* can be configured to display a warning message when you try to delete files and it has been turned off in this manner.

*libtrash* is an ingenious and effective solution to potential data loss and goes a long way to proving the rule that the simplest ideas are the best.

## BACK-UP TOOL

## Disk ARchive

■ **VERSION** 1.2.0 ■ **WEB** <http://dar.linux.free.fr/>

**T**alking of back-ups, the reason many of us find backing up our precious data such an onerous task is that we haven't got the tools to do the job properly. Wouldn't it be nice to have a back-up medium capacious enough to store all of your data without endless swapping of cartridges or disks? If you have to resort to save to Zips disks or CD-Rs, then *Disk ARchive*, or *dar*, will help you get the job done with a minimum of fuss.

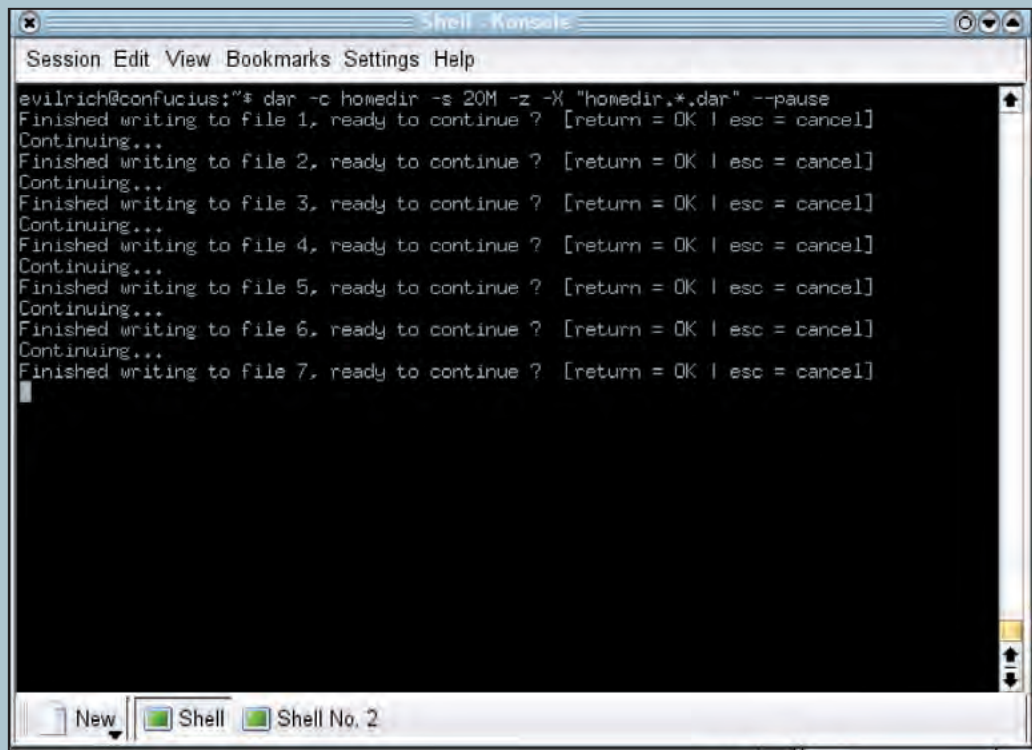
As its name sounds, *dar* is very much an equivalent to the standard *tar* command which was originally meant to create tape archives. The difference with *dar* is that it can split the archive it creates into slices of a specified size as it goes, optionally pausing or running a user-defined command between slices to allow you to mount a new disk or burn a slice to CD-R. Like GNU *tar*, *dar* supports incremental back-ups and compression (although with *dar* compression is internal to the archive and currently only *gzip* is supported), but it offers many other features that *tar* doesn't.

*dar* includes support for backing-up extended filesystem

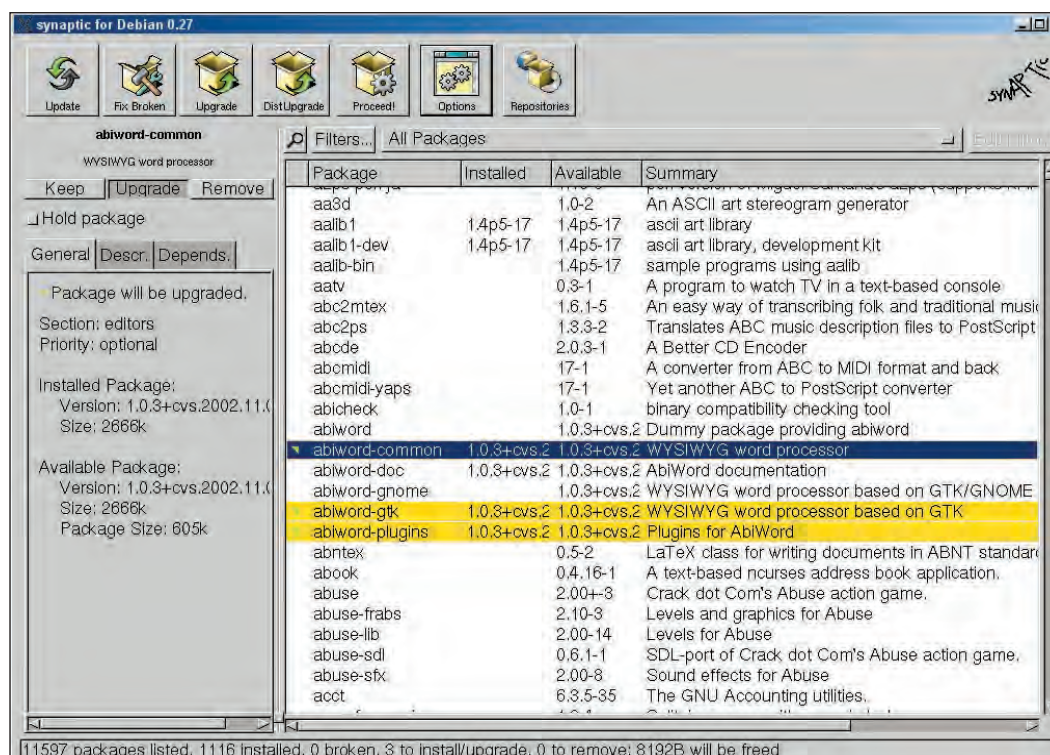
attributes, a simple password-based scrambling method for protecting archives, and the ability to specify which files are to be compressed.

*dar* is optimised for back-up to disk media. Thus a *dar* archive includes a catalogue which lists all the files in the archive, and listing the catalogue only requires access to the first and last slices in an archive. This allows a particular file to be located directly when restoring, without have to sequentially look through all the slices in turn. Similarly, the catalogue speeds up access to the reference archive when doing an incremental backup. Also, *dar*

includes a command for managing archive catalogues, called *dar-manager*. *dar-manager* can extract catalogues from specified archives and store them in a database. You can then use this database to quickly locate a particular file in any of the archives in the database, a handy feature when you have a series of incremental archives and you don't know in which archive the file is stored.



**Dar** can split a back-up archive into fixed-size slices as it creates it.



**Synaptic's uncluttered layout is much easier for newcomers to navigate than the bewildering *dselect*.**

## APT FRONT-END

# Synaptic

■ VERSION 0.27 ■ WEB [www.nongnu.org/synaptic](http://www.nongnu.org/synaptic)

**W**e have often sung the praises of Debian's *APT* in this magazine, and rightly so. *APT* stands for *Advanced Package Tool* and it is the powerful system that Debian and some other distros use for managing the installation and updating of software. What is less praiseworthy about *APT* is its user interface. Traditionally, Debian has offered the console-based *dselect* as a front-end to *APT*, and this piece of software is probably the reason why many are put off using Debian at all. While it is direct and quick for the seasoned hand to use, its complexity is overwhelming to the newcomer. Even *dselect*'s most hardened fans will tell you that it is a fairly hideous piece of user interface design.

When Conectiva adopted *APT* for package management in their RPM-based distro, they were quick to realise that simply providing *dselect* alone to control *APT* simply wouldn't do. So the *Synaptic* project (originally called *Raptor*) was born. *Synaptic* is a fully-fledged graphical front-end to *APT* and cunningly blends ease-of-use and flexibility. Its

clean layout and tool-tip help make it simple for novices to get grips with, but its boasts some powerful features, such as filters, for the more advanced user.

## An apt update

Earlier releases were based on the *WindowMaker* desktop's *Wings* toolkit and I was never that keen on its look and feel. A simple change of toolkit, and *Synaptic* now looks a lot cleaner and more professional and fits in better with the look and feel of most users' desktops. Since version 0.25, the *GTK+* version is the default, but you can still build against *Wings* if you prefer.

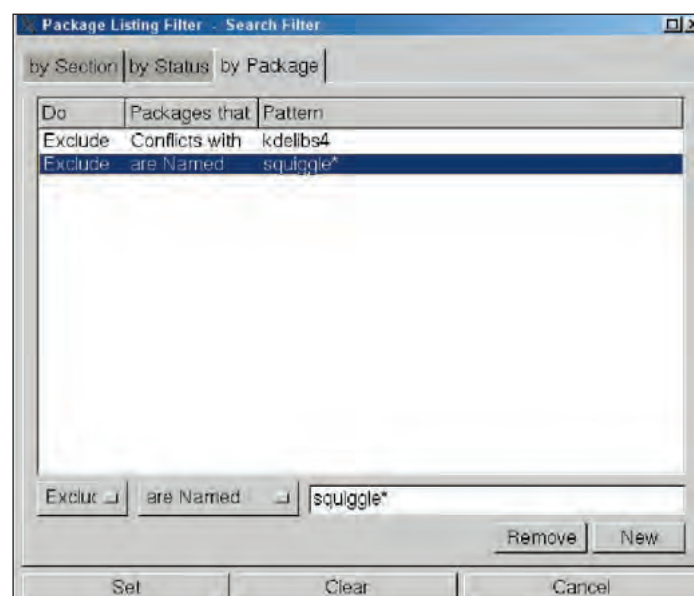
For those that haven't experienced *Synaptic* before, it presents a simple main window, the main part of which is the package list. To the left is an info area which gives tab panes showing brief and detailed descriptions of the currently-selected package, and a list of its dependencies. Just above this are a set of buttons to mark this package for installation or removal, to revert it to its previous installation status or to mark it as 'held' (this last means that it won't

be updated even if a newer version becomes available). At the top of the main window is a button strip which launches the main *APT* tasks, such as updating the package list, fetching and installing updated packages, modifying settings, and managing the repository list.

The main window shows a flat rather than hierarchical list of packages, sorted alphabetically by name. By default, this lists all of the packages that *APT* knows about, both installed and not installed, but the contents of the list can be controlled

and made more manageable by filtering. It can be filtered to show only those packages that are installed, only those that are not installed, only packages with updates available, and so on. As well as these built-in filters, you can define your own filters and save them for later re-use. User-defined filters can filter on any combination of package status (that is, whether installed, not-installed, upgradeable, held, etc.), package section, and pattern or regular expression matching on various package fields such as its name, description, dependencies, etc. A new feature (in combination with Debian's *deborephan* command) is filtering those packages that are orphaned, that is, those that no other package depends on. This is a useful feature for weeding out obsolete or unused software.

*Synaptic*'s filtering ability is probably its main selling point and really comes into its own when you are wrestling with a package list of over 10,000 packages. The package list, being flat, is easier to navigate, the filtering controls are more advanced than any rival *APT* front-end. The interface design still needs a few tweaks – such as reducing the number of button-clicks required to perform certain tasks. *E.g.*, in any of the dialogs which can be popped up. In the dialog for editing filters or the dialog for modifying preferences – you need to perform your changes, select 'Save' to make them take effect and then select 'Close' to quit the dialog. But with the move to *GTK+*, *Synaptic* is now much more pleasant to use and just feels slicker all round.



**The ability to define and save package filters simplifies the management of large package lists.**



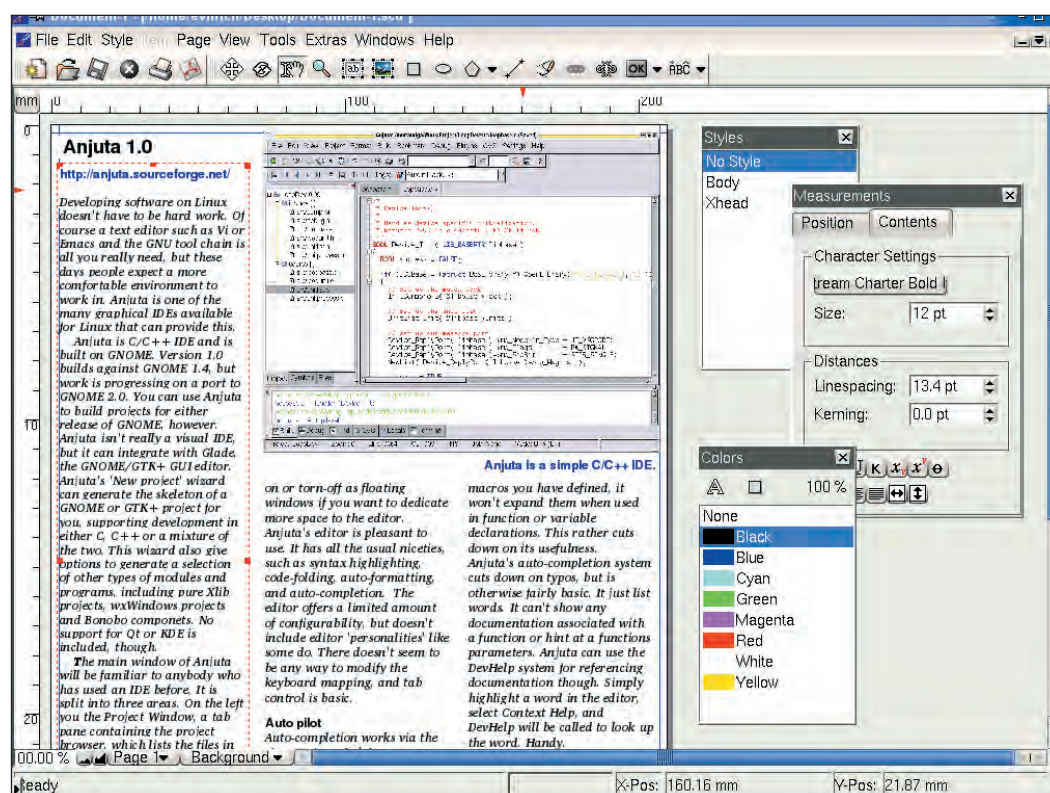
## DESKTOP PUBLISHING

## Scribus

- **VERSION** 0.9.2
- **WEB** <http://web2.altmuehlnet.de/fschmid/>

Unix and desktop publishing are not two concepts you often see side-by-side. Linux boasts many powerful open source office suites, but until recently there was nothing to rival commercial packages such as *Quark Express* for document layout. (Of course, there's *TeX*, the typesetting system beloved by the academic world, but I can't really see that catching on in the design shops.) *Scribus* is a project to create a free desktop publishing app based on the Qt toolkit. It's been around for a while, but recent releases have begun to add enough functionality to make it useful for real work. Version 0.9.2 is the current developer release and is definitely not stable enough for a production environment.

*Scribus* follows the standard DTP app model, using a frame-based page layout. Frames can contain text or graphics, and you can import plain ASCII text into a text box and a number of bitmapped picture formats or EPS images into a picture box. It includes a selection of basic drawing tools, such as rectangles, lines and bezier curves. Font support is good, with typographic controls to set kerning, line-spacing and so on, although Type 1 fonts only are supported. If you have used *Quark* before, then



**Scribus compares favourably against commercial DTP packages in terms of features.**

*Scribus* will be very familiar, since its interface bears more than a passing resemblance to that commercial heavyweight. Similar to *Quark*, you have a toolbar of main drawing tools, and extra windows can be popped up containing tools and settings which will modify the current object. Thus there's the Measurements window, which lets you perform fine-tuned adjustment to the position, size and contents of the selected frame. The Colours window lets you set the foreground and background

colours, and the Styles window lets you apply pre-defined paragraph styles. What works less well in *Scribus* is the placement of these various windows. You can tear off the main tool bar as a vertical strip, which makes the interface even more *Quark*-like, but if you place it too close to the top of the main window it gets docked back into the main window. Similarly, the various ancillary tool windows cannot be placed too close to the right edge of the window or they will be docked there. You really need more control to be able to maximise your screen real estate.

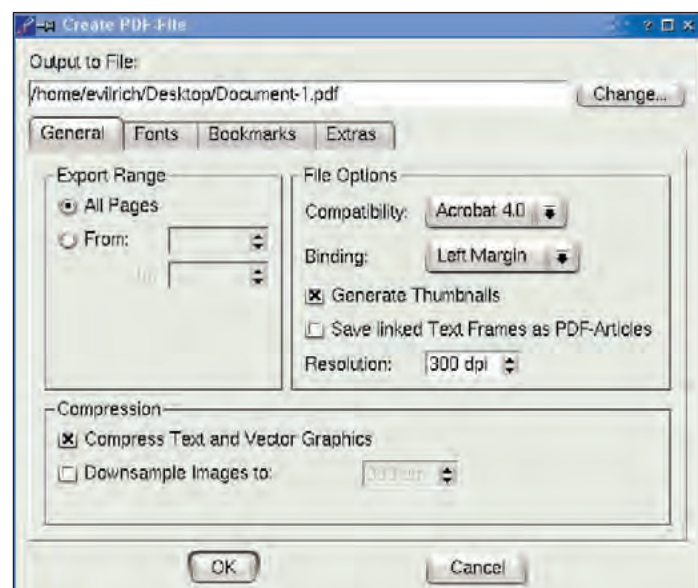
Text layout works as expected, and gives you all the usual formatting controls and the ability to flow text around boxes. Redrawing of text boxes is painfully slow, making any kind of editing of text painful. You can flow around curves as well as rectangles, but there's no option to flow text around both side of a box or shape should you need it. The Styles editor lets you create paragraph formatting styles, but this only lets you choose settings such as alignment, vertical spacing and indentation. There's no facility for associating a font style with a paragraph or creating named font styles at all. *Scribus* does include automatic hyphenation, with hyphenators for English, German, French, Italian and Spanish. It has no spellchecker, however.

A DTP program is only as good as

the output it generates and *Scribus* is well equipped here. It supports RGB and CMYK colour throughout, a necessity for professional work, and it can print separations. Its star feature, however, must be its PDF generator. This provides a good of control over output, and is suitable for generating PDFs for print or for on-screen use. It allows you to select which fonts are to be embedded in the PDF, to compress text and vector graphics, and to downsample image resolution. It supports the hypertext capabilities of *Acrobat Reader*, and can generate bookmarks and presentation effects such as pages wipes and dissolves.

*Scribus* is extensible, both via a plug-in mechanism and scripts. A Javascript interpreter is built-in, and a basic script editor is also provided. Python scripting is available via an additional plug-in, but no documentation is available for either, so their use is somewhat limited. A plug-in for exporting a page as an SVG image may also be installed.

*Scribus* is an impressive attempt at a fully-featured DTP application. It shows much potential, but needs more work before it can be considered a professional package. Speed and stability issues need addressing, and the user interface has some idiosyncrasies which need ironing out. After that, it's documentation, documentation, documentation. **LXF**



**Scribus's PDF export function is particularly well implemented.**

# OPTIMISE your Linux box

From initial choices to final tweaks, **Richard Drummond** gives you a full run down on how to make your Linux boxes faster.

Computers get ever faster, but they're still never fast enough. In this article we'll look at how you can squeeze the last drop of speed out of your Linux setup – whether you are using it on the desktop or as a server. Time, after all, is money, and by expending a little effort on tuning your system, you can reap the rewards of increased productivity, improve the throughput and hence business on your web server or even temporarily put off that expensive hardware upgrade.

If you are using a Linux distribution such as Red Hat, Mandrake or Debian, then there is usually plenty of scope for optimisation. These distros are general-purpose and certain compromises have had to be made to obtain that generality. They are configured to work in a broad variety of rôles and on a wide range of systems, not specifically on your

system for the job you want to perform. By tailoring an installation for your needs, you can gain performance.

Of course, the hardware you use is the principal factor determining your system's performance, no matter how much tuning you do. There's no substitute for raw CPU speed, stacks of memory, fast disk access and network bandwidth. However, you should avoid falling into the trap of believing that processor clock speed is all-important. Generally, adding more memory will serve you better than adding more MHz. Disk speed is crucial, too. And, while ATA technology has made leaps and bounds in recent years, SCSI still rules the performance roost, especially when it comes to supporting multiples disks. If you need disk performance, a fast SCSI RAID array is hard to beat.

Talking about disks, the easiest place to start optimising your system –

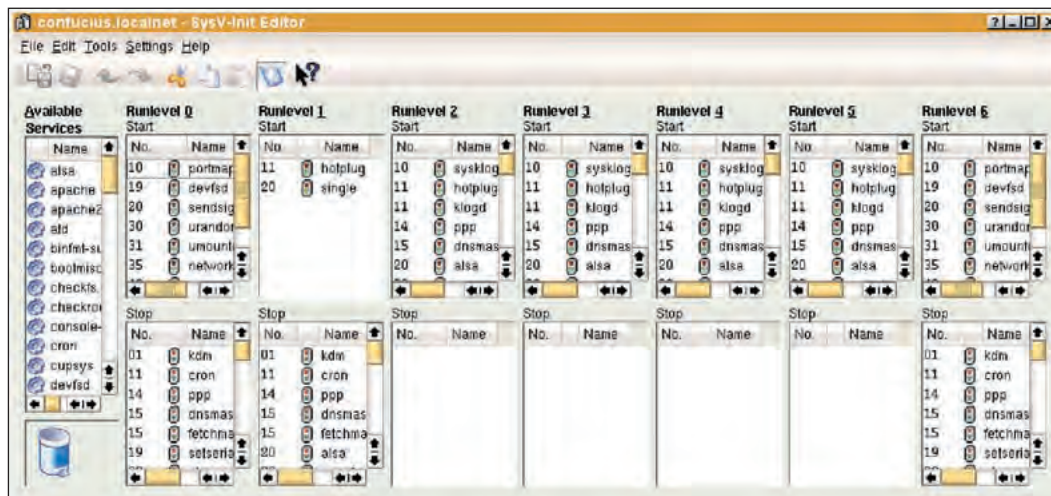
if you are using ATA hard drives, that is – is to check that your drives are being used in their most efficient mode. ATA drives generally support various transfer modes, but you need to ensure that the Linux IDE driver is actually using the fastest mode that your drive supports. If not, disks accesses in Linux could be up to twenty times slower than they might be.

## Hard driving

The ATA standard specifies several transfer modes, which determine how fast an IDE interface can talk to a drive. These fall into two classes – PIO (or Programmed I/O) modes and DMA (Direct Memory Access) modes. DMA allows the IDE interface to transfer data directly into your computer's memory without any CPU overhead, so DMA is preferred if your system supports it. Also, recent improvements to the ATA standard allows for higher speed DMA modes. PIO mode 4, the fastest PIO mode, allows transfers of up to 16MB/s, while the top DMA mode, Ultra DMA mode 6, specifies transfers of 133MB/s (not that current drives can read data that quickly).

Generally your system's BIOS will switch a drive into its highest supported PIO mode on start-up, but you need to enable DMA support in the Linux IDE driver to use DMA modes under Linux. This is either done globally by selecting the 'Generic PCI bus-master DMA support' and 'Use PCI DMA by default when available' options and support for your IDE chipset from the ATA drivers page while building your kernel, or by using the *hdparm* tool as root to tune the IDE driver on a per-drive basis at run time. Recent distros have improved

A runlevel editor such as KDE's *ksysv* can simplify the task of disabling superfluous system services.



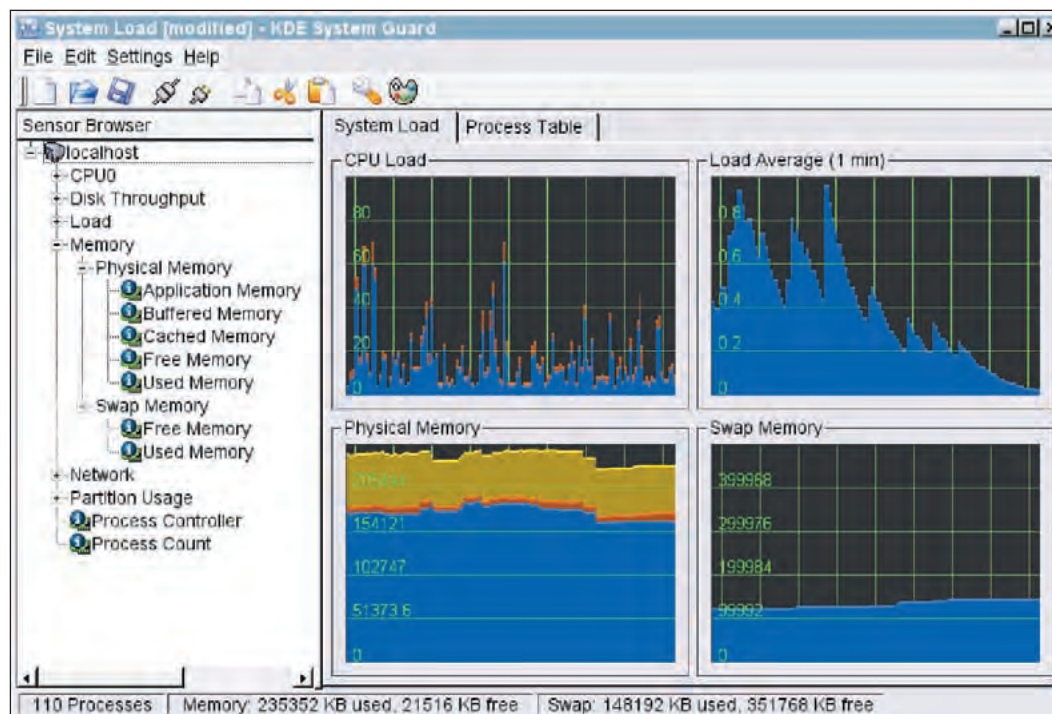


## cover feature



‘Distros like Red Hat, Mandrake and Debian are general-purpose and certain compromises have had to be made to obtain that generality’

# Optimisation



**Using monitoring tools to keep track of memory and swap usage. This will help track down applications which hog memory.**

◀ IDE support and most will enable DMA by default, although, notably, Debian's kernel doesn't.

You can test a drive's speed by calling *hdparm* with the switches **-tT**:

```
hdparm -tT /dev/hda
```

On my system, with DMA disabled, this reports access to my ATA/100 drive's buffer cache as 180MB/s and raw disk access as 2.34MB/s – clearly a lot less than what the system is capable of. We can improve this with

```
hdparm -c3 -d1 -m16 /dev/hda
```

Here, the **-c3** switch enables faster 32-bit transfers to the IDE chipset (note that transfer between the chipset and the drive is always 16-bit, though); the **-d1** switch enables DMA; the **-m** switch enables more efficient multiple-sector transfers (here 16-block transfers). Re-running *hdparm* in test mode now gives buffer access at a more respectable 206MB/s and disk access as 34.78MB/s. Now that's more like it!

On some chipsets and drives, you may need to explicitly set the drive in a particular transfer mode with the **X** switch. For example

```
hdparm -d1 -Xudma5 /dev/hda
```

will enable UltraDMA5 (or ATA/100) transfers. You can check what modes a drive supports with the **-i** switch to *hdparm*. This will print out various information on a drive, including what transfer modes it supports.

Note using *hdparm* is potentially dangerous. You should read the manual page and test any options you use carefully. When you are sure you've got a stable setup, add a line to enable your selected *hdparm* options in your init scripts, preferably after any filesystem checks.

## Software pruning

The next most simple step to optimising your system, is to remove any software you don't need and make sure you've made appropriate choices for software that you do need.

Most Linux distros install a lot of excess baggage that you don't need that, at worst, can waste memory and, at best, just waste disk space. Moreover, the default package choices may not always be the best for your requirements.

The best place to start is with X. Do you really need X? It wastes a lot of system resources if you are not using it. If this is a server box, then you certainly don't need X running all the time, even if it is handy to have around when you want to do configuration locally. If your system is a server and is set up to launch X with a graphical log-in screen at startup, you should disable this and run X manually on demand with **startx**.

On most distros, you can disable the graphical log-in manager by simply changing the default run level. Usually, run level 5 starts a graphical log-in, while run level 3 provides the normal multi-user console login. Load up the file */etc/inittab* in your editor and look for the line **id:5:inittab**. Change the **5** to a **3** and save the file. Changing the default runlevel in *inittab* will only take effect after a reboot, however. To alter the run level of a running system, use the command **telinit**.

Debian-based distributions are different in that, if you have a graphical log-in manager installed, this will be configured to run by default for all run-levels. To disable it, you have to remove the link which starts the log-in manager in the startup scripts corresponding to your run-level. For example, to disable the KDE log-in manager, *kdm*, in runlevel 2, use

```
rm /etc/rc2.d/S99kdm
```

Alternatively, use a configuration tool such as *ksysv* or *linuxconf* to manage your init scripts.

## Slimming services

Next, disable any services you don't need, and think carefully about the ones you do need. Don't just blithely accept the web or ftp server that gets installed by default with your distro, but investigate alternatives. Have you looked at *vsftpd* (<http://vsftpd.beasts.org/>), a very fast, scalable ftp server, or *MaraDNS* ([www.maradns.org](http://www.maradns.org/)), a simpler alternative to *BIND*?

How do you determine if you're running unnecessary services? Well, a service can be run in one of two modes. It is either run as a daemon (that is, it

## Start at the beginning

### Installation advice

The best time to start thinking about optimisation is before you even install Linux. Many of the choices you make here will affect the speed of your system, and it's easier to get it right now, rather than having to change things later.

First, choose the right distro for the job. If you are installing a server, you may want to investigate server-optimised distros. Also, consider choosing a distro built for your particular CPU variant. For instance, Mandrake is optimised for Pentium processors, while Red Hat and Debian are compiled for 386s. Choosing one that's better optimised for your processor can potentially offer a performance boost – especially on desktop systems – and could save you the job of recompiling later on. Of course, you could opt for a source-

based distro such as Gentoo Linux which compiles packages specifically for your system as it installs them, but for most people the amount of time this takes to perform the build and install will outweigh the gain in performance.

Having chosen your distro, next plan how you are going to use disk space and think about what filesystems you will use. Do you need journalling? If you have multiple disks, can you take advantage of RAID? How much swap space do you need?

Be ruthlessly efficient when it comes to the actual installation. Don't opt for the lazy automatic installation. In particular, this applies to package selection. Don't go for the easy 'install everything' option. Think about what software you really need. Don't waste time, memory and disk space on stuff that you won't use.



## Swap shop

### The paging slow down

The first thing to consider when allocating disk space to Linux is the swap file. The swap file is used by Linux when main memory becomes full – Linux caches – or swaps out – pages of memory to the swap file until they are needed. Thus fast access to the swap file is crucial for heavily loaded systems.

Linux supports swapping either to a file or a dedicated swap partition. You should always opt for the latter because it doesn't have the overhead of an underlying filesystem, so is faster.

**How big should a swap partition be?** There's no definitive answer to this question. As a rule of thumb, the swap partition should be at least as large as the amount of main memory you have, and most people suggest an upper limit of twice the amount of main memory. You can use the *vmstat* tool (part of the standard *procps* suite) to monitor your system's swap usage over time.

**How you improve swap performance?** If you are starting with a clean disk, one thing you can try is to allocate the swap

partition as the first partition of the disk or as near it as possible. This will locate the swap partition physically on the outer edge of the drive where the drive head moves fastest over the disk and so access will be fastest. Another thing to try, if you have multiple disks, is to split the swap partition over multiple disks. Linux supports parallel access to multiple swap partitions (you don't need to explicitly use RAID), and this will permit better swap performance. Simply create a swap partition on each disk and

mount them with the same priority. (You can specify a priority with the keyword **pri=** in the option field in */etc/fstab*.) This works best with SCSI disks, but you can use this technique with ATA disks as long as the disks are on different IDE channels. (The ATA protocol does not allow simultaneous access to a master and slave on the same channel.) Of course, the best system performance is gained by having sufficient memory that you don't need to use virtual memory at all.

runs continuously, listening to a particular network socket or sockets) or on demand (via the network superserver *inetd* or *xinetd*). The command

```
netstat -ip --ip
```

will list any daemons running and the ports they are listening on. To determine which services are configured to run on demand, examine the file */etc/inetd.conf*.

Consider whether each service is necessary. Running superfluous services will not generally consume that much memory, but will slow the start-up time of your box and can potentially be a security risk.

If this is purely a desktop machine, many of the default services will be unnecessary. For example, if you are collecting and sending mail by directly logging in your ISP's mail servers, then do you need a mail server such as *sendmail*? And do you really need to run a web server? Unless you are doing web development, probably not. If your system has a web-based documentation server, then you don't need the heavyweight *Apache* for this job. It can be replaced with something lighter and quicker such as *Boa*.

Use a run-level editor such as KDE's *ksysv* or a general config tool such as *linuxconf* or *Webmin* to disable unnecessary daemons. Alternatively, this can be done manually by deleting the appropriate links in your start-up scripts. To disable *inetd* services, edit the */etc/inetd.conf* and comment out the services you don't need.

For the remaining services that you do need, consider whether you they should be run as a daemon or on demand. For example, if a machine isn't a full-time mail server, then you can run your mail service on demand. It will be slower to start up, but it will

only consume resources when it is being used. To do this, delete its entry from the init scripts and add an SMTP service to launch it from your *inetd.conf*. For example, I run the mail server *exim* from *inetd* with the following entry:

```
smtp stream tcp nowait mail
usr/sbin/exim exim -bs
```

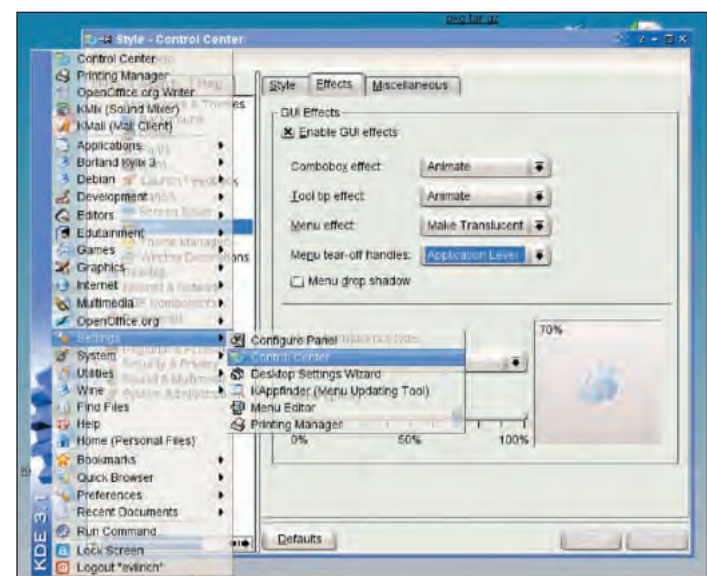
Finally, check that the periodic command scheduler, *cron*, isn't running unnecessary jobs. Typically, *cron* will be set up to run a host of housekeeping and accounting tasks on a daily basis, such as rebuilding the man page database, updating the locate database and rotating log files, but, depending on your requirements, many of them may be unnecessary on your system. Consider whether it would worthwhile disabling some of these, and run these jobs manually when you need to. The simplest way to manage *cron* is with a crontab editor such as *gcrontab*, but you can edit */etc/crontab* and its associated configuration files manually too.

## Desktop dynamite

If you are tuning a desktop system, there's plenty of scope to get more

**'If you are tuning up a desktop system, there's plenty of scope to get more speed, starting with your X server and desktop settings'**

speed. First, check your X config. Are you running the right X server for your hardware? Most cards will give the best performance under the latest *XFree86* 4.2.x using the appropriate accelerated driver. This is preferable to use than any generic server such as the *fbdev* driver



or the the *SVGA* server under *XFree86* 3.3.6. Examine the X log file – usually */var/log/XFree86.0.log* – and check that acceleration is enabled. If you have a supported 3D card, check that this is successfully enabled too. This will give a huge performance boost with *OpenGL* apps. Again, examine the log file to see if direct-rendering is enabled. With some AGP cards you

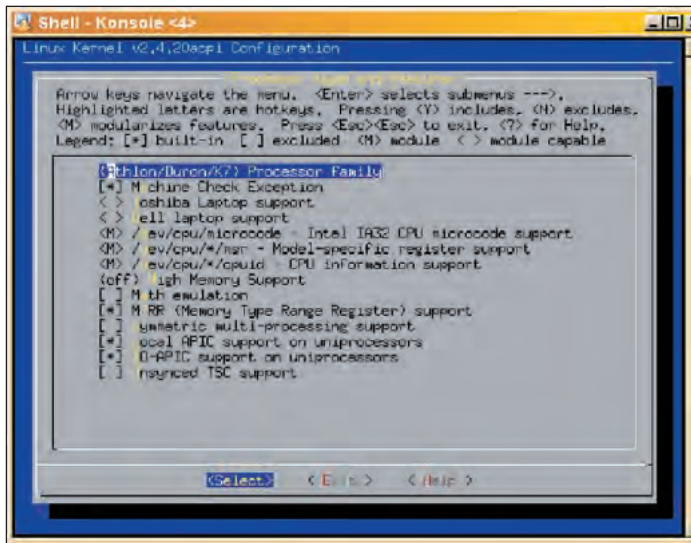
**Try to avoid overdoing the eye-candy. Translucent menus look cool, but will slow your system down.**

may have to explicitly specify which AGP mode to use in your X config. Consult the X docs for your driver.

If you have X running smoothly, are you using the best screenmode? Generally, 15-bit or 16-bit modes will be the fastest on most graphics cards,



# Optimisation



When configuring your kernel, make sure you select the appropriate CPU type for your target system.

◀ although with newer cards 32-bit modes may be equally as quick. Of course, 32-bit modes require more memory, so you may prefer to stick with 15- or 16-bit modes. You should experiment and

one mono-spaced font. Disable any font servers you are running, and get X to access fonts locally. If you have enabled anti-aliased font rendering with *Xft* and the X Render extension then running a font server is a waste of time, because applications which use *Xft* will only access local fonts.

Are you running an appropriate window manager and desktop environment? While KDE and GNOME are comfortable and easy-to-use, can you make do with something lighter, such as *XFce* (see [www.xfce.org](http://www.xfce.org))? If you must have GNOME or KDE, try replacing the default window manager with something simpler and more efficient. Be equally critical of the apps that you run on the desktop. For instance, is it really necessary to run the full *Mozilla*? Might not the lighter *Galeon* do instead?

## Build-it-yourself

The majority of distributions ship software as pre-compiled binaries,

toolkits such as *Qt* and *GTK+*, and even desktop environments such as KDE can all benefit rebuilding with optimisations for your target system.

First check whether you have an appropriate version of *GCC* installed. Generally, the newer the better, but note that when it comes to compiling C++ software, different version of *g++* may not be compatible – it is best to stick with the version that ships with your distro.

Rebuilding software is generally straight forward, as most will use configure scripts to set up compilation options. Unpack the source tarball, cd into the directory and do

```
./configure
```

```
make && make install
```

to build and install an app. Note that **configure** will support various options, which you can list with the **-help** switch. Examine this output and decided which are required for your system. Reading any supplied docs will usually help. A standard configure option is the **-prefix** switch which specifies the install path for the package.

To pass extra compilation options to the compiler to enable optimisations when using a configure script, it is often simplest to use environment variables. The variable **CFLAGS** will specify extra compilation option for C programs, while **CXXFLAGS** specifies options for C++ programs. For example:

```
CFLAGS="-O3 -march=i686"
```

```
./configure --prefix=/usr
```

will run the configure script and tell the compiler to use full optimisation and to build for i686 (P II and better) CPUs.

We don't have the space to cover all the optimisations that *GCC* supports (for that consult the manual), but the following options may be useful to get *GCC* to optimise for speed and so produce faster executables.

```
-march=x
```

**'By rebuilding selected packages from source for your system, your CPU, and using only the options you require, you can gain a big performance boost'**

see which suits you and the amount of video memory you have.

Next make sure you are not wasting memory with too much eye-candy. Solid backdrops are much quicker to render than gradients and images. Opaque window moving and sizing, translucent menus, animated chooser gadgets all look pretty but will waste memory and processing power. Reign in any tendencies you may have for excess theming.

Are you using too many fonts? To save memory, restrict font usage in your applications to two or three, say, one serif font, one sans-serif font and

which, while easier for the end user to install, are probably not optimally configured for your setup. By rebuilding selected packages from source for your system – for your particular CPU and using only the options that you require – you can potentially gain a big performance win. However, this can be time-consuming, so it is best restricted to major software components where the effects of recompilation will be most noticeable.

Any processor-intensive app is a good candidate for rebuilding. On desktop systems, *XFce86*, graphical

## Filesystem tips

### Tips and tweaks

Linux now supports several filesystems which are suitable to host your system. The faithful *ext2* filesystem now has competition from *ext3*, *ReiserFS*, *JFS* and *XFS*. Which is the fastest? The only sure way to find out is to experiment yourself.

You might think that the non-journalled *ext2* filesystem is faster than *ReiserFS* or *XFS*. This is actually not

generally the case, although *ext2* is faster than *ext3*. For the times that *ext2* is faster, the benefit of journaling for filesystem integrity outweighs any performance advantage that *ext2* might have. And although *ReiserFS* and *XFS* perform better than *ext3*, they are less well supported by tools, so you may actually prefer to use *ext3* anyway.

A useful tip for getting the best from your filesystem is to try and gain as much parallelism from you disk subsystem as possible. If you are running a web server, for example, store your server logs on a separate disk from your document root. Alternatively, using the kernel's software RAID driver in RAID-0 mode (striping to multiple disks) is a

cheap way to gain performance.

A useful mount option for gaining maximum speed from a filesystem is **noatime**. This disables the updating of a file's last access time property, and can be particularly effectively in cases where you don't need this information. With *ReiserFS*, the **notail** can also boost performance when writing small files.



This option will enable compilation for the specified CPU model *x*. On x86 platforms 'x' might be one of i386, i486, i586, i686, Pentium4, K6, Athlon, etc. (You should check the manual for your version of *GCC* to see which models are supported. Newer versions of *GCC* support optimisations for more CPU types.) This will generate instructions and optimisations for the given CPU and may produce executables which will not run on other CPU models. Conversely, the option `-cpu=x`, will generate optimisations for CPU but will generate code that will run on any model in the CPU family.

#### **-O3**

Enables the highest optimisation level supported by *GCC*, including some optimisations which sacrifice executable size for speed.

#### **-fomit-frame-pointer**

Don't keep the frame pointer in a register. This optimisation saves the overhead of setting up the frame-pointer in functions, and potentially frees up an extra CPU register. It is not supported by all architectures, though, and can break compatibility with debuggers (although that's probably not an issue for the end user).

#### **-ffast-math**

This enable various optimisations for floating-point arithmetic that can lead to faster computation, but may violate ANSI or ISO standards. It is generally safe to use this option, but it may break programs which rely on an exact floating-point arithmetic.

#### **-finline-functions**

Allows *GCC* to compile small functions in-line rather than calling them as subroutines.

#### **-funroll-loops**

Allows the unrolling of some loops when the number of iterations is known at compile time.

## Kernel tweaking

When it comes to building packages yourself, often the best place to start is with the Linux kernel itself. While configuring and compiling the kernel may sound daunting, by building for your CPU version and only including the options you need, you can speed up the kernel, consume less memory and boost start-up times.

We can't go into too much detail about kernel configuration issues here

– that would warrant an article of its own – but we can give some tips. If you are unsure about kernel building, the Kernel HOWTO at the Linux Documentation Project ([www.tldp.org](http://www.tldp.org)) gives a good overview.

The first thing to do is to go to the 'Processor type and features' page and ensure you pick the right CPU type for your target machine. While you're there, make sure you have 'MTRR support enabled' (this allows the CPU to control access and caching to specified blocks of memory and is used by X and the kernel framebuffer drivers to boost performance to your graphics card's video memory). If you are building for a uniprocessor machine, de-select 'Symmetric multi-processing support'. This will build a faster kernel.

As far as the majority of kernel options are concerned, only include support for drivers or features that you really need. Don't waste time and memory of things that you don't need. A question often asked is whether you should build kernel modules as built-in or loadable modules. Loadable modules have the advantage of only requiring resources when they are loaded and used, but some people argue that built-in modules are more efficient. If that is true, the difference is slight. As a rule of thumb, configure modules that you need at boot time or that you need continuously while your system is up as built-in modules; any others can be loadable modules. So, drivers for disks and filesystems should be built-in. Filesystem drivers that you use occasionally can be loadable modules. I generally configure things like plug-and-play support, the serial port driver, input driver support and things that are useful to have even before you have access to a filesystem as built-in modules. I configure sound card drivers, network device drivers, and USB modules to be loadable. Of course, your mileage may vary. For example, if you need to boot from a network, you need support for your network card built-in.

If you want the top performance from your kernel, you should consider various third-party patches and extensions to the kernel. One example is Robert Love's pre-emptible kernel

## Turbocharging Apache

### Cutting down on the World Wide Wait

Since a top application for Linux is web-serving, many will want to now how to tune *Apache* set-up for speed. The *Apache* docs at <http://httpd.apache.org/> is the definitive for source of reference, but here are seven top tips to get you started.

**1** If you are using *Apache 1.3*, consider running *Apache 2.0*. It's new support for multi-threaded worker processes can really boost performance on heavily loaded systems.

**2** Do you really need access logs? If not, turn off logging by directing log output to `/dev/null` with the **TransferLog** option.

**3** Unless you use `.htaccess` files to

control access to directories, disable this with **AllowOverride None** so that *Apache* will not look for an `.htaccess` file for each request.

**4** Disable DNS look-ups for each visitor to your site by setting **HostnameLookups off**.

**5** Consider using a separate server for serving static content such as images or static html from your site. Choose a server which offers better performance for static contents such as *Boa* or *Tux*.

**6** Rebuild *Apache* yourself and only build the modules that you need.


**7** If you are using PHP, consider using a PHP accelerator. See our PHP acceleration feature in *LXF 34*.

patch. (see [www.tech9.net/rml/linux](http://www.tech9.net/rml/linux)). This decreases system latency by allowing processes executing kernel code to be pre-empted. On interactive systems this can boost the overall responsiveness of the system. It doesn't make your box any faster, it just seems faster, and it's probably of little value on servers.

There are many useful performance patches that haven't made it to the stable kernel, such as improvements to the scheduler, to virtual memory performance, and to Linux's threading model. Such thing are beyond the scope of this article, though. See our feature on kernel patching in *LXF 29* or visit sites such as Kernel Traffic (<http://kt.zork.net/>) to keep abreast of the latest patches.

## It doesn't end here

In this article we covered some of the basic, general techniques you can apply to gain better performance for your system. You can do a lot more. It all depends on the amount of time you spend on tweaking. This is likely a diminishing return, however. Expending double the effort, won't necessarily give twice the benefit.

We haven't had the space here to discuss many specific apps. Once you have your basic system, your kernel and your filesystem running at top speed, it's time to turn to tuning apps. The techniques you'll use here will depend on the particular app. There's no substitute for reading the docs and finding out what configuration options are supported. Above all, experiment and have fun! 



# desktopdreams

**Linux may be the open source darling of the media, but it's not alone, especially in the quest to create a satisfying desktop environment. Nick Heron discovers how a small group of OS projects, though humble in nature, are determined to do it their way.**

**S**yllable OS was released into the wild by its developers in July. Though they got less press than a new distro release, that's OK. They're different from a new distro, they're part of a new breed of open source projects aspiring to create a superior desktop.

Linux users will be familiar with some of the themes at play in this crowd. They're determined developers with uncompromising standards, working in a field dominated by big commercial interests or long held traditions, and they're dissatisfied with the state of current offerings. Moreover, they represent diversity and a desire for viable alternatives.

Syllable is part of a group of OS projects that have been made possible by the success of open source software and Linux, both its proliferation and technical accomplishments. When building these projects, coders call on a range of tools and inspiration that have come from similarly minded coders worldwide.

"If it has done nothing else (and we all know it's done a lot besides), GNU/Linux has shown that an open source operating system can be built as well as, and even better than, proprietary software, and it can carve out an ever-growing and relevant space for itself in the world. GNU/Linux has illuminated the path and shown us that what

we're attempting is very possible and very doable," explained developer Erik Jaesler a team leader from the OpenBeOS project.

All of these projects enthusiastically embody the open source do-it-yourself spirit, or the build-it-if-you-want-and-trust-that-others-will-recognise-your-bold-gesture-and-its-worth spirit. Like any open source project there is the potential that they will influence and benefit the course of other projects. Their current goal is simple: build high-performance desktop environments.

We're all familiar with complaints about the speed of X11 or its ability to handle modern GUI functions or its appropriateness for all manner of applications and games. These questions of suitability are raised whenever anyone wonders why Linux isn't replacing Windows on the desktop. This debate is especially difficult when both Windows and OSX are actually improving their interfaces. They may be resource hogs, but the end results sometimes seem worthwhile.

The goal set by these new projects may seem daunting to most... but remember, the quest to build a better tool is seductive.

**'Linux users will be familiar with some of the themes at play: determined developers with uncompromising standards working in a field dominated by big commercial interests'**



## SYLLABLE

The youngest member of this nascent movement is Syllable OS, a full fork of AtheOS – a hobby OS created by developer Kurt Skauen. In its relatively short life AtheOS has earned a small and loyal following and a reservoir of applications.

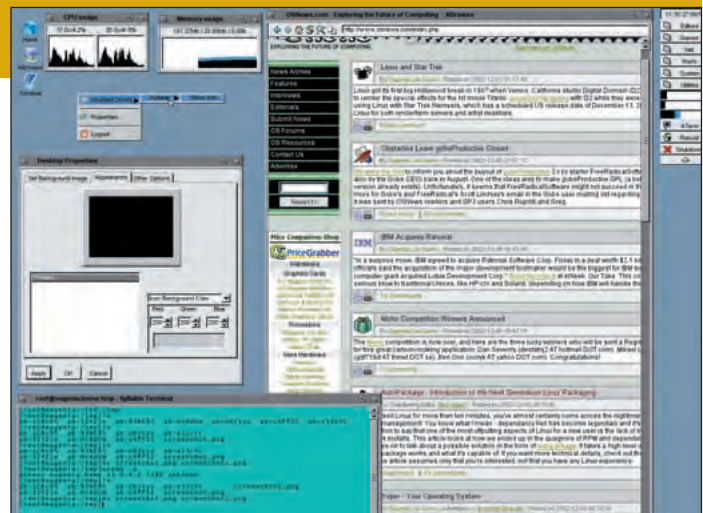
AtheOS became attractive to developers and users looking for an OS project with some vitality. The AtheOS kernel, written from scratch, has a fully pre-emptive kernel, a 64-bit journalled filesystem, supports SMP (Symmetric Multi Processing), and has a built-in network TCP/IP stack. AtheOS supports loadable device drivers and filesystems. And though in some ways it resembles X11 because the client and server don't have to be on the same machine, the AtheOS GUI is more closely integrated into the OS than X11 is with Linux.

Back in July of 2002 a group of AtheOS fans started Syllable: "I guess I would say that it all was started because a bunch of us could not wait around for Kurt Skauen anymore," said Rick Caudill one of the founders of Syllable. "I saw

a good OS go to waste. AtheOS had so much potential. One guy created all of AtheOS himself, and it was great! Kurt is a brilliant programmer. Up there with some of the greats, but as he said many times, "it's just a hobby."

Kristian Van Der Vliet, a UK based developer, is the Syllable Project Manager and lead GUI developer. "We want something that looks and acts consistently, and is easy to use," said Van Der Vliet. In simple terms, this means if a user ever has to edit a config file by hand, or if you must read a HOWTO before changing a setting, configuring the desktop or running an app, then it's too hard. "If you must spend time thinking about how to do something beforehand, then we haven't achieved our goal," said Van Der Vliet.

"As for the construction of the display server, Syllable has an application server, which deals with the actual graphics, and a GUI class library which provides all the utility classes and GUI widgets. The application server, graphics card drivers and library all run in user space."



Syllable desktop – take a look at the code to see where the work has gone.

Syllable's responsive interface has a minimalist approach right now. "It seems that the current trend is to make the user interface as flashy and bright as possible. WindowsXP, KDE 3 and to a lesser extent OS X, have all done the same thing. It seems that everyone has forgotten that the point of a user interface is to make it easy for users to manipulate data visually, not to smash them over the head with something that looks like an explosion in a finger-

paint factory," said Van Der Vliet.

"I don't want Syllable to end up looking like that; I think that the main goal of a user interface should be to get out of the way, become invisible. The user just wants to get their work done, not admire the view!"

"The desktop should follow the same sort of philosophy, and simply try to get out of the user's way," said Van Der Vliet. <http://syllable.sourceforge.net/>

## COSMOE

Syllable was not the first fork of AtheOS code. This honour goes to developer Bill Hayden, who fused the display server of AtheOS to the Linux kernel and created CosmoE.

Few can argue that perhaps the greatest barrier to adoption and proliferation of a new OS is adequate driver support. Linux enthusiasts, if they don't remember a time when they were buying from the relatively small pool of hardware that was supported, will have heard the stories. Marrying a faster, dynamic display server to the Linux kernel solves these problems for CosmoE explained Hayden. "In my opinion, the greatest usability feature of CosmoE is not in its interface, but rather that it sits on top of the Linux kernel, thus inheriting the plethora of Linux hardware drivers."

Hayden is also quick to point out that there is nothing intrinsically tying

CosmoE to the Linux kernel. CosmoE could be relatively easily ported to FreeBSD or Darwin.

One of the foundational features of CosmoE is that it is not X11-based. "X11 is really showing its age and is long overdue for a replacement that is more elegant and easier on the developer and the user. No matter what pretty face you put on X11, from a usability standpoint, it lags far behind the MacOS and Windows display environments. CosmoE aims to close the gap," said Hayden.

The CosmoE display server communicates with client applications via TCP/IP. Like X11, the server and client need not be on the same machine. Upcoming releases will be using *DirectFB* (See *DirectFB?* box) for graphics rendering. *DirectFB* is a graphics, input and windowing system that integrates several applications into one shared

"desktop." Unlike X11 *DirectFB* doesn't have a Client/Server architecture, instead it uses IPC mechanisms like shared memory and messaging. Under *DirectFB* all applications directly execute graphics operations and are able to write directly into their graphics memory.

Despite a number of releases, the possibility of significant evolution of the interface is something that Hayden is well aware of. "I think it may be premature to associate the current CosmoE interface with what the end product will look like. My work on CosmoE so far has been mostly geared toward under-the-hood infrastructure changes to enable the 'opportunity' to create a great interface on top of it. We're not there yet," said Hayden.

"I'm looking to build something that anyone can look at and immediately feel comfortable with. I've been inspired by the BeOS's elegance

and simplicity. You don't need obnoxious graphics and a slew of 'wizards' when things are already simple enough to understand in the first place. That is not to say that I want to build a dumbed-down idiot-proof OS, but rather something that is both beautiful and powerful from a less-is-more perspective," said Hayden.

As a developer project, CosmoE has some, but not many, full featured apps available. "CosmoE is still very much a piece of software to work 'on', not 'with'. The nice thing is that as the Be API compatibility improves, the pool of apps will grow simply by the fact that most Be apps can now be compiled for CosmoE with little or no changes," said Hayden. In addition CosmoE features support for the AtheOS, Linux APIs, and for Macintosh's Carbon.

Developer releases of CosmoE are available at: [www.cosmoE.com](http://www.cosmoE.com)

# DesktopDreams

## BeOS HISTORY

**Two other desktop OS projects,** OpenBeOS and the Blue-EyedOS, draw their inspiration directly from the BeOS, produced by the now defunct Be Inc. These two projects were spawned independently of each other by developers seeking to recreate the BeOS experience.

So, what is the big attraction of the BeOS?

The BeOS was a single user, preemptively multi-threaded operating system with an innovative 64-bit journalled filesystem (BFS) and support for Symmetric Multi-Processing. The BeOS was written from scratch with the most modern operating system theory of the time, the best that solid C++ design could offer. The entire concept was optimised for multimedia applications such as video editing, or audio applications. It was an effort to bring high-end graphics and display workstation capabilities to the PC. Being very POSIX compliant, meant that Linux and Unix command line software could be ported to BeOS with relative ease.

Be Inc. was started in 1990 by a couple of former Apple computer executives, Jean Louis Gasee and Steve Sakoman. Initially the stated goal of the company was to build low cost, high-performance SMP machines that were ideal for digital content creation and multimedia. Their first creation was the BeBox, a machine running five Hobbit processors from AT&T. The box also featured a lightweight and nimble

operating system. Then Be moved to the PowerPC platform for several years. It turned out that people were very interested in their high-performance, cruft free OS. Be began a move that they originally said they wouldn't do, they ported the OS to Intel and were brought into competition with Microsoft.

At times Be's business prospects seemed excellent. Be was often described as a trade show darling. This ended up being a somewhat backhanded compliment. Among other things, BeOS impressed people with the ability to run multiple streaming videos simultaneously without flutter. The BeOS typically booted under 20 seconds, and it opened applications nearly instantly. Be Inc. trade show booths routinely attracted gaping crowds that would stand in amazement at demos. Then, discovering that it didn't have MS Outlook, or some such thing, they would stroll away thinking that Be was just a curiosity.

At one point Be Inc. was on the verge of being purchased by Apple, which had plans to turn BeOS into the base for OSX. However, Be rejected the lucrative deal and Apple went on to purchase Steve Jobs's NeXT Computers. Years later Be's once bright future had faded. The OS was released free for download in an effort to boost the user base. This small (45MB) executable download installed the BFS and the entire operating system inside one 500 MB file in either an ext2 for FAT partition.

In early 2000, with dwindling capital, Be changed directions and tried to become a leader in the

"Internet Appliance" market by turning their core technology into BeIA. The most prominent product to come out of the effort was the poorly marketed and supported eVilla from Be's partner Sony. The eVilla entered a crowded and questionable market for Internet Appliances and fizzled. When this new effort didn't yield lucrative returns, Be began a series of layoffs and eventually sold the rights to its intellectual property and the contracts the contracts of 50 engineers to Palm Inc. Roughly the same deal that Apple had offered earlier was now sold to Palm for a fraction of the price, \$11 million USD

In February 2002 the remnants of Be filed suit against Microsoft Corporation for the destruction of Be's business resulting from the anticompetitive business practices of Microsoft.

Before it ceased operations, Be Inc. managed to open source some small parts to the OS, the *Tracker* and the *Deskbar*. The *Tracker* managed the desktop itself and the *Deskbar* which shows running applications. *Tracker* can be described as an amalgam of the best-of-breed in file manager features. *Tracker* also capitalised on the file 64-bit journalled filesystem features. *Tracker* leveraged the BFS heavily. MP3 fanatics can go on for days about how much easier it is to organise their music because of *Tracker's* ability to sort on extended attributes. In conjunction with the BFS, system searches in *Tracker* offered nearly incomparable interface speed. Copies of BeOS Personal Edition and BeOS software are still available for download at [www.bebits.com](http://www.bebits.com)

## OPENBeOS

### The largest of the open source

efforts to recreate the BeOS is OpenBeOS (OBOS). This project was started within days of Be's announcement of the deal with Palm. By this time the community of Be users and developers had all but given up on the future of their favourite OS, however some were prepared to start the hard slog to recreating the BeOS for themselves. The goal of OBOS is simple: to create a clone of BeOS 5 from scratch then develop from there. The greater goal is that, by developing their own open source version of the BeOS, it will become a powerful tool in the hands of the open source community, and that its future will not be determined by corporate concerns.

Erik Jaesler is an American developer living in Santa Cruz, and the leader of the Open BeOS App/Interface team. Jaesler sums up the appeal of the BeOS user experience in a word: smooth.

"As a long-time Windows user, there was a tiny shift to a more Mac-like way of interacting with the UI, but it wasn't long before Windows was bugging the snot out of me. The UI interaction of BeOS was natural and convenient. Of course, there's the legendary responsiveness – unlike Windows, you never find yourself waiting interminably for a window to paint itself," said Jaesler.

## DirectFB?

### An integrated windowing system

**D**irectFB is a free project from Convergence GmbH, a Berlin based company creating software for digital set-top-box appliances. *DirectFB* development is geared towards need for digital television applications on embedded systems. Convergence determined that a Multimedia Home Platform (MHP) would require management of multiple display layers (graphics, video, background) with alpha transparent graphics in the graphics layer and alpha blended graphics operations as an extension to the Java 1.1 AWT. These

are functions that no existing graphics solution under Linux (including X11) could provide.

*DirectFB* lead developer Denis Oliver Kropp introduces us to his project.

**LXF** *DirectFB* was originally designed for embedded devices. How well is it suited for use on the desktop?

**DENIS KROPP** One key aspect of *DirectFB's* development was scalability. Running on an embedded device it saves resources. *DirectFB* can be built in single application mode which uses simple

mutexes only and direct calls instead of message dispatching. A *DirectFB* installation requires 200KB+ disk space depending on the drivers and modules built. Applications can be linked statically with drivers and modules linked. There are no *DirectFB* components (or any files) installed then besides the executable of the application itself. On machines with a recent graphics card it utilizes all available hardware acceleration as supported by the driver. The multi application core allows a complete desktop system to be built on top.

However, for a full featured desktop *DirectFB* requires some work on the interoperability between applications. There need to be hooks for a common window manager that is used for all applications including native *DirectFB* and X11 applications using common window borders (with an alphachannel for shadows, etc.). Currently there are some shortcuts built into the *DirectFB* core to move, resize, restack, etc. windows. X11 windows are currently managed by the X11 window manager only.

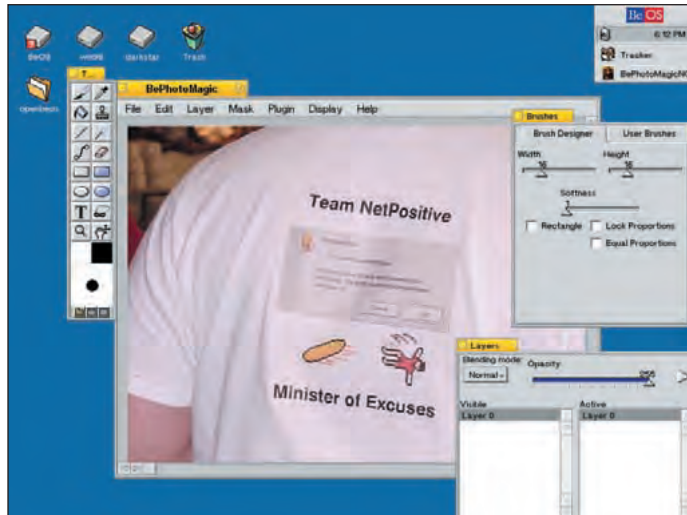


"People are so used to waiting eons for their Windows apps to launch that I've actually seen jaws drop when users first see how quick BeOS is. Maintaining that responsiveness is our number one goal. Windows and MacOS can crib all the little features they want, but they don't touch BeOS when it comes to the sheer quickness and unusual nimbleness of the BeOS experience," explained Jaesler.

To achieve a nimble UI, the OBOS team will be doing the same thing the Be, Inc. team did and pervasively multithread the entire UI. Every window that is created causes two threads to be spawned: one in the app for the window's message loop, and another in the app server, where all the real graphics action happens. Beyond that, the plan is to create clean, fast code. Understandably, building an OS from scratch has some challenges, however considering that the goal isn't to innovate, but simply to duplicate the BeOS R5 experience, a good deal of the planning and decision making for the first release of OpenBeOS has already been made.

If you've used BeOS, you'll be familiar with the desktop of OBOS. If not, *Tracker* is hybrid desktop with overtones of the old MacOS, but it also features multiple desktops, and gives access to a powerful command line shell.

"Although there has been a lot of work with various \*nix desktops – and they're definitely getting better – I



**The familiar BeOS desktop, with all its responsiveness, is what the OpenBeOS team are aiming to bring to the open source community.**

don't think there is a truly compelling open source desktop environment yet. This is the niche that OpenBeOS will fill. While the \*nix desktops work valiantly to hide the complexity of the underlying system from the average user, BeOS just doesn't have the complexity to begin with. The user is never going to be confronted with the grim realities of a \*nix backend, because it doesn't exist. Yet, for those of us with propellers on our hats, the shell environment is right there at our beck and call, and all that sophistication is available. BeOS is a built-for-purpose desktop GUI operating system and that makes a huge difference," explained Jaesler. Jaesler tells us that work on

OBOS is moving along nicely, though he's certain that everyone involved wishes it were going faster. "OpenBFS is on beta 5 and I think they'll probably be ready to release soon; a lot of the unsexy guts of the rest of the system are making good progress as well, but that's not the sort of thing that shows up in a screenshot," said Jaesler.

Jaesler explains some of the plans for the future OBOS, "We want to expand the variety of widgets available and open up the system for more user customization. I tend to take issue with those who want to customize every last detail of the UI; I have a very utilitarian view of computers, and really see them as a tool to help me

get work done, rather than an expression of my personality. Having said that, it would be silly to not enable relatively simple customisation: setting various system colors (menu color is all that can be tweaked now) and different window borders for starters – nobody wants to look at an ugly box, after all. We're investigating several schemes to extend the customization further than that in a way that makes it easy for third party widgets to take advantage of the system and doesn't turn the UI into a complete CPU hog. Beyond that, we'd like to do some refactoring and behind-the-scenes rearchitecting to make the API even more logical and easy to use. The UI system and its API are really wonderful and quite mature as they are, so the main challenge for us will be integrating our changes in a way that feels natural, rather than making any drastic alterations," explained Jaesler.

And how will the OBOS community continue to grow? "The crisp, clean nature of the BeOS experience, both in use and development, is a big draw; it's what brought in most of the people who already use and work on it. The opportunity to be involved with something different, something that's cutting a new path is also a real attraction," said Jaesler.

The OpenBeOS project plans to have a preliminary release ready by late spring or early summer 2003.

After adding the remaining framework for a full featured desktop environment *DirectFB* will be a good choice for Desktop OSs. And *XDirectFB* can run in the background for X11 backward compatibility.

**LXF** What would an X11 developer encounter if they wanted to switch to *DirectFB*?

**DK** *DirectFB* is just a library, there's no server that needs to get started. The first application started is called the "master" and it opens the input devices and sends

events to other "slave" applications. The master application can be any *DirectFB* application, it just needs to run as long as there are other applications using *DirectFB*.

One big difference from X11 is that *DirectFB* windows have an own surface (graphics memory for the whole window content). There are no expose events, applications just draw if anything changes in the user interface. This model is required for overlapping translucent windows of different processes.

The graphics operations include alpha blended blitting and some other features that are available via the texture mapping unit. In fact *DirectFB* windows are textures.

*DirectFB* has an integrated font rendering that supports anti aliased unicode fonts at full speed, e.g. 900.000 characters per second with a font size of 22 on a Matrox G400. Strings are always UTF-8 encoded.

**LXF** Do you have any specific examples of how *directfb*'s architecture shines?

**DK** *XDirectFB* may be started as the master application. All X11 applications can be run and the user can adjust the opacity of each X11 window. Additional *DirectFB* applications (windowed or fullscreen) run in slave mode. Non-X11 windows and X11 windows are mixed on one desktop. You can even start two *XDirectFB*'s in one session having two separated X11 desktop environments and window managers integrated in one desktop on the screen.

[www.directfb.org](http://www.directfb.org)  
[www.convergence.de](http://www.convergence.de)

## BLUE-EYED OS

**A move closer to Linux arena is** the Blue-Eyed OS (B.E.OS). This bold project aims to duplicate the BeOS user experience and offer compatibility with BeOS applications by using X11 and the Linux kernel.

Guillaume Maillard, a French developer is one of the founders of Blue-Eyed OS and the current Team Leader, and has done a good deal of the work in the graphical front "In the beginning of 2001 Be Inc. was almost completely silent and they had stopped the development of the OS," said Maillard. So he began researching X11. After months of investigation and some small prototypes in July, 90 percent of the technical choices were done and major prototypes were ready.

Maillard hopes that Blue-EyedOS will demonstrate that a robust UI is possible under Linux. In fact, he's of the opinion that his design strategy will be able to outperform the original BeOS. Maillard also promises that the interface will be fully anti-aliased. "X11 has its own set of advantages as a place to begin, a lot of framework is done already on the kernel and X11, and it's just a matter of modifying them so that X11 pleases now. There isn't the need to do this from scratch," said Maillard.

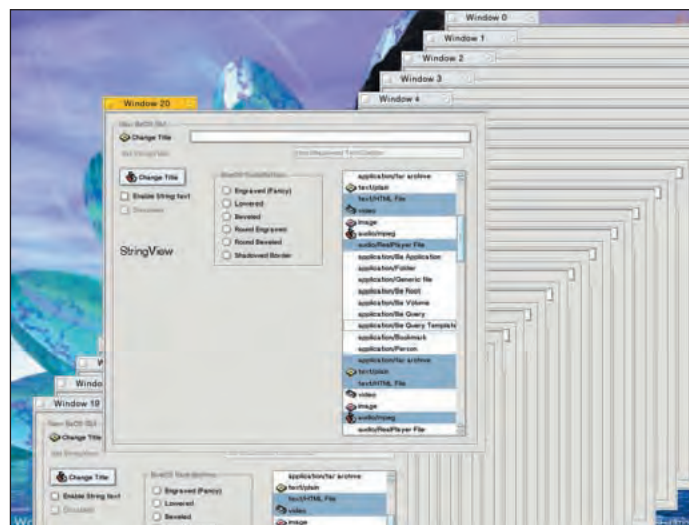
The plan for boosting the performance of X11 in B.E.OS sounds deceptively simple. Because X11 uses a server-client design, explains Maillard, if developers reduce the number of exchanged message, the latency is reduced. That's how B.E.OS manages fast window managing. It may sound odd but Maillard is quite serious about this, "I often see people saying that *XFree86* is slow. You can have this feeling, as I do, by using KDE or GNOME. But if you don't redraw point after point, much better performances are there," said Maillard.

Beyond the recompiled BeOS applications, Linux users will find that the pool of X11 apps should be available. "There are no restrictions, X11 apps will be integrated in the B.E.OS rendering model by creating a fake **XWindowManager**. Anyway using *Qt* or *GTK* based apps under B.E.OS will not be recommended because it will 'break' the UI design consistency," said Maillard.

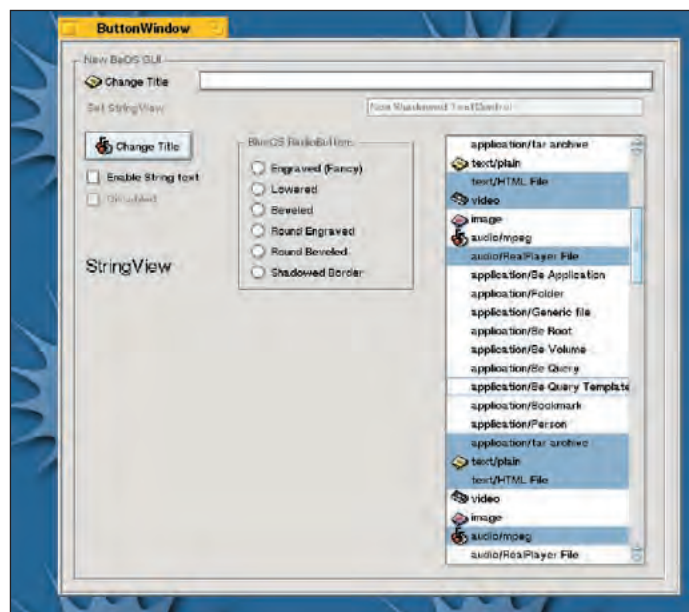
Maillard is skeptical about the need to recreate the BFS since a wide assortment of quality journalled filesystems have been brought to Linux in the last several years. *ReiserFS* and *XFS* are two prime candidates to replace BFS believes Maillard. "The biggest challenge in this project is the filesystem. The Be FS supported attributes and queries. But the Linux VFS doesn't support that, so we will have to emulate them and use the native attributes under *XFS*," said Maillard.

So, is it Linux? Will a typical Linux user find a system they'd be happy to move to? Maillard is optimistic. "Because I've had long, and often very technical discussions about Linux, *XFree86* and sometimes B.E.OS with both BeOS users and Linux users, I'm convinced that B.E.OS could be a great alternative," said Maillard.

Maillard has been writing extensively about B.E.OS and been frequently interviewed of late about his project and his surprisingly positive views about the potential of X11. From the outside it looks like an informal PR campaign to promote B.E.O.S. "I've only received positive feedback, a lot of people are impatient to see B.E.OS. I've received mail from Linux developers (KDE, and some great app developers). A lot of people have asked when will B.E.OS be complete, but I was not able to answer. Without developers strictly nothing will appear soon," said Maillard.



It may look and feel like the BeOS, but it's based on Linux and *XFree86*.



UI classes running on the original application server.

The short term goal for B.E.O.S is to finish the interface part of the OS and release something soon, perhaps in the new year. The project is still looking for coders who have a lot of time and motivation.

So, could this project fail? Not in

the least thinks Maillard, after all the source code will be available forever and he's certain that it will take up his time for the next ten years.

Blue Eyed OS plans to have a preliminary release available for download sometime in 2003.

## CONCLUSION

**Who knows, maybe with the** growing number of Linux kernel patches coming from corporations,

individual developers looking to make an impact on a project might bolster the ranks of these projects, where

their contributions are sure to be appreciated. If you're dissatisfied with the current choice of desktops, the

beauty of open source is that you have a chance to be involved in the shaping of the next batch. [LXF](http://www.linuxformat.co.uk)



# What on Earth is... HELIX?

Multimedia over the Internet used to be a challenge to setup and use, but **Robert Smith** looks at the latest incarnation from Real Networks to solve the client problem.

## »» What is Helix?

Helix is the latest offering from Real Networks, to allow the easy distribution, development and viewing of digital multimedia across the Internet. It aims to support all the major current media types, along with some of the smaller ones and newer ones, to produce a universal platform for all streaming multimedia on the Internet.

## »» OK once again, but in English :

Finding a multimedia player for streaming digital media has never been a hard task, but finding the correct one is. There are the big guns such as *Real One Player* (Real Networks), *Windows Media Player* (Microsoft) and *Quicktime Viewer* (Apple). There are also the smaller players such as *win-amp* and *XMMS* which allow you to effectively listen to audio streams across the Internet. Each of these programs support certain types of media, which means that to effectively view or listen to wide sources of digital multimedia across the Internet, you need at least the first three viewers. Here we hit the first major problem, operating system support. The only program available for Linux and most Unix operating systems is *Real Media Player 8*, which is the older version of the *Real One* player. Due to a large company's tactics, it is doubtful that we will ever be able to get Windows media player for Linux, nor, as Apple are currently trying to promote their own version of Unix, a version of *Quicktime*.

## »» So Where does Helix Fit in with this?

Helix intends to be "the" platform for viewing, producing and serving all digital multimedia across the Internet. Instead of having to have a separate

application for each type of multimedia available, Helix will simplify this by providing the *Helix DNA Client* which allows you to view multiple different types of multimedia available hosted using their *Helix DNA Server*. This doesn't mean that you will now be able to view *Quicktime* or Windows movie files, but if more media companies distribute their projects through the native Helix production suite because of its advantages, you will be able to view and listen to more media.

## »» Helix is a Viewer Program then?

No, Helix DNA is a multimedia framework developed by Real Media to be the best solution for digital media distribution on the Internet. The actual applications consist of the *Helix DNA Server* (to serve all the digital content), the *Helix DNA Client* (to view all the information served by the *Helix DNA Server* and other servers) and the *Helix DNA Producer* (to produce Helix compatible files for serving).

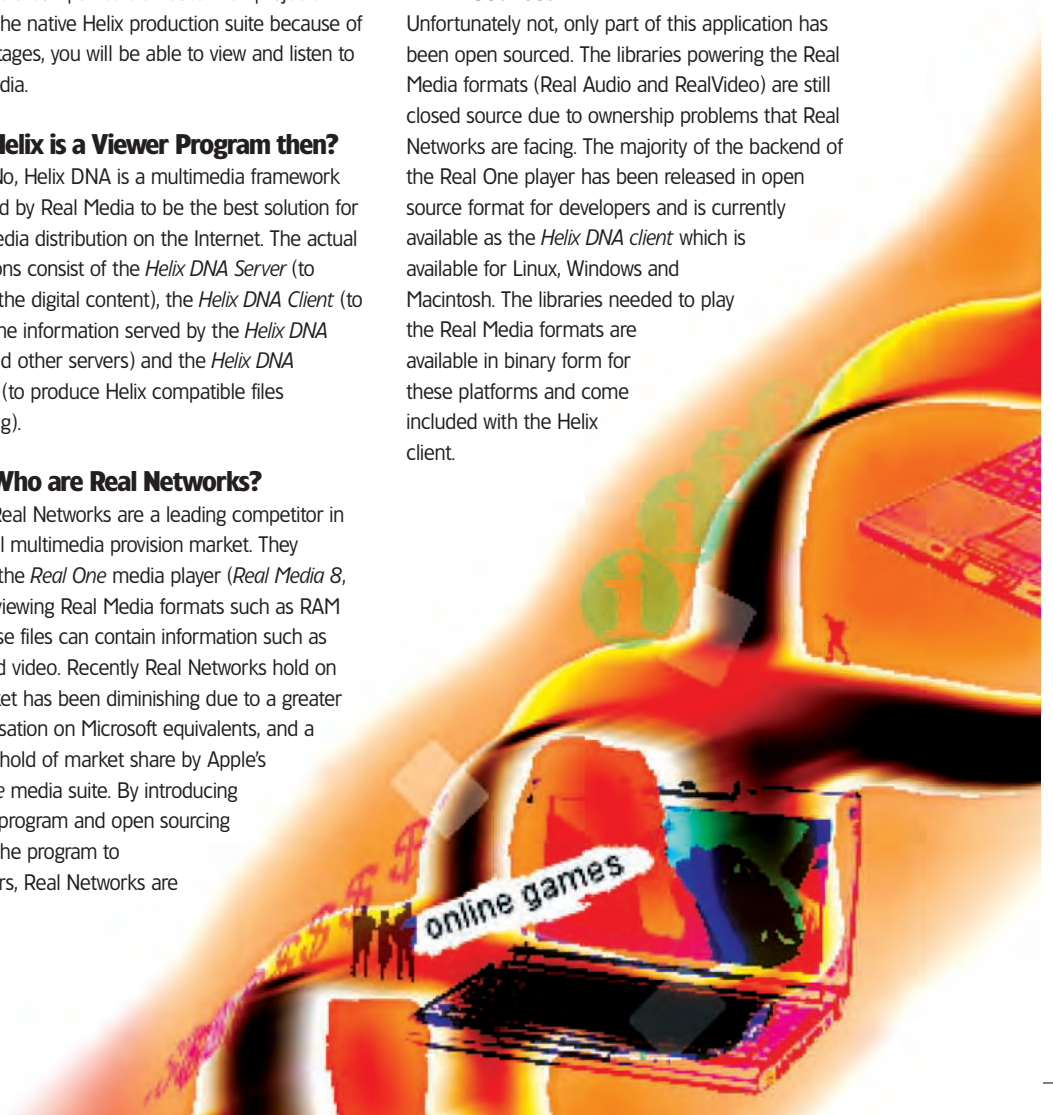
## »» Who are Real Networks?

Real Networks are a leading competitor in the digital multimedia provision market. They produce the *Real One* media player (*Real Media 8*, etc.) for viewing Real Media formats such as RAM files. These files can contain information such as audio and video. Recently Real Networks hold on this market has been diminishing due to a greater standardisation on Microsoft equivalents, and a constant hold of market share by Apple's *Quicktime* media suite. By introducing this new program and open sourcing parts of the program to developers, Real Networks are

attempting to utilise the large open source developer base instead of having to do all the development themselves, along with improving adoption of their new system and platform support.

## »» So Real One Player is now Open source?

Unfortunately not, only part of this application has been open sourced. The libraries powering the Real Media formats (Real Audio and RealVideo) are still closed source due to ownership problems that Real Networks are facing. The majority of the backend of the Real One player has been released in open source format for developers and is currently available as the *Helix DNA client* which is available for Linux, Windows and Macintosh. The libraries needed to play the Real Media formats are available in binary form for these platforms and come included with the Helix client.



# WhatOnEarthHelix

## » So only Linux, Windows and Macintosh will be supported?

One of the main reasons that Real Networks have open sourced the media framework is so that development to other platforms can be done in a shorter amount of time. If a platform is not supported and you are a developer you can join the Helix community and help port the application to the platform. This means that support for Helix can be provided for all architectures and operating systems from Palm pilots to the Sharp Zaurus.

## » So what will I be able to view on the Helix DNA Client?

There are restrictions upon what sort of media you can view on the Helix client. These restrictions will be reduced as more open source development takes place and more Helix

Networks are also working with Xiph.org to produce an Ogg vorbis plugin for the client. If the community development environment takes off, then plugins for wav files, jpeg, gif, png, etc. could

initiation of a multimedia session. All this data can be transferred across the Internet in either TCP/IP format or UDP format; this will allow a larger number of devices and products to use the Helix DNA client.

## » This sounds a lot like Real One Player?

The backend powering the *Helix DNA Client* is almost identical to that of *Real One Player*. The major differences are the added operating independent features available on *Real One Player* such as the DVD/CD burning utility, CD Ripping, play lists and a more advanced interface. These features were deemed unnecessary for the product as they would complicate porting to other hardware/software combinations and would not improve the product as a media viewer. The open source licence means that these features can be implemented for specific operating systems by a greater base of programmers. In the same way that *Galeon* uses the *Mozilla* rendering engine *gecko*, a program under Linux could implement all the features missing from *Helix DNA Client* whilst using the Helix DNA backend for the media interpretation.

## » The Helix Initiative

The Helix initiative is made up of three major areas: The Helix platform, the Helix community and the Helix product suite. The Helix platform is the product of seven years of research and development by Real Networks, and consists of the three main programs (client, server and producer). The

become available in the near future making it more than just a useful audio and video tool. The Helix DNA client also provides tools for audio mixing, re-sampling audio data streams and volume handling/management.

## » What transfer protocols does the Helix Client support?

Different types of transfer protocol are used to transfer different types of data; for web pages the http protocol is used and for large files ftp is used. Like the rest of the web, the digital multimedia sector has a set of its own protocols to achieve optimal transfer of data across the Internet. The Helix DNA client supports the Real Time Streaming Protocol (RTSP) for multimedia presentations, the Real Time Transport Protocol (RTP) a packet format for multimedia data streams, the RTP Control Protocol (RTCP) to monitor service quality and transfer any information from the client to the server and the Session Description Protocol (SDP) to help

client plugins are made available from the community project. The current version of the product supports, MPEG layer 1 or 2 and H.263 via open source protocols, and Real Audio and Video versions from version 8 to the latest version G2 available via binary plugins. Real



# WhatOnEarthHelix

community is the method that allows you to get your hands on the source code to use in development and help with other projects, backed and supported by Real Networks. The product suite is the main commercial side of the venture, it includes commercial versions of the producer, client and server modules allowing you

your name and a chosen user name. Once you have done this a link is sent to your email account giving you an URL to visit to confirm your login. You then will be asked to read and agree to both a site use licence, a copy of the licence that you choose and EULA for the binary Real Media support. Once you have successfully acknowledged these contracts, you will then be able to login to the Helix community website and download the source and binaries available.

## As a developer what will I receive?

Primarily you will receive access to *Helix DNA Producer* and *Client* sources. You'll also get the binaries required to support Real Media products (currently Real Audio and Real Video. Real Text and Real Pix support may be available later), so that you can build support for these formats into any product you design. You will also gain access to IRC channels, mailing lists and information on the TCK.

## What is the TCK?

The TCK is the technical development kit supplied by Real Networks to aid in the correct development of Helix DNA compatible programs, modules and patches. It is a test suite, that checks that any application developed using Helix DNA technology is interoperable with other programs using the same backends. This will mean that if you develop a program, it can be checked to see if it is still compatible with the main products. When developing under the RCSL (commercial development, etc.) your software will have to comply to these standards before release is allowed.

## What licence is the program source released under?

Real Networks have released the sources for the Helix DNA media infrastructure under their own open source licences. There are two main licences, the Real Networks Public Source Licence (RPSL) and the Real Networks Community Source Licence (RCSL). The first licence is open source, and has been submitted to the open source initiative to check that it correctly follows their guidance. The Helix DNA client and producer are available under this licence. If you develop the source under this licence then all programs using this source must be open source as well, using the same licence. The RCSL is almost identical to a traditional commercial source licence. This licence allows you to develop and use the code for free if you are doing research or development but there is a licence fee when you distribute the code. All three of the Helix

to do more as a business or company.

## Where can I get the Helix source code from?

As part of the Helix DNA project, Real Networks have set up a community web site at [www.helixcommunity.org](http://www.helixcommunity.org). At this web site you can sign up to be a community developer, and therefore qualify to gain access to the source code for development and research purposes. When you join the community you can gain access freely to the source code licensed under the real networks public source license. To sign up to the community you will have to give Real Networks a working email address,

DNA programs are available under this licence including the server, and with the producer and client you can choose which licence is more applicable.

### »» What sort of things can you do with the Helix DNA framework?

Everybody knows that if you have a fast enough connection to the Internet, then you can view streaming information broadcasts across the web. The BBC makes great use of these facilities on their websites, especially the news part, where you can download news clips in both audio and video mode. You can also log onto a web radio station. There are hundreds out there, so do a quick search on Google. The world of digital multimedia is just beginning to become big as the government pushes for the UK to go broadband. With small bandwidth modem connections, what you can actually do is limited, but with a broadband connection you can audio and video conference, play multimedia games across the net, and view even better quality videos. In the future if the plugins are created for the client and server, you could even look around a virtual 3D store and check out what you are about to buy securely and easily in 3D using the Helix DNA client.

### »» So how long will I have to wait till my mobile phone has a Helix DNA client?

This depends on how the whole IT industry takes to the Helix DNA idea and solutions. If the product is made available in such a way as to benefit hardware and software developers, then the next version of a mobile phone may well have the ability to watch movies. Already PDAs have the ability to run movies using other software, the open source backing may be the influence needed to see all PDAs running the *Helix DNA Client*.

### »» I already have a media client that I like, why should I change?

If your media client is open sourced, then changing will probably not be to a much greater advantage – in the future your client may become Helix DNA compliant, and boast some of the features offered by the system. If you are using a commercial closed source application, then you will probably

find changing, a step away from being tied in to continuous and costly upgrade cycles. By using the Helix client or offspring from the project with the same backend, you will not only gain the ability to break free of this cycle, but also will benefit from a host of new features offered by one of the industries leading companies. Then again, if all you ever intend to do is listen to Ogg files, is there any point in getting a new program for it?

### »» So how much will this cost me?

The success of the Helix DNA project depends on wide spread adoption of the product. This means that the browser, server and producer will be available in limited editions for free. The client program will always remain freely available, much as is the *Real One* player, so that you can view the media hosted on the new platform. The producer is currently available for free as a limited features product. The features available in this basic version include conversion to Real Video 9 and Real Audio 8 from other formats, production of three different rate samples for different connection speeds and output to a single server. This is a small feature set in comparison to the *Plus* version of the program, but is enough to demo the helix infrastructure and get a feel for the system. The server version is available in source code for further application development for specific web applications and a demonstration version is available for download. After you have used the demo version, the server is available at a variety of prices starting around \$2000, depending on the services that you require.

### »» What advantages does the server give my company?

To support 90% of the digital media market, a company needs to deliver three different streams of data. Each of these media types used to require a separate server setup to stream them across the Internet. In the past to distribute *Quicktime* you needed a server product from Apple, for windows media streams you required Windows Media technology server, and to distribute Real Media you required their server architecture. To provide this service it used to cost money in three ways, the server hardware, the operating system licence and the server product. The *Helix DNA Server* reduces the capital required to serve all these media formats across the Internet, as it removes the need for separate servers. Each *Helix DNA Server* provides support to stream all the three major media types simultaneously, and so can reduce the cost of the server products. The *Helix DNA Server* will also be available for more than just Microsoft OSs, so again this will reduce the budget requirements, as instead the stream could be run from a Linux or Unix box instead. By using the Helix DNA server you would therefore hypothetically save both money on hardware (as you would only run one process instead of three separate servers), on server technologies and on maintenance. The security of the service provided could also be improved, by using a generally more secure OS such as Linux.

### »» What does the Helix DNA producer allow me to do?

For digital multimedia to be successful the recordings that users receive have to be of a high quality. The *Helix DNA Producer* has many features to allow you to provide high quality media. An Inverse

Telecine Filter removes any extra frames added to a video during transfer to a digital production environment from a video; this helps when reducing the file size of an encoded file. Firewire (IEEE 1394) live capture support is integrated into the producer to allow capture of video along with encoding live content from a DV camcorder. The producer helps prevent data loss during transfer using its "loss protection" system that sends additional information in each packet to allow reconstruction of lost frames. Along with advanced audio re-sampling these features allow the production of better video media for streaming across the Internet.

### »» Is this the start of another monopoly?

Real Networks have recently seen a reduction in the usage of their services and a corresponding increase in the use of equivalent services, mainly the Microsoft equivalent. Although they still hold the majority of the digital media streaming and service provision market, their margin is reducing quickly. By open sourcing their products and allowing other developers to do their work, and getting more people aware of the software, Real Networks are attempting to re-address the balance of market share. If this scheme is successful they may gain a greater share of the market, and therefore will have more control over the development of the digital media sector. The decision to open source their products means they do not have complete control over how their products are developed, and what the developers do. In many ways this is the same as VA Linux and their OSDN network; they have control over the entire network, but trying to take too greater advantage of their position would cause a massive backlash and their company would no longer have the ability to survive. Real Networks could face the same problems if they try to go against the open source developers who have helped them gain the monopoly, if indeed they do.

### »» Great so where can I find out more?


The best site to look for open source development information is the helix community website at:

[www.helixcommunity.org](http://www.helixcommunity.org)

For information about the commercial products available from this scheme you can checkout the Real Network company website at:

[www.realnetworks.com](http://www.realnetworks.com)

### »» Is this the future for Digital Multimedia across the net?

Perhaps, but as with most technology, it is only time that will tell. The open source nature of the program may encourage some people to adopt the technology, along with those who live for the latest technology. If major hardware and integrated systems vendors adopt Helix DNA, then the chances of adoption are ever increased. 





# Tutorials >>

Our experts offer help and opinions on a whole host of Linux applications

## Your guide to getting things done!

Whether you are just starting out in Linux, or an experienced veteran, there's always more to learn. Every issue of *Linux Format* is packed full of practical advice, and nowhere is it more concentrated than in our tutorials pages.

Here you'll find expert guides to all sorts of things, from Basic Linux usage to understanding and deploying network solutions, from simple script coding to the complexities of Perl regular expressions, Java server apps and more. We aim to bring a good mix of tutorials to each issue, but if you have any suggestions for topics you'd like us to cover, why not contact us, by post, by email ([linuxformat@futurenet.co.uk](mailto:linuxformat@futurenet.co.uk)) or log on to our website and post your suggestions in our special forums? ([www.linuxformat.co.uk](http://www.linuxformat.co.uk)). Hope to hear from you soon!

**Nick Veitch** EDITOR

## THIS MONTH TEACH YOURSELF...

### System programming

This popular series concludes with a look at sockets and how programmers can make good use of them **p70**

### Perl

File manipulation and directory searching, the Perl way. Master your filesystems! **p74**

### Corba

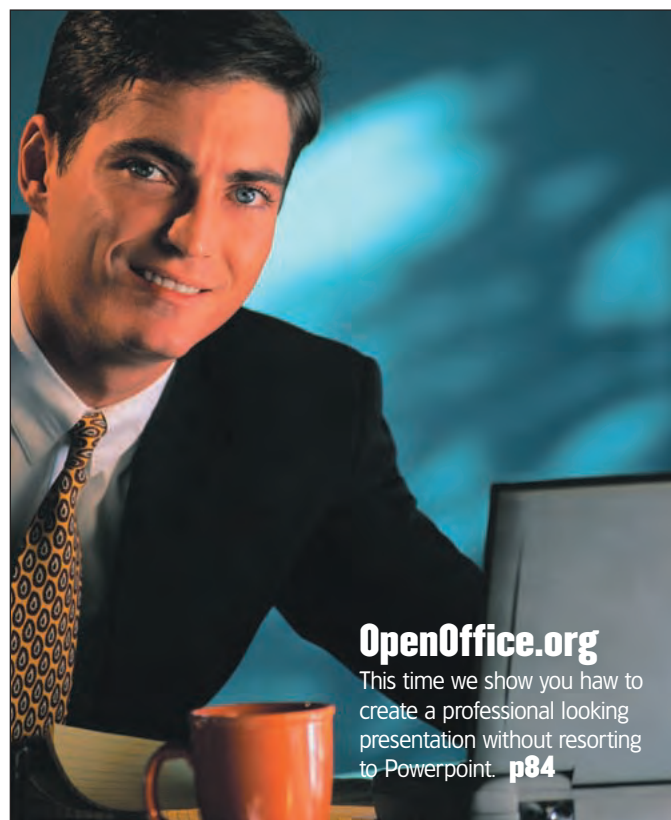
Middleware without SOAP. Find out how to integrate disparate web services with a little cross-platform magic **p76**

### Java

Turning our widgets into beans opens the door to reusability for Java coders **p82**

### PHP

It isn't just for Hypertext – find out how PHP can take care of your images too! **p88**



### OpenOffice.org

This time we show you how to create a professional looking presentation without resorting to Powerpoint. **p84**

## How code is represented

**Including code in magazines can be tricky, but we hope our notation will help it become clear. When lines are too long for our columns, the remaining text appears on the next line in a solid blue box:**

```
procedure
TfrmTextEditor.mniWordWrapClick
(Sender: TObject);
otherwise, there is usually a gap
between lines:
begin
mniWordWrap.Checked := not
end;
Usually, you'll find the code on
our CD/DVD too.
```

## TIP OF THE MONTH!

As well as providing you with handy tips for commands and software you do know, we'd like to think that this small corner of the magazine can introduce you to the occasional gem hidden away on your Linux installation that can be used for all manner of purposes.

`lsdf` probably falls into the latter category. The name is short for List Open Files, and, not surprisingly, that's exactly what it does.

Running the command with no options simply lists every file that is currently open. This is likely to generate a few screenfuls of

## Listing open files

information on most systems.

Fortunately there are plenty of options to focus the output!

As Linux treats devices and sockets as special types of file, `lsdf` is capable of listing all files open to a particular device. For example:

```
lsdf /dev/hda5
```

will list all files open on the specified partition. Useful if you wish to unmount a partition for whatever reason while not interrupting any processes (or risking your data).

As we mentioned, it can also monitor sockets, which has plenty of potential for intrusion detection, as

well as just service monitoring.

```
lsdf -i
```

lists all internet sockets, including the processes, user, PID and node details.

You can refine the search to a particular port name too. For example, on a system running a web server, you might expect the command

```
lsdf -i :80
```

to list several instances of httpd listening on this port.

There are of course, plenty more ways to use this tool. As it provides standard output, it can be grepped or used as part of a script. Check the man page for more info.

## NETWORK PROGRAMMING

# Programming with Sockets

Chris Brown rounds off the series with a look at sockets.

**T**his month we're going to look at the system calls that allow us to communicate across a network. These calls, which go under the general name of "sockets", were invented by the folks at the University of California at Berkeley, to provide an application program interface to the transport services offered by the then-newly-invented TCP/IP and UDP/IP protocols. They have underpinned \*nix network programming ever since.

We're going to restrict our attention to the connection-oriented protocol, TCP, and ignore its connectionless cousin, UDP. TCP is, arguably, easier to use, because once a connection has been established it provides two communicating processes with a reliable communication channel identified by a regular Unix file descriptor, which can be read and written, duplicated, closed, and so on, just like the descriptors we've seen earlier in the series. The reliable communication channel is sometimes called a virtual circuit, and is really an illusion, much as the telephone network provides the illusion of a length of copper wire connecting two telephones.

## The Client/Server Model

When sockets are used to establish a connection between two programs across a network, there's always an asymmetry between the two ends. One end (the server) creates a "communication endpoint" at a known address, and waits passively for connection requests. The other end (the client) creates its own communication endpoint and actively connects to the server's endpoint at its known address. Once communication is established, **read()** and **write()** calls may be used to send data between the client and the server. Typically, the client sends some sort of request or command, and the server send back a response. The detailed format of this interchange is specified by a set of rules for that particular service, which we call the application protocol.

A server's endpoint is identified by two numbers. First, there is the IP address of the machine where the server is running. Second, within each machine, each end-point is identified by a 16-bit port number. To connect to a server, the client needs to know both these numbers. For standard network services, there is a fixed association between the service and the port number on which it listens. For example, *SWAT* (the *Samba Web Administration Tool*) listens on port 901, and ftp servers listen on port 21. The standard associations between services and port numbers are held in the file `/etc/services`. As we'll see, clients can look services up by name in this file to discover the port number they should connect to.

**Fig 1** shows the steps taken by both server and client to establish a TCP connection, along with the system calls used at each stage. A telephone-based analogy helps explain the sequence of events.

Let's suppose that Pete (our server) is planning to offer a dial-

a-joke service. First, he must get himself a "communication endpoint" (a telephone). Analogously, the server begins by creating a socket. Next, the server needs to associate, or "bind", its pre-defined port number to the socket. This is analogous to Pete getting himself a nice memorable phone number for his phone. (One of my local airport taxi services, for example, has the phone number 2747747; alternatively in the USA, where people still take some notice of the letters associated with each digit on the phone, he might try for a number like 1-800-HEAR-JOKE. In any event, Pete needs to have a phone number which his potential clients know about.) Armed with his joke book and his phone, Pete now enters what we might call his "main service loop". At the top of this loop, he sits and waits for his phone to ring. Similarly, our service enters a loop and blocks on an **accept()** call at the top of the loop. This call waits for, and accepts, a connection request from a client.

## Meanwhile, back at the client...

This is probably a good time to consider what happens at the client end. Jean has decided to call Pete's service. First she needs a phone – her own communication endpoint. Analogously, a TCP client also needs to create a socket. Jean's phone will have a phone number (there's no such thing as a phone without a number) but, for the purpose of this call at least, no-one needs to know what it is. (Pete is not going to call Jean back.)

Analogously, our client needs a port number bound to its socket, but doesn't care which. Consequently, there is usually no explicit **bind()** operation at the client side. Instead, the client will simply go ahead and use the socket, and the system will bind an (essentially arbitrary) port number to the socket on the client's behalf. Jean now picks up her phone and dials Pete's number. Analogously, the client performs a **connect()** call, specifying the IP address and port number of the server it wishes to connect to.

Pete's phone rings and he stirs from his slumber to answer it. Analogously, the server retrieves the connection request and the **accept()** call returns. Now we have a virtual circuit established between the two parties. In Jean and Pete's case, each is holding a phone. In the case of our client and server, each has a file descriptor referencing the connection. Now begins some sort of dialogue. Jean might ask "Tell me a triple pun", and Pete might reply "There was once a rich Texan who bought a cattle ranch for his four boys. He called it Focus, because it's where the sons raise meat". (Think about it ...) This interchange may be repeated, until Jean has heard as many jokes as she needs. Then, if she's polite, she'll say "thankyou" and "goodbye" and both parties will know to put the phone down. Similarly, in a properly designed application protocol, both client and sever will know when the message exchange is complete, and will close the connection. If the client unexpectedly closes its end of the connection, and the server tries to keep reading, it will receive an "end of file" indication from the **read()** call.

At the client end, we're pretty much done now. Pete, however, keeps his telephone (with the same tel number) and sits down to wait for another call. Similarly, a server will usually return to the top of its service loop and call **accept()** again to wait for another client.

## The Server Code

Well, we can't postpone looking at some real code for any longer. Because our focus is on the system calls used to establish a TCP connection, we're going to implement a trivial service which simply reads an integer value from a client, squares it, and writes the result back. Line numbers are for reference, they are not part of the code:



```

1 /* Trivial TCP service to square integers */
2
3 #define SQUARE_PORT 1888
4
5 #include <stdio.h>
6 #include <sys/types.h>
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9
10 /* Useful function to create server endpoint */
11
12 int create_tcp_endpoint(int port)
13 {
14     int sock;
15     struct sockaddr_in server;
16
17     sock = socket(AF_INET, SOCK_STREAM, 0);
18     if (sock < 0) return -1;
19     server.sin_family = AF_INET;
20     server.sin_addr.s_addr = htonl(INADDR_ANY);
21     server.sin_port = htons(port);
22
23     if (bind(sock, (struct sockaddr *)&server,
24             sizeof(server)) < 0)
25         return -2;
26     listen(sock, 5);
27     return sock;
28 }
29
30 int main()
31 {
32     int fdnet, value, sock;
33
34     /* Create an endpoint to listen on */
35     if ((sock = create_tcp_endpoint(SQUARE_PORT)) < 0) {
36         fprintf(stderr, "cannot create endpoint");
37         exit(1);
38     }
39
40     /* Enter our main service loop */
41     while (1) {
42
43         /* Get a connection from a client */
44         fdnet = accept(sock, NULL, NULL);
45         printf("Got a connection on fd %d\n", fdnet);
46
47         /* Read the incoming integer */
48         read(fdnet, &value, sizeof value);
49
50         /* square it */
51         value = value * value;
52
53         /* and return it to the client */
54         write(fdnet, &value, sizeof value);
55
56         /* close the connection and wait for another */
57         close(fdnet);
58     }
59 }

```

At line 3, we define a symbolic constant for the port number our server will use. Apart from picking one which was not in

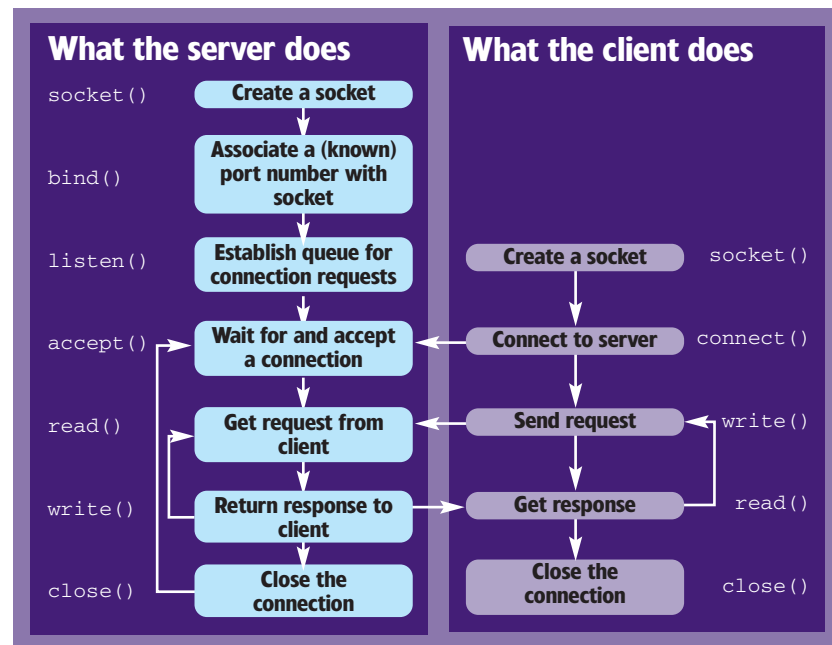


Fig 1: **Connection-oriented server and client operations.**

/etc/services (making it fairly unlikely that another server will already be using that port), the choice was arbitrary. When we write the client, we'll see a different method of defining the server's "well-known" port which avoids hard-wiring it into the code.

The code to establish an endpoint to listen on is pretty much "boiler plate" (that is, it looks the same every time) and we have encapsulated it into a handy function called `create_tcp_endpoint()`. The function takes an argument specifying the port number to be bound to the endpoint. The endpoint is created by the call to `socket()` at line 17. The first argument, which here has the value `AF_INET`, meaning "Internet address family," tells Linux that this socket will use TCP/IP addressing. Here we meet an interesting design feature of the sockets API ... it doesn't make any assumptions about the network protocol that's going to be used, or (by implication) the exact form of an "endpoint address". Other legitimate values for this argument, for example, include `AF_IPX` (indicating the Novell IPX protocol family) or `AF_X25`, indicating the old ITU X25 protocols. We will see other instances of extra complexity in the sockets API as a result of this protocol-independent nature of the design.

The second argument, `SOCK_STREAM` in this example, means that we want a connection-oriented socket, using TCP. Other values you may see here are `SOCK_DGRAM`, meaning a connectionless socket using UDP, or, more rarely, `SOCK_RAW`, which gives access to the IP layer directly and requires you to build your own headers for whatever the overlying protocol is. You might use raw sockets, e.g., if you were writing a version of ping, which uses the ICMP protocol, or, less honourably, if you were trying to crash a machine's TCP/IP stack by deliberately sending malformed TCP packets.

## Getting into a bind()

Next we need to bind our "well-known" port number to the socket. This is conceptually trivial and it would be nice if the call was simply:

```
bind(sock, port);
```

but unfortunately the protocol-independent nature of the sockets API complicates the details. We need to declare a (protocol-specific) data structure called a `sockaddr_in` (standing for internet socket address), and fill in its three fields, as at lines 19–21. Line 19 tags this structure as belonging to the "internet address family". Line 21 defines the port number. Line 20 takes a bit more explaining. >>

# TutorialSystemProgramming

◀ In this context, putting in a value of **INADDR\_ANY** means that this server is willing to accept connection requests received on any of its network interfaces. For most machines, which have only one network interface and only one IP address, the point is moot, but for so-called multi-homed hosts (*i.e.* with interfaces to more than one network) this parameter lets you specify that connection requests are to be accepted from only one of the interfaces. Contrary to what you'll sometimes read, this parameter has nothing whatever to do with the client's IP address.

Once this structure has been filled in, we pass its address to a **bind()** call at line 23. It's important to test the return status of this call to verify that it succeeded. In particular, it will fail if some other server is already listening on this port. Of course, this "shouldn't happen" if all servers are correctly using their registered port numbers, but occurs, for example, when you're trying to start a new version of a server and the old one is still running.

The final step is to establish a queue for connection requests on this socket by calling **listen()**. The name is misleading, it suggests that this call actually blocks listening for connection requests, but this is not the case. It simply sets the socket up to receive connection requests. The second parameter to this call specifies the length of the queue of requests. Setting it to **5**, as I do here, marks me as a member of the "old school" because **5** was the maximum supported under the original BSD socket implementation. The limit for Linux is much larger (128 if you believe the man page.) This queue is used to hold incoming connection requests from new clients if the server is currently busy (dealing with the current client presumably). If a client tries to connect and the queue is already full, its connection request will be refused.

Finally, our **create\_tcp\_endpoint()** function returns the descriptor of the socket it has just created. I call this the "rendezvous descriptor" – they one from which we can retrieve new connections.

## Application-specific code

Now let's look at the **main()** function of our server. It begins by creating a connection endpoint at line 35, using the function we just examined. At line 41 it enters its main service loop. At the top of the loop it blocks on an **accept()** call, which waits for (and accepts) a connection request from the client. (With sockets, you can't separate the act of retrieving the connection request from the act of accepting it. The most unsociable you can be is to accept the connection, decide you don't like the look of the client, and immediately close the connection. This is like *not* having Caller ID on your phone. You can't find out who's calling without picking up the phone.)

The **accept()** call at line 41 returns a new descriptor, **fdnet**, which I call the "connection descriptor". This is the descriptor which

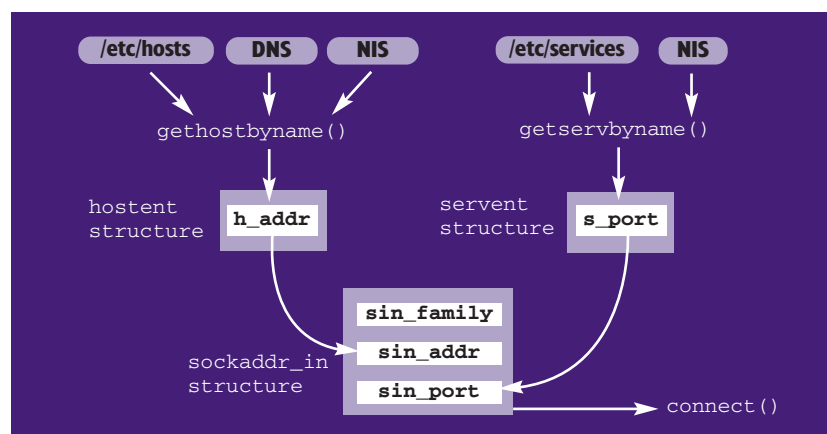
the server uses to talk to the client, as we see in lines 48 and 51. This particular service is pretty simple, reading an integer value from the client, squaring it, and writing the result back. After this, the server closes the connection descriptor and loops around to wait for another connection request on the rendezvous descriptor.

## The Client Code

```
1 /* Client for square service */
2
3 #include <stdio.h>
4 #include <sys/types.h>
5 #include <sys/socket.h>
6 #include <netinet/in.h>
7 #include <netdb.h>
8
9 int tcp_connect(char *host, char *service)
10 {
11     struct hostent *host_info;
12     struct servent *serv_info;
13     struct sockaddr_in server;
14     int sock;
15
16     /* Create a socket */
17     sock = socket(AF_INET, SOCK_STREAM, 0);
18
19     /* Look up the host's IP address */
20     host_info = gethostbyname(host);
21
22     /* Look up the service's port */
23     serv_info = getservbyname(service, "tcp");
24
25     /* Copy these into a socket address */
26     server.sin_family = AF_INET;
27     memcpy((char *)&server.sin_addr, host_info->h_addr,
28            host_info->h_length);
29     server.sin_port = serv_info->s_port;
30
31     /* Now make the connection */
32     connect(sock, (struct sockaddr *)&server, sizeof server);
33     return sock;
34 }
35
36 int main(int argc, char *argv[])
37 {
38     int fdnet, x, y;
39
40     /* Connect to service */
41     fdnet = tcp_connect(argv[1], "square");
42     printf("fdnet = %d\n", fdnet);
43
44     x = 5;
45     /* Conduct brief dialogue with server */
46     write(fdnet, &x, sizeof x);
47     read(fdnet, &y, sizeof y);
48     printf("%d squared is %d\n", x, y);
49 }
```

On the client side, we've also encapsulated most of the boiler-plate socket code into one handy function, this time called **tcp\_connect()**, defined at lines 9–33. It takes two arguments: the name of the host on which the service is running, and the name of

Fig 2: Finding the required service.





the service. For example, the call `tcp_connect("neptune", "ftp")` would try to connect to the ftp server on the host neptune.

This function begins by creating a socket, just like the server does. Next the client needs to fill in a `sockaddr_in` structure to specify the IP address and port number of the server it wants to connect to. To do this it must look up the machine and the service by name in the relevant configuration files. In the simplest case, it looks in `/etc/hosts` and `/etc/services`. Alternatively, hostname-to-IP translation may be provided by a naming service such as DNS or NIS. Similarly, service name to port translation may come from a central NIS server. Fortunately, the complexities of this are hidden behind cunning library functions called resolvers. For example, `gethostbyname()` (called at line 20) takes a hostname and returns an IP address; hiding the details of how naming services are provided on that particular machine. Similarly, `getservbyname()` hides the details of how service to port translation is handled. The code which copies the data out of the structures returned by these resolvers into the `sockaddr_in` structure (lines 27–28) is a little complicated. Hopefully **fig 2** will help you get the big picture of what's going on here even if you can't figure out every \* and & in the code.

With the `sockaddr_in` structure filled in, at line 31 the client calls `connect()`, passing a pointer to the address structure. If successful, this call returns a file descriptor which the client uses to communicate with the server. Notice that in the server there are two descriptors, which we called the rendezvous descriptor and the connection descriptor, whereas in the client there is only one.

The application-specific code in our client is contained in the `main()` function at lines 35–49. It begins by calling `tcp_connect`, passing a host name (taken from the command line arguments) and a hard-wired service name, `square`. Since `square` isn't a standard service, for this to work you'll have to add a line into `/etc/services` to say that the square service uses port 1888. It would also work fine to hard-wire the port number into the code, as we did in the server. The benefit of looking it up at run time is that the port number can be changed without editing and re-compiling the source. With the connection established, the client writes a single integer to the server at line 45 and reads one back at 46.

## Our grand finale: A Web server

You could be forgiven for finding the `square` server a trifle dull. So let's finish off with something more worldly – a minimal, functioning web server. Here's the code:

```
1 #define WEBROOT    "/webroot"
2 #define HTTP_PORT  8080
3
4 #include <stdio.h>
5 #include <fcntl.h>
6 #include <sys/types.h>
7 #include <sys/socket.h>
8 #include <string.h>
9 #include <netinet/in.h>
10
11 void text(int f, char *s)
12 {
13     write(f, s, strlen(s));
14 }
15
16 int main()
17 {
18     int fdnet, fd, count, sock;
19     char filename[100], request[1024], reply[1024];
```


```
20
21     /* Create an endpoint to listen on */
22     if ((sock = create_tcp_endpoint(HTTP_PORT)) < 0) {
23         fprintf(stderr, "cannot create endpoint");
24         exit(1);
25     }
26
27     /* Enter our main service loop */
28     while (1) {
29         /* Get a connection from a client */
30         fdnet = accept(sock, NULL, NULL);
31
32         /* Read the GET request & build the file name */
33         read(fdnet, request, sizeof request);
34         strtok(request, " ");
35         strcpy(filename, WEBROOT);
36         strcat(filename, strtok(NULL, " "));
37         printf("Requested file: %s\n", filename);
38         fd = open(filename, O_RDONLY);
39
40         if (fd < 0) {
41             text(fdnet, "HTTP/1.1 404 Not Found\n");
42             text(fdnet, "Content-Type: text/html\n\n");
43             text(fdnet, "<HTML><BODY> Not found </BODY>
44             </HTML>");
45         }
46         else {
47             text(fdnet, "HTTP/1.1 200 OK\n");
48             text(fdnet, "Content-Type: text/html\n\n");
49             while((count = read(fd, reply, 1024)) > 0) {
50                 write(fdnet, reply, count);
51             }
52             close(fd);
53             close(fdnet);
54         }
55     }
```

This server uses the same boiler-plate code to create an endpoint as in the previous example. (We have not repeated the code for `create_tcp_endpoint()` in the listing.) The interesting part is the main service loop at lines 28–54, which implements a minimal subset of the HTTP protocol. At line 33 an HTTP request is read from the client. This will look something like:

```
GET /index.html HTTP/1.1
```

Lines 34–36 extract the filename from the **GET** request and append it to a pre-defined "web root" directory, yielding a full path name like `/webroot/index.html`. At line 37 this is printed out as a debugging aid. (Real servers wouldn't use their `stdout`; they'd write to a log file.) The file is opened at line 38, and (if this succeeds) the server returns an HTTP response header to the client (lines 46–47) followed by the content of the file (lines 48–50). If the file cannot be opened, an HTTP "not found" status is returned (lines 41–43).

This implementation is admittedly pretty minimal, but it does actually work, and I would encourage you to try typing it in and running it. Put a simple HTML file into the "webroot" directory, and point your browser at it. You'll learn a lot.

Well, that completes our rather rapid scramble through the world of system calls. There's a huge amount of detail we've not done justice to, and lots of excitement we've bypassed altogether. But hopefully you have some idea of the range of services provided by the virtual machine we all know and love – Linux. 

## FILE MANIPULATION

# Directory searching

Charlie Stross gets into filesystems and explains how to do fun things with directories and files

Perl's reputation as the Unix Swiss Army Chainsaw is pretty well-known, but when we work on a Linux system we tend to use a shell – either a pretty graphical file manipulation front-end such as GNOME's *Nautilus* or KDE's *Konqueror*, or a command-line shell such as *Bash* (which is also an interpreter for a fairly powerful string-substitution language).

One particular reason we use a shell is because they're good for manipulating files, and files are the lowest-level unit of data storage that we normally deal with directly. But Perl can do (albeit not very interactively) all the jobs we expect of a shell – and in many cases, it can do it faster and more elegantly.

First of all, let's look at some basics. The **open()** command lets us associate a file with a data structure called a *file handle*; options to the filename tell **open()** whether to open the file for reading, writing (truncating it to zero length), appending (writing additional data without truncating it), or reading and writing. **open()** returns any error conditions in the special variable **\$!**. The file handle itself is a special type of variable that keeps track of an open file, and can be read from or written to using a variety of commands (**read()**, **print()**, **sysread()**, **syswrite()**, and so on). When a file needs to be flushed to disk and read/write operations are done, we call **close()** on the file handle.

For example:

```
my $tmpfile = "/tmp/tmpfile.$$"; # name of temporary file
my $line = ""; # scratch variable

open (TEMPFILE, ">$tmpfile") ||
    die "Error opening $tmpfile for writing: $!\n";

print TEMPFILE "This data is written to $tmpfile\n";

close TEMPFILE;

open (TEMPFILE, "<$tmpfile") ||
    die "Error opening $tmpfile for reading: $!\n";

print "-- Start of file\n";

while ($line = <TEMPFILE>) {
    print $line;
}

print "-- End of file\n";
```



```
close TEMPFILE;
```

## Fun with directories

Directories are a bit different from files. Originally, in the Unix world a directory was a special file containing a list of records. Each record was of a fixed length; the first 32 bits specified an inode number, and the rest of each record (up to the first null byte) was a filename associated with that inode. (An inode is a special data structure stored in a Unix filesystem that points to all the data blocks associated with a file and keeps track of the files' creation and modification times, among other things.)

Linux considers directories to be special files – you can't simply **open()** them or write to them. (This will be a relief to those of you who share your columnist's memory of discovering the hard way that on Solaris 2.4 and other Unixes it was possible to **open()** a directory and scribble all over it in blue crayon, thus wreaking havoc on a filesystem.) Part of the reason for this is that Linux supports a variety of filesystems in which a directory is *not* a simple list of inodes and filenames – *ReiserFS*, for example, or NTFS and HFS – and so a higher level abstraction is needed. But Perl still lets you get at the contents of a directory, by using the **opendir()**, **readdir()**, **telldir()**, **seekdir()**, and **closedir()** commands. Here's how we slurp a list of all the filenames in the current directory:

```
$dir = "."; # present working directory
@flist = (); # list of files

opendir(THISDIR, $dir) ||
    die "Unable to opendir($dir): $!\n";
@flist = readdir(THISDIR);
closedir(THISDIR);

print "There are ", scalar(@flist), " items in $dir\n";
```

This example is a bit crude – in particular, beware the consequences of doing it on a truly huge directory! (**@flist** will bloat up.) You may prefer to do something smart using **telldir()** and **seekdir()**. When you call **readdir()** in scalar context, it returns the next filename from the current directory. **telldir()** returns the current position of the **readdir()** location, and **seekdir()** seeks to that position in the current directory. (Unfortunately the value returned from **readdir()** and used by **seekdir()** isn't a straightforward integer – it's a magic internal pointer and you can't do useful things like increment or decrement it manually.)

## Testing file attributes

A filename is just a human-friendly name associated with an inode; inodes keep track of file contents. To examine an inode's attributes, Perl gives us two different mechanisms; the **stat()** command, and the file test operators. **stat()** first; inodes contain a



bunch of attributes, and **stat()** lets us examine these for a given filename or filehandle. **stat()** returns a thirteen-element array, containing: filesystem device number, inode number, file mode (permissions), number of hard links to the file, numeric user ID of file's owner, numeric group ID of file's owner, device identifier (for special files such as devies), file size in bytes, last access time, last modification time, last change time, preferred blocksize for file system I/O, and the actual number of blocks allocated to the file. All times are reported as seconds since the Unix epoch began (in 1970).

For example, to check the name of a file's owner:

```
$uid = (stat $filename)[4];
@user_pwent = getpwuid($uid); # get /etc/passwd entry for
user ID $uid
print "File $filename belongs to user ", $user_pwent[0], "\n";
```

The file test operators are somewhat different; they're unary operators that apply to a filename or file handle, and let us test various attributes of the file, returning their value: for example, the **-r**, **-w**, and **-x** operators test whether a file is read/write/executable by the current program's effective user ID and group ID, while the **-e** test examines whether a file exists, **-f**, **-d**, **-l** and **-p** test whether a file is a plain file, a directory, a symbolic link or a named pipe, and so on. These operators are boolean; you can chain them together using the logical OR and logical AND operators ('||' and '&&'). To save the overhead of a low-level **stat()** system call, Perl caches the previously-examined inode in a special variable called simply **'\_'**; so we can do tests like this:

```
if ( -e $file && -r _ && -x _ ) {
    # $file exists and is both readable and executable
    ...
}
```

## File manipulation modules

The built-in file test operators, directory and file read/write commands, and extras like **stat()** give Perl a firm base on which to mess around with files. But there are a number of recurrent tasks that frequently crop up when itemising files in directories, and many of these are handled by the core modules in the **File::<thingy>** hierarchy – some of which are distributed in the standard Perl distribution (as of 5.8.0).

Core File modules you can count on in a recent Perl include **File::Find**, **File::Compare**, **File::Basename**, **File::CheckTree**, **File::Copy**, **File::Path**, and **File::Temp**. Let's go through them and see what they do.

**File::Find** is actually the back end of the *find2perl* program, a tool which lets you recursively search filesystems for files that match some criterion. *find* works by letting you specify a sequence of criteria against which each file it encounters is matched; when files fail to match they're weeded out. For example:

```
find / -depth -type f -name 'ez' -print
```

Causes *find* to start searching from the root of the filesystem **/**, conducting a depth-first traversal. *Find* then rejects any files that are not of type **f** (ordinary file), rejects anything that doesn't match the pattern **ez\***, and prints the name of anything that's left over.

*find2perl* does much the same – but instead of *searching* for a file, it emits a block of Perl code that does the same job that *find* would. For example:

```
find2perl / -depth -type f -name 'ez' -print
```

emits the following:

```
#!/usr/bin/perl -w
eval 'exec /usr/bin/perl -S $0 ${1+"$@"}'
if 0; # $running_under_some_shell

use strict;
use File::Find ();

# Set the variable $File::Find::dont_use_nlink if you're using
AFS,
# since AFS cheats.

# for the convenience of &wanted calls, including -eval
statements:
use vars qw/*name *dir *prune*/;
*name = *File::Find::name;
*dir = *File::Find::dir;
*prune = *File::Find::prune;

sub wanted;

# Traverse desired filesystems
File::Find::finddepth({wanted => \&wanted}, '/');
exit;


sub wanted {
    my ($dev,$ino,$mode,$nlink,$uid,$gid);

    (($dev,$ino,$mode,$nlink,$uid,$gid) = lstat($_)) &&
    -f _ &&
    /^ez.*\z/s &&
    print("$name\n");
}
```

If you want to do something elaborate with the files you find using this routine, just replace **print("\$name\n");** with a call to your own subroutine.

The other common File utilities are less obscure.

**File::Basename** does much the same job as the traditional library routines **basename()** and **dirname()**; it lets you parse a filename, extracting the directory path component, file suffix, and file name, separately. **File::Copy** provides equivalents to the *cp* and *mv* commands, for copying and moving files. **File::Path** allows you to create and remove directory trees, including multiple nested directories in one go. **File::Temp** provides routines for generating the name and handle on a temporary file – one that is guaranteed to be unique to the current process. **File::Compare** allows you to compare the contents of two files or file handles, to check for similarity – it's equivalent to the Unix *cmp* command, not *diff* (which is supported by the non-core module **Text::Diff**, available from CPAN, if you need to do detailed differential analysis). And **File::CheckTree** allows you to run many file test operators in parallel on a tree of files; you'd use this if you want to check that a bundle of files you're distributing is consistent with the state that a program requires before it will run safely.

All these modules come bundled with the standard Perl distribution; there are a lot more **File::** modules on CPAN, but they aren't guaranteed to be present on all Perl installations. 

## MIDDLEWARE

# Tao of CORBA

Warren Brown guides us towards the dark side of distributed programming with an introduction to CORBA.



**C**ORBA (Common Object Request Broker Architecture) is the equivalent of bumping into a ghost in a dark house: you've heard that they're not really dangerous and some people even say they are fun to hang out with, ... but you'd rather not take the chance and run like hell! That's how most programmers feel when faced with CORBA for the first time.

It's not so much that CORBA is difficult, but the new way of thinking about objects being elsewhere can lead to headaches, nausea and sometimes death (of the programmer's spirit that is).

So what is so different about CORBA? Well, it's actually a specification that companies or individuals can use to develop an application programming interface (API). Some notable ones are *omniORB*, *ORBacus* and *MICO*. Unfortunately specifications can be interpreted differently by whoever reads them. Try it sometimes – write down your idea of the perfect partner and give it to some friends to work on. You will probably get vastly differing results! You wanted blond with blue eyes, but they got you a redhead with green.

The initial CORBA specification was 'ambiguous'. In your CORBA endeavours you may come across the term BOA (Basic Object Adapter), this has now been deprecated and replaced with POA (Portable Object Adapter), mainly due to the poor spec that the BOA had. The ORB component provides object reference, activation, and state related services to an object implementation. These may differ according to use.

## So, why use CORBA anyway?

Picture this... You start your new job as a C++ programmer – polymorphism, inheritance, operator overloading and so on, are pieces of cake, you can do it in your sleep. Then you meet your new boss, and it turns out that you are working at a company that uses C, C++, Java, Smalltalk and Python.

The Web Portal in London uses Java on Windows (yuck!), the New York branch uses Python on Linux doing various housekeeping tasks, and the insurance section based in Holland uses Smalltalk on AIX. Your office is mainly C/C++ and most of the functionality and business rules are on your servers. Now, because they've upgraded the Web portal, they would like customers to have access to their accounting information from a webpage. Instead of re-inventing the wheel, some bright spark suggests CORBA. This starts an avalanche of ideas, like why they need to send information via email to Holland and have it imported into the database there when they could also use the objects based at Head office. Needless to say, the Python guys, not wanting to be outdone, would like to give some of the info they have to the Head office and to the other sites. Guess what, it falls into your lap, because you're the new guy on the block, and the others know how to duck in the face of extra work.

## Middle what?

So what makes CORBA adaptable to this situation? CORBA is middleware – no not Middle Earth, that's where Frodo lives.

Middleware is the stuff that sticks/plumbs different systems together, i.e. one program can be written in Java and the other can be written in C++, and with the help of middleware they can talk to each other, or use each other. With CORBA you can have a class written in C++, and the other application, be it Smalltalk, COBOL, Java, or whatever, can instantiate the object and use it, like it would any other object. The key to all this, is a language known as IDL or interface definition language. This is not a programming language but a descriptive language. You use this language to define prototypes for your methods. The nice thing about IDL is that it is very much like C++ in its syntax.

The reason why we use IDL in the first place is that it is a neutral language, I can take my IDL and compile it for any language that implements the CORBA API. So, if I want to use it for C++, I would compile it using *idlnidl* (*omniORB*), if I were using Java, then I could use *idl2java* or *idlj*. The IDL compiler will then create the necessary files to allow us to create programs that can communicate with other disparate systems. Mapping of data types is done by the IDL as well, so we don't have to worry about **int** or **float** sizes, etc. We just use its definitions in our code.

## Coding time

It could be enlightening at this point to do some CORBA. We can start with a simple program written in C++, and then 'CORBA-ise' it.

```
#include <iostream>
#include <string.h>
#include <stdlib.h>
using namespace std;

class hello
{
public:
    hello() {}
    char* say_hello(void);
};

char* hello::say_hello(void)
{
    return strdup("Hello Cruel Corbaless World");
}

int main()
{
    char *data;
    hello mytest;
    data = mytest.say_hello();
    cout << data << endl;
    free(data);
}
```

If you want to compile this code, then save it as *hello\_simple.cc* and compile with:

## Object Adapters BOA and POA

The Basic Object Adapter – BOA – allows an object server to interact with the ORB. A server uses the BOA to inform the ORB when an object is ready to perform operations.

The Portable Object Adapter – POA – supercedes the Basic Object Adapter as the primary way of making implementation objects available to the ORB. The POA addresses complex issues more fully than the BOA, and its specification is written in IDL.



```
g++ -o hello_simple hello_simple.cc
```

```
./hello_simple
```

As you can see, this simple program just returns a copy of a string using **strdup**, then we need to reclaim the memory that **strdup** used.

Now we need to convert it to CORBA because our Python buddies were so impressed by our coding skills that they want to use the object in their Python programs. Firstly, we need to create an IDL to describe what we would like to send across the 'Net:

```
interface hello
{
    string say_hello();
};
```

Save it as *hello.idl*. Notice how the **char\*** is replaced with **string**. The idl compiler will map this to whatever language you plan to use. **char\*** for C/C++, **string** for Java, etc.

## OmniORB

It will greatly help at this point if we can get an IDL Compiler. I'm going to use *omniORB* because it is freely available, and is probably the best of the lot (Shame on you AT&T!).

You can get *omniORB 4.0* from <http://sourceforge.net/projects/omniORB>. I'm using Red Hat 8.0 and it was just a case of:

```
./configure && make && make install
```

Albeit with deprecated warnings which you can ignore. I add the following to *./configure*:

```
./configure --prefix=/usr
```

To make my base prefix */usr* instead of */usr/local*.

Compilation can take quite a while, just be patient. If you have any problems, however, there is an *omniORB* mailing list and they are more than capable of solving most problems. One of the authors of *omniORB* "Duncan Grisby, can be found on the mailing list and is usually more than happy to help with any problems that you may encounter."

Now that *omniORB* is installed we can work on our small program again. To compile the IDL code we need to run:

```
omniidl -bcxx -Wbh=.h -Wbexample hello.idl
```

**bcxx** tells *omniidl* which backend processor to use, because it can also produce Python code with **bpython**.

**Wbh=.h** tells *omniidl* to create *.h* files and not the default *.hh* files. **Wbexample** will create an implementation file that we can use to fill in the coding details.

This will produce three files, *hello.h*, *helloSK.cc* and *hello\_i.cc*. The file *hello\_i.cc* is where we will put our code; let's have a quick look inside the file.

```
...
hello_i::hello_i(){
    // add extra constructor code here
}
hello_i::~hello_i(){
    // add extra destructor code here
}
// Methods corresponding to IDL attributes and operations
char* hello_i::say_hello(){
    // insert code here and remove the warning
    #warning "Code missing in function <char* hello_i::say_hello()>"
}
```

That's our method with constructor and destructor. Seeing as we don't do anything when we create the object, we can leave the 'structors as they are. We just need to fill in the details for our method, so let's go ahead and do it.

```
Char * hello_i::say_hello(){
```

## Some IDL mappings

### Talking C++

OMG IDL	C++	C++ Out Type
short	CORBA::Short	CORBA::Short_out
long	CORBA::Long	CORBA::Long_out
long long	CORBA::LongLong	CORBA::LongLong_out
unsigned short	CORBA::UShort	CORBA::UShort_out
unsigned long	CORBA::ULong	CORBA::ULong_out
unsigned long long	CORBA::ULongLong	CORBA::ULongLong_out
float	CORBA::Float	CORBA::Float_out
double	CORBA::Double	CORBA::Double_out
long double	CORBA::LongDouble	CORBA::LongDouble_out
char	CORBA::Char	CORBA::Char_out
wchar	CORBA::WChar	CORBA::WChar_out
boolean	CORBA::Boolean	CORBA::Boolean_out
octet	CORBA::Octet	CORBA::Octet_out

```
CORBA::String_var s = CORBA::string_dup("What a cool
Corba filled World!");
return s._retn();
}
```

**CORBA::string\_dup** is part of the CORBA API, and **String\_var s** is a smart pointer; it will manage the memory used by the string we created, so we don't need to worry about freeing memory. We can compile the code with the following:

```
g++ -o hello_hello_i.cc helloSK.cc -l. -lomniORB4
(you may need to specify -L/usr/local/include )
```

You may need to include:

```
#include <stdlib.h>
```

If you have problems with the **exit()** function. On Redhat 8.0, I had to remove the *.h* from *iostream.h* and insert:

```
using namespace std;
```

Now run it with:

```
./hello
```

Inside it's own terminal. You should see the following:

```
IDL object hello IOR = 'IOR:010000000e00000049444c3a6
8656c6c6f3a312e30000000010000000000000640000
00010102000e0000003139322e3136382e322e31333000f
88a0e000000f1a9c73d0000624700000000000000020
0000000000000800000001000000005454410100000
01c0000000100000001000100010000001000105090101
000100000009010100'
```

That long alphanumeric sequence is the glue that holds everything together. It's called an IOR (Interoperable Object Reference). To see what it represents, you can use the following:

```
catior <IOR:01000000e... insert your ior string here>
```

The output is:

```
Type ID: "IDL:hello:1.0"
```

```
Profiles:
```

```
1. IIOP 1.2 10.0.0.2 35576 "....=.bG...."
```

```
TAG_ORB_TYPE omniORB
```

```
TAG_CODE_SETS char native code set: ISO-8859-1
```

```
char conversion code set: UTF-8
```

```
wchar native code set: UTF-16
```

```
wchar conversion code set: UTF-16
```

We need to create a client, to connect to our server and instantiate the infamous **hello** class. Let's create a file called *client.cc*:

```
#include <omniORB4/CORBA.h>
```

```
#include <hello.h>
```

```
#include <iostream>
```

```
#include <stdlib.h>
```



# TutorialCORBA

```

using namespace std;

int main(int argc, char* argv[])
{
    CORBA::ORB_var orb;
    try
    {
        orb=CORBA::ORB_init(argc,argv);
        CORBA::Object_var obj = orb->string_to_object(argv[1]);
        hello_var hello = hello::_narrow(obj);
        cout << hello->say_hello() << endl;
    }
    catch(const CORBA::Exception &)
    {
        cout << "Corba Exception instantiating orb\n" << endl;
    }
}

```

To compile, do the following:

```
g++ -o client client.cc helloSK.cc -l. -lomniORB4
```

Then to run the client you do the following inside another terminal (Make sure your server is running):

```
./client <IOR:01000000e... insert your ior string here>
```

And you should see the familiar string output!

'Not particularly impressive', I hear you say! Well, it does get better. Let's investigate the client code... First we need to create an instance of our ORB (Object Request Broker), with the line:

```
orb = CORBA::ORB_init(argc,argv).
```

This initialises the **orb**, and creates an ORB Object, which we can use. The next line creates an object from the string we passed on the command line; we then narrow the object and get a reference to the hello object from the server.

The last line of the block then uses the method on the server to get the string. Quite straightforward – the first few lines are basically boilermaker templates, we just need to change the methods that we call, give it a go, write some code to return different strings based on the number you send. Here's the interface for it:

```

interface more_strings
{
    string send_code(in short choice);
};

```

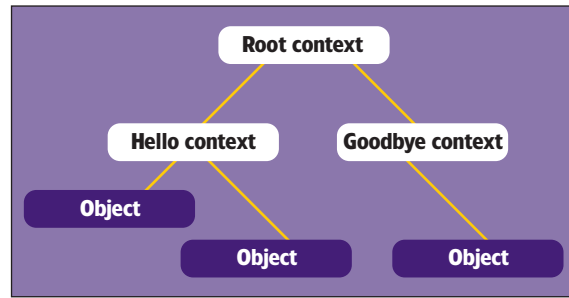
## The NameService

From the last example, you can see that without the IOR, the client is useless. If you run the client without it, you will get an exception. Every time we run the client it will need the IOR, we can give it the IOR on the command line, we can have the client read a file to get the IOR if we wish, even email the client the IOR (LDAP is also a good choice). The most common method of retrieving the IOR, however, is by way of the **NameService**.

The **NameService** is very much like the Domain Name Service (DNS), in that we store IOR's in a tree like structure.

We can start at the top of the tree, then branch off to Web Hosting and have another branch to Corporate Hosting, these two branches can also fork off to mail servers branches and at the end of mail servers will be a node/cell with an object reference. By giving the path to the **NameService**, it can retrieve the object reference that we want.

Now the **NameService** in itself is a CORBA Object, so in order to contact the **NameService** we need an IOR. (Chicken or the egg syndrome or what?). Fortunately, omniORB has a



Simple representation of a NameService.

configuration file that we can use to tell our clients/servers where the **NameService** is, and what port it is listening on. Then they can get a reference to the **NameService** and use its methods.

We need to modify our 'hello server' to contact the **NameService** and then the client will also need to be modified, finally we need to put the *omniORB* config file in place.

Let's get started on the server. It's slightly more complicated than the client because we need to define a path in the **NameService** to store our IOR. The new *hello\_i.cc* file should look like this:-

```

//
// Example code for implementing IDL interfaces in file hello.idl
//
#include <iostream>
#include <hello.h>

//
// Example class implementing IDL interface hello
//
using namespace std;
class hello_i: public POA_hello,
               public PortableServer::RefCountServantBase {
private:
    // Make sure all instances are built on the heap by making the
    // destructor non-public
    ~hello_i();
public:
    // standard constructor
    hello_i();
    virtual ~hello_i();

    // methods corresponding to defined IDL attributes and operations
    char* say_hello();
};

//
// Example implementational code for IDL interface hello
//
hello_i::hello_i(){
    // add extra constructor code here
}
hello_i::~~hello_i(){
    // add extra destructor code here
}

// Methods corresponding to IDL attributes and operations
char* hello_i::say_hello(){
    CORBA::String_var s = CORBA::string_dup("What a
CORBA filled world");
}

```

### Note. Not just trees

You can create other graphs besides trees. Trees are easier to visualise when learning about the **NameService**. Also you can have nodes inside a **NameService** that point to other **NameServices**. This is known as a 'Federated NameService'



```

    return s._retn();
}

// End of example implementational code

int main(int argc, char** argv)
{
    try {
        // Initialise the ORB.
        CORBA::ORB_var orb = CORBA::ORB_init(argc, argv,
        "omniORB4");

        // Obtain a reference to the root POA.
        CORBA::Object_var obj =
        orb->resolve_initial_references("RootPOA");
        PortableServer::POA_var poa =
        PortableServer::POA::_narrow(obj);

        // We allocate the objects on the heap. Since these are
        // reference counted objects, they will be deleted by the POA
        // when they are no longer needed.
        hello_i* myhello_i = new hello_i();

        // Activate the objects. This tells the POA that the objects
        // are ready to accept requests.
        PortableServer::ObjectId_var myhello_iid =
        poa->activate_object(myhello_i);
        //*****
        // {
        // IDL interface: hello
        CORBA::Object_var ref = myhello_i->_this();
        /* CORBA::String_var sior(orb->object_to_string(ref));
        cout << "IDL object hello IOR = " << (char*)sior << " " <<
        endl;
        } */

        CORBA::Object_var initServ =
        orb->resolve_initial_references("NameService");
        CosNaming::NamingContext_var rootContext =
        CosNaming::NamingContext::_narrow(initServ);
        CosNaming::Name_var contextName;
        contextName.length(1);
        contextName[0].id = (const char*) "hello";
        contextName[0].kind = (const char*) "context";
        CosNaming::NamingContext_var testContext;
        try
        {
            testContext =
            rootContext->bind_new_context(contextName);
        }
        //catch(CosNaming::NamingContext::AlreadyBound& ex)
        catch(...)
        {
            //context is already there so make testContext point to it
            CORBA::Object_var nobj;
            cout << "Rebinding again" << endl;
            nobj = rootContext->resolve(contextName);
            testContext = CosNaming::NamingContext::_narrow(nobj);
            if( CORBA::is_nil(testContext) )

```

```

        {
            cerr << "Failed to narrow naming context." << endl;
            exit(0);
        }
    }
    CosNaming::Name_var objectName;
    objectName.length(1);
    objectName[0].id = (const char*) "hello_ior";
    objectName[0].kind = (const char*) "Object";
    try
    {
        testContext->bind(objectName, ref);
    }
    //catch(CosNaming::NamingContext::AlreadyBound& ex)
    catch(...)
    {
        cout << "Rebinding" << endl;
        testContext->rebind(objectName, ref);
    }
    // Obtain a reference to each object and output the stringified
    // IOR to stdout
    //*****

    // Obtain a POAManager, and tell the POA to start accepting
    // requests on its objects.
    PortableServer::POAManager_var pman =
    poa->the_POAManager();
    pman->activate();

    orb->run();
    orb->destroy();
}
catch(CORBA::SystemException&) {
    cerr << "Caught CORBA::SystemException" << endl;
}
catch(CORBA::Exception&) {
    cerr << "Caught CORBA::Exception" << endl;
}
catch(omniORB::fatalException& fe) {
    cerr << "Caught omniORB::fatalException:" << endl;
    cerr << " file: " << fe.file() << endl;
    cerr << " line: " << fe.line() << endl;
    cerr << " msg: " << fe.errmsg() << endl;
}
catch(...) {
    cerr << "Caught unknown exception" << endl;
}

return 0;
}

```

Once again, mostly boilerplate code. We get a reference to the NameService with:

```
orb->resolve_initial_references("NameService")
```

Then we narrow it. **CosNaming::Name** is a sequence data type similar to a C++ Vector. This is where you specify the path you want to create in the **NameService**, in this case the path will be hello.context/hello\_ior.Object.

A naming context consists of two parts, id and kind. In graph terms, the id is the arc or edge, and the kind is a description of

## Narrowing an object

Narrowing an object means taking a base type object reference, and converting it to a desired type.



# TutorialCORBA

« what is located at that edge. Therefore id “hello” is a context, and **hello\_ior** is an Object. You can leave out the kind if you want and just create the edges. Just think trees or DNS when you do.

Now the first time you run your server, it will create the path using the **bind** method, but if you run it again, **bind** will cause an exception. Therefore we catch the exception, and just follow the path until we can rebind our reference. According to the CORBA specification, we can just use **rebind**, but some **NameServices** won't allow you to do this, so rather be safe than sorry and try the **bind** first.

The client is a tad easier because we just need to set up the path, and get the information from the **NameService**, narrow the object and use it. I've got one in the oven that I prepared earlier:

```
#include <omniORB4/CORBA.h>
#include <hello.h>
#include <iostream>
#include <stdlib.h>

using namespace std;

int main(int argc, char* argv[])
{
    CORBA::ORB_var orb;
    orb=CORBA::ORB_init(argc,argv);
    CosNaming::NamingContext_var rootContext;
    try
    {
        CORBA::Object_var obj =
        orb->resolve_initial_references("NameService");
        rootContext =
        CosNaming::NamingContext::_narrow(obj);
        if(CORBA::is_nil(rootContext))
        {
            cerr << "Failed to narrow the root naming
            context" << endl;
            exit(0);
        }
        CosNaming::Name name;
        name.length(2);
        name[0].id = (const char*) "hello";
        name[0].kind = (const char*) "context";
        name[1].id = (const char*) "hello_ior";
        name[1].kind = (const char*) "Object";
        try
        {
            CORBA::Object_var nobj =
            rootContext->resolve(name);
            hello_var hello = hello::_narrow(nobj);
            cout << hello->say_hello() << endl;
        }
        catch(CosNaming::NamingContext::NotFound& ex)
        {
            cerr << "Context not found in NameService"
            << endl;
        }
        catch(const CORBA::Exception &)
        {
            cout << "Corba Exception instantiating orbln" << endl;
        }
    }
```

```
}
```

To compile, just do as we did before for the client.

As per the server, we get the initial reference to the **NameService**, then we create the path we want, using a **CosNaming::Name** structure, resolve the name, cast it to our object and use it. It's as simple as that! For your own objects, you would just change the path information.

The **NameService** can be anywhere in the world, and with *omniORB*, you just need to use a configuration file to tell your client/server/servant where it is.

The default file goes in /etc and is called omniORB.cfg. So go ahead and create one and put the following line in it.

```
InitRef = NameService=corbaname::localhost
```

To run the NameService, do the following:

```
omniNames -start &
```

You may need to create a directory in /var, called omninames i.e. /var/omninames. Now run your server in a separate terminal, and it should create an entry in the **NameService**.

To check that it has you can run the following application:

```
nameclt list hello.context
```

That should return **hello\_ior.Object**. You can then run

```
nameclt resolve hello.context/hello_ior.Object
```

And that should return an IOR, which you can then check with **catior**. You can change localhost to your hostname, but make sure that /etc/hosts has correct entries in.

Therefore, if you machine has an IP of 10.0.0.2 and your hostname is corba.cool.com, then Red Hat may put the following into your hosts file:

```
127.0.0.1 corba.cool.com localhost.localdomain localhost
```

Rather change this to:

```
127.0.0.1 localhost.localdomain localhost
```

```
10.0.0.2 corba.cool.com corba
```

Now you can run your client in another terminal with:

```
./client
```

And it should get itself an IOR from the nameservice and return our illustrious message. Just make sure that the hello server is running.

## Help

If you have any problems when running your server or client, then you can append **-ORBtraceLevel 25** onto the end of the command line, and this will display debugging information, the lower the number, the less information you will get. You can send this information to the mailing list, along with a description of the problem you are having, and someone should be able to help you.

More information on *omniORB* can be found in the docs directory of the source code. *OmniORB.pdf*, is a good place to start, and there are also examples you can try out.

Of all the CORBA books out there, the one that stands out in my opinion, is *Pure CORBA*, by Fintan Boltan (SAMS publishing, ISBN 0-672-31812-1). It provides both C++ and Java examples, and covers beginners and advanced topics. [LXF](#)

## Some CORBA links

[www.omg.com](http://www.omg.com)  
<http://omniorb.sourceforge.net>  
[www.pure-corba.com](http://www.pure-corba.com)  
[www.omg.org/gettingstarted/omg\\_idl.htm](http://www.omg.org/gettingstarted/omg_idl.htm) (Good for idl details)  
[www.infosys.tuwien.ac.at/Research/Corba/OMG/corb2prf.htm](http://www.infosys.tuwien.ac.at/Research/Corba/OMG/corb2prf.htm)



ROLL YOUR OWN JAVA BEANS

# Speaking Java

**Richard Drummond** continues our tour of Java's component model and does some DIY bean-building.



**W**e started to talk about JavaBeans last time, with a view to turning the multi-column list widget we've been developing into a bean which can be plugged into a visual IDE, such as NetBeans, and used in the point-and-click construction of GUIs.

The latest version of the *MultiList* widget can be found on the coverdisc, and, while this isn't complete yet, it is functional enough to be used in applications. In fact, it's used in the *Index* applet in the HTML front-end of this month's disc. Don't take the code too much to heart, however. It's still pretty messy, and was built for speed, rather than with an eye to good design. There's still plenty of cleaning up, modularisation and features to be implemented. But it's a starting point for our experimentation with beans.

If you remember from the last issue, a Java class must meet certain conventions if it is to be used as a bean. Not surprisingly, our *MultiList* widget does. Thus it has a zero-argument constructor, and all of its property accessors follow the **getX()** and **setX()** naming conventions that beans require. We could actually use this class as a bean as it stands, but it can be made more user-friendly by implementing a **BeanInfo** class.

## I want information

We said last time that the classes which make up the run-time behaviour of a bean can be augmented with auxiliary classes that provide extra design-time features and information. One of these auxiliary classes is the **BeanInfo** class. The **java.beans.BeanInfo** interface declares methods through which a bean builder tool can obtain information about the properties a bean has, the methods it provides, the events it can generate and so on. In most cases, however, a **BeanInfo** class isn't strictly necessary. Unless your bean has bound or vetoable properties, the JavaBeans system can work out all the information it needs via introspection. However, implementing a **BeanInfo** class for your beans can make them easier to use.

To create a **BeanInfo** class for your bean, you simply need to implement the **BeanInfo** interface, providing method implementations which return information relevant to your bean. The convention is that a bean's **BeanInfo** class is named after the bean's main class with the suffix 'BeanInfo' tagged on. Thus our *MultiList* widget should have a **BeanInfo** class called **MultiListBeanInfo**.

In most cases, you wouldn't actually implement **BeanInfo** directly, but instead sub-class the **java.beans.SimpleBeanInfo** class. This is an adapter class, if you like, and provides empty implementations of the methods that **BeanInfo** declares. It's generally quicker just to use this and override the methods that are appropriate to your bean.

Here's an implementation of our **MultiListBeanInfo** class:

```
package net.rcdrummond.awt;

import java.beans.*;
import java.lang.reflect.*;

public class MultiListBeanInfo extends SimpleBeanInfo
{
    public static Class BEAN_CLASS = MultiList.class;

    static PropertyDescriptor[] properties = {
        createPropDesc( "listData", "The object that provides the
data displayed in this list" ),
        createPropDesc( "background", "The background colour
of the list" ),
        createPropDesc( "foreground", "The foreground colour of
the list" ),
        createPropDesc( "listFont", "The font used to render the
list body" ),
        createPropDesc( "titleFont", "The font used to render the
column titles" ),
        createPropDesc( "cursorRow", "The list row that the
cursor is currently on" ),
        createPropDesc( "topRow", "The first visible row shown in
the list" )
    };

    public java.beans.BeanDescriptor getBeanDescriptor() {
        return new BeanDescriptor( BEAN_CLASS, null );
    }

    public int getDefaultPropertyIndex() {
        return 0;
    }

    public java.beans.PropertyDescriptor[]
getPropertyDescriptors() {
        return properties;
    }

    static PropertyDescriptor createPropDesc( String name,
String description ) {
        try {
            PropertyDescriptor p = new PropertyDescriptor(
name, BEAN_CLASS );
            p.setShortDescription( description );
            return p;
        }
        catch( IntrospectionException e ) { return null; }
    }
}
```

}

This class is quite straight forward. The **getBeanDescriptor()** method returns a **BeanDescriptor** object, which encapsulates general information about the bean that this class represents, such as the bean's class and whether it has a **Customizer**. The **getPropertyDescriptor()** method returns an array of **PropertyDescriptor** objects, each of which provides information about one of the bean's properties, giving the property's name, a helpful description, whether it has a custom property editor, and so on. Here, we've implemented a helper method, **createPropDesc()**, to simplify the creation of the **PropertyDescriptor** objects. Additional **BeanInfo** methods describe the methods provided by and the events fired by the bean, but we haven't implemented these here.

How the information provided by a **BeanInfo** class is used by a bean builder tool depends on the tool you are using. NetBeans, for example, displays the properties described here in its GUI property editor rather than the property's that would be found by introspection. Thus you are presented with the properties that are important for setting up that bean, rather than being over-burdened with all of the bean's properties, including those of all its parent classes. This is part of the beauty of the **BeanInfo** system. It lets the bean designer document what properties are important in a way that is easily accessible at design-time.

## A jarring experience

The next step is to pack our bean into a jar file, so that it can be easily deployed. This includes the classes which implement the run-time behaviour of the bean itself, and any auxiliary design-time classes such as **BeanInfo** classes, custom property editors, and customisers. You can store several beans in a single jar file, a useful tip for saving space when beans share code. The naming of the jar file itself isn't significant. Just choose something memorable.

Each bean stored in the jar must have an entry in the jar's manifest. A jar's manifest, if you recall, contains various meta information about the classes stored in the jar, and a default manifest is generated automatically by the **jar** tool. We must add extra information to this default manifest for each bean stored in the jar, so that bean builder tools can locate them. Firstly, the file path, relative to the root of the jar, must be specified for each bean's class. Secondly, each bean must be flagged as being a bean with the attribute **Java-Bean: true**.

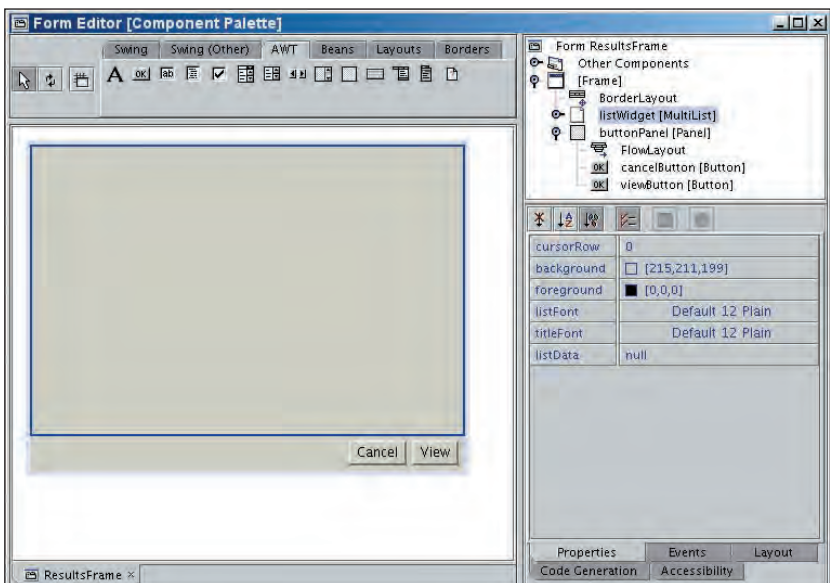
The process of creating a jar is straightforward and, if you haven't done it before, it is much like creating a tarball. The additional first step is to prepare a skeleton manifest file containing any extra information we require. Here's an example for our *MultiList* bean.

**Name:** net.rcdrummond.awt.MultiList.class

**Java-Bean:** true

The blank line before the **Name** attribute here is significant. Although it's not strictly required, NetBeans, for instance, won't recognise a bean stored in a jar file unless its manifest entry is preceded by a blank line (and this includes the first bean).

If we now save this skeleton manifest as a file called *manifest*, we can then use it to pack the jar with a command such as:



```
jar cvfm ../multilist.jar ./manifest \
net/rcdrummond/awt/MultiList.class \
net/rcdrummond/awt/MultiListBeanInfo.class \
net/rcdrummond/awt/MultiListBody.class \
net/rcdrummond/awt/MultiListHead.class \
net/rcdrummond/awt/MultiListModel.class \
net/rcdrummond/awt/MultiListAdjustmentEvent.class \
net/rcdrummond/awt/MultiListAdjustmentListener.class
```

Of course, we'd typically include this procedure in a *make* script or *Ant* build file, rather than doing it manually.

It's worth pointing out that you must remember to include all the class files that your bean depends on in the jar file. Pay particular attention to any anonymous inner classes, because these are easy to forget. Thus, in our example, the compiled anonymous inner classes **MultiList\$1.class**, **MultiList\$2.class**, etc., will be brought in by the wildcards specified above.

## Deploying your bean

Now it's time to put our bean to use. To use a bean at run-time, all you need do is to ensure that the bean's jar file is added to your classpath when executing your code. The job of installing a bean for use by a particular bean builder tool, however, will depend on the tool that you are using.

With Sun's BeanBox, for example, you can either copy your bean's jar file into BeanBox's jars directory, or you can add it at run-time with the 'Load jar' option from the 'File menu'. An entry will appear in the 'Toolbox' for the new bean, and you can design away as normal.

Similarly, NetBeans can be made to use a new bean with the 'Install New JavaBean' option from the 'Tools' menu. Locate the jar with the file dialog, and then NetBeans will pop up a list displaying all the beans available in that jar. For each bean that you select, it will ask you on which tab-pane in the component palette you would like it installed.

The current release of NetBeans has a minor problem with the *MultiList* widget. It doesn't like you adding non-empty containers to the component palette, which our widget is since it is derived from the AWT's **Panel** class. The first time you use it, NetBeans will complain with an exception. Just cancel this and carry on as normal. [LXF](#)

**Our MultiList bean in all its glory, deployed and being designed with in NetBeans.**

## NEXT MONTH

We'll continue our discussion of JavaBeans next time, and take a closer look at the how beans can generate and listen for events. In particular, we'll see how beans can use the **PropertyChange** mechanism to get notified when another bean's state is modified.





# OpenOffice.org for power users

**In this month's OpenOffice.org Masterclass, Neil Lucock tries to give an illustrated talk on "What I did on my holidays" while upside down at 5,000 feet. If the typing appears upside down, please invert the magazine.**

**O**penOffice.org's presentation software, *Impress*, was the first thing I ever used in the suite. It's competing with Microsoft's *PowerPoint*, the industry standard, so I'll be considering how well it compares and whether it offers a viable alternative. As always, I'll presume that you have a working version of *OpenOffice.org* (OOo) on your PC, know how to open and save files and make directories. The review was written in OOo *Writer* running in KDE 2.2.2 in Mandrake 8.2.

## Getting started

OOo is a genuinely integrated office suite. Impress benefits as you can use the tools provided in Draw. I'll cover using those tools in detail in the next tutorial, but I'll need to discuss a few in order to make a presentation. Launch OOo and choose File>New>Presentation to start *Impress*. The autopilot (wizard) starts by offering to either open an existing presentation, use a template (although OOo 1.0 doesn't include any, you can save your own), or create a new one. The next dialogue asks for your slide design. Again there are no options offered. OOo is the Open Source version of *StarOffice* 6 (SO), the templates and

designs are in the commercial product. You can define how the finished document will be used; as slides, overhead projector transparencies, paper or as an on-screen presentation. Next define the transition from one slide to the following one (if you are creating an on-screen presentation). You can also set the display times if you want your presentation to autorun. Set it to manual (the default) for the time being, you can alter it later. The final step in the wizard is to define how the slide looks. They give you a useful selection of 20 layouts to use. I prefer to start with a blank slide. Click the Create button and the *Impress* interface appears.

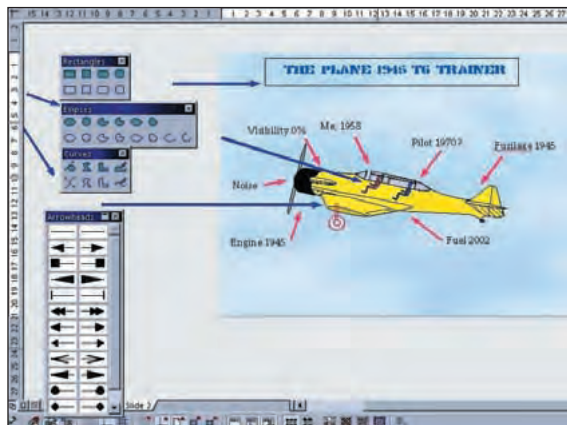
I'll make a short presentation and we'll have a look at the essential tools in *Impress*. I recently flew a small plane and did some aerobatics, I'll use it as the subject of my presentation. I think I'll need six slides.

The first slide is going to be the title. I've chosen a military stencil font for the titles as the aircraft was a US military trainer. There's a "T" icon on the left tool bar near the top, it lets you put a text box in your presentation. Most of OOo's icons in *Impress* have a sub-menu that you can bring out by clicking and holding the left mouse button down. The small green arrowhead tells you that it's there. Once the sub menu is displayed, you can make it stay open by clicking on its title bar. You can then reuse a tool without having to find it in a menu. I draw a lot in my presentations and I find this feature makes *Impress* easy to use. There's a couple of other useful tools on the text tool bar. The "T in a box" scales your text to the size of the box you drag. If you

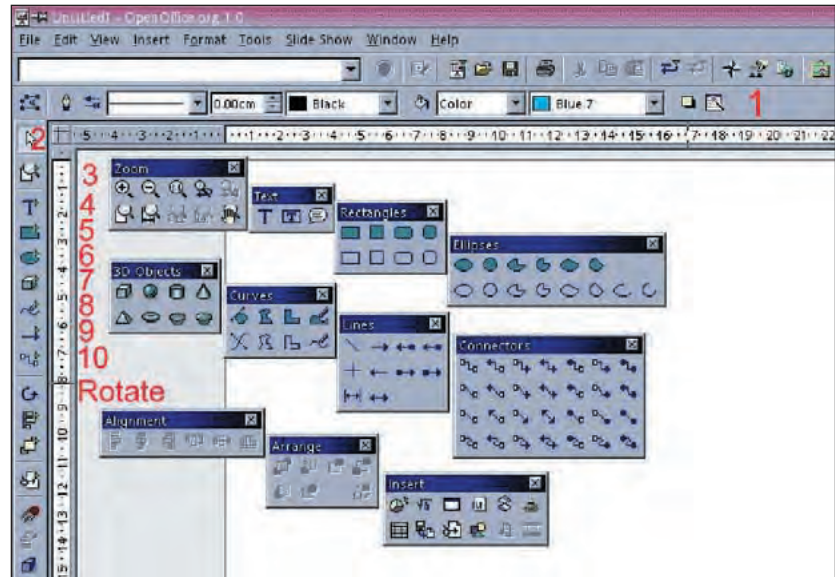
change the size of the box, the text inside it automatically re-scales itself to fit the new box size. The "Callout" (which looks like a speech balloon) is for labelling things. Click the item you want to label, drag the line out to where you want the details to appear and it gives you an empty box. Click the T on the text sub-menu and you get an insert point. Type in the details you want to display.

## Picture this

I've put a picture of the plane in (Insert>Graphics) and resized it. The clouds background is from The Gallery. OOo includes the backgrounds, fancy bullet points and 3D objects that are in SO. You launch it with the icon that looks like a framed picture to the extreme right of the top line of icons. Just find a picture you like and drag and drop it. A hint: put the background in last, otherwise it becomes difficult to select text and any other pictures. When you drop your background onto your slide, it covers up everything. About a third of the way up from the bottom of the left hand tool bar is a useful series of icons for positioning. We need to send the background image to the rear, so it is behind everything else. Each item (either a picture, drawing or text) on the presentation slide can be compared to a picture on a sheet of transparent plastic. Newly made items are nearer to you (on their own sheet of plastic) and will cover items made previously. We have the option to send things backwards or bring them forwards.



Pilots are a series of ellipses. Note that OOo's spell checker is active, underlining fuselage. The plane was traced from the photo. If you can recognise it as a plane, it is good enough.



The left side sub-menus. 1, This area changes according to what is selected. 2, The select arrow. 3, Zoom tools. It includes a tool to make the display the true size, e.g. When printed. 4, Text tools. 5, Filled and unfilled rectangles. 6, Ellipses. The tool on the far right of the top line was used to make the clouds. 7, 3D objects, rotate them, alter the lighting, to be covered in the next article. 8, Bezier curves and shapes. Really part of Draw. 9, Lines. 10, Connectors. 11, The rotate tool. Select something, then click on this to interactively turn it. Under Rotate we have Alignment, useful if you want to align objects relative to the top, side, bottom etc. of the slide. Arrange brings an object in front of another or puts it behind it. Insert puts graphics, spreadsheets, photographs and formulae into your presentation.

Click on an item to select it, then click on the icon in the sub-menu to send it to the back. There's also an easier way. Right click after selecting the object and one of the options is Arrange. Choose what you want to do from the options offered.

I want the presentation to be light-hearted rather than serious, so I've added an arrow in red and the word "AAAAAGH!"

For the second slide, I drew the plane using the drawing tools, then put a note to everything I want to talk about. I know that when I come to the one that says "Engine 1945" I'll talk about the noise, the vibration and the fact that you can't see in front of the plane when it's on the ground. "Fuel 2002" reminds me of the quote from the company's website that the plane is a very efficient method of turning gasoline into noise. These notes mean nothing to anyone else but will prompt me to talk about each item. It's like having a secret list that no one else can read. You >>

## Different Views

For good preparation

Impress shows you the Drawing View as a default. (View>Master View>Drawing View) This allows you to edit the slides and make new ones. The other options are:

**OUTLINE VIEW:** This allows you an overview of your show. Clicking the small icon to the left of the preview shows you what is on each slide. You can rearrange them by drag and drop.

**SLIDES VIEW:** If you did not set a general rule for slide transitions when

you started Impress, you can define how each slide appears, the speed it arrives at and, (if you have the display set to automatic) how long each slide displays for before moving to the next image. There's a timer you can (in theory) use to practice getting the length right and an icon to hide slides. Perhaps you have prepared a presentation but realise that the person who spoke before you has already covered some points. Just temporarily hide the slides that you don't need in your presentation. Right-click for a menu to include it again.

**NOTES VIEW:** Notes have three functions. Firstly, they remind you exactly what it was that you were going to say before nerves, distractions or just plain ignorance left you standing in front of a crowd of people without a clue as to what you are supposed to say next. They are a backup for the occasional mental segfault. Secondly, they can be printed and distributed to the audience after your presentation. It saves them taking notes. Thirdly, they are a useful backup if your equipment breaks or is otherwise unavailable. If you carry your notes, you

can still do your presentation. Photocopy your notes, hand them to the audience and refer to the pictures on each sheet. Good presenters always have a "Plan B" in case something goes wrong.

Finally, there's a **HANDOUTS VIEW:** OOo makes handouts with your pictures on. If you want to add your details, a few arrows or lines, you have all the usual OOo tools available. I put a simple text box at the top, then made a new hatching type and put that over one of the slides that I don't plan to show.

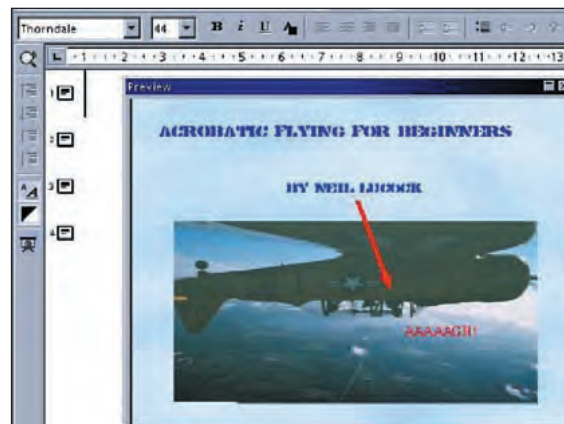
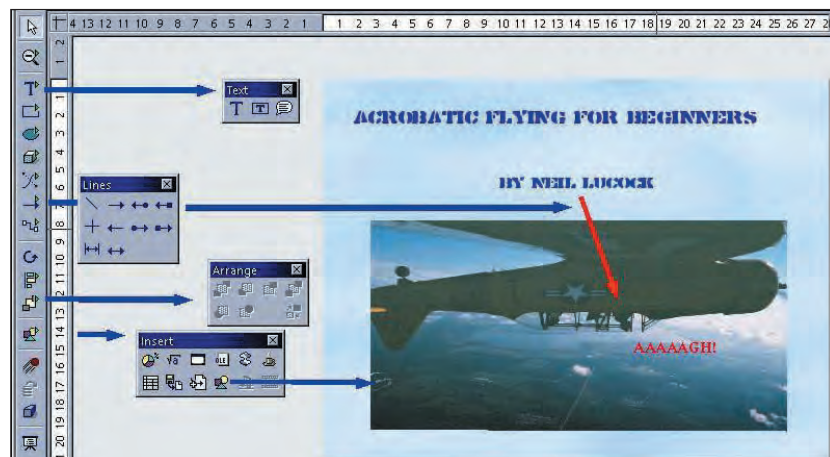


cover all the items in your talk because they are in front of your eyes. The plane was drawn by importing a picture, drawing around it, then deleting the picture to leave the outline. Tidy it up and add a few details. It does not have to be a work of fine art, presentation drawings need to be simple and clear. As long as people recognise what you have drawn it is good enough. Use **Insert>Slide** to make a new one after the one you are editing. **Edit>Delete Slide** gets rid of it.

I made some simple clouds for the slide with the flight path on it. There's an **Ellipse** toolbar on the left of the work area and an **Ellipse Segment** tool in that. It produces a circle with a flat section. Make one, make the line around it invisible (the tools above the work area change depending on what is selected) and fill it with a pale yellow. Rotate the object using the **Rotate** tool on the left. Now copy it several times and make the copies different sizes to make a cloud shape. Drag a selection box around them all. Either use **Ctrl+Shift+G** or **Format>Group** to join them together. Pale yellow doesn't stand out too well, make a copy of your cloud, colour it pale grey and place it behind the pale yellow one.

Sometimes you need to include large images in your presentation, and they slow down your machine as you are working on it. Perhaps your presentation will be printed out on monochrome printers as a handout and you want to make sure

**You can make an acceptable presentation using just a few of Impress's tools. I've shown exactly what I used for this slide.**



**The Outline View. Drag and drop your slides into a different order.**

that the details are still there without colour. *Impress* can display everything in colour (the default) but you can have it display as a greyscale or in black and white. **View> Display Quality** allows you to change how it appears. Note that this change is not permanent, you can put the colour back in again.

## Making Slide Shows

On the menu bar there's a **Slide Show** heading. *OOo* has the usual keyboard shortcuts (**Ctrl-C**, **Ctrl-V** etc.). It says that **Ctrl-F2** runs the slide show. No so, it just drops the program onto the KDE taskbar. You can configure your own shortcuts for *OOo* tools. Choose **Tools>Configure>Keyboard**. Of course, **Alt-S**, then **Alt-W** will work, using the underlined letters in the menus.

**Slide Show** runs the presentation. Clicking the left mouse button or pressing the space bar takes you to the next slide (unless you have set it to run automatically). Right clicking during the presentation was not consistent. It would sometimes take me back a slide, at other times it briefly flashed the previous slide on the screen. I prefer *PowerPoint's* method, where a right click gives you a menu. Someone always asks a question that begins "On the previous slide you showed....".

**Rehearse Timings** gives you a small digital clock on the bottom left hand side of the screen. This is a useful feature that is not really well thought through. It launches your slide show,

## Doing a Presentation?

Get the basics right

So, you have to stand in front of some people and talk about something? I'm not going to discuss presentations in detail, but I need to address some issues in order to use *Impress* effectively.

The first thing to do with any talk, speech or tutorial is plan it. That involves finding a pencil and some paper and writing down what subjects you need to cover in the order you think best. Most presentations benefit from a heading showing the subject, who is speaking and what topics are to be covered.

It's a good idea to do a summary too.

A lot of people seem to forget that presentation software is just a tool. They think that it will do the work for them if they just use the supplied wizards to make their presentation. They then wonder why their audience is acting bored while they are reading it at them. Treat your software as you would a blackboard. If you need to show something, it can display an image for you. You can then talk about the image you have displayed, explaining it.

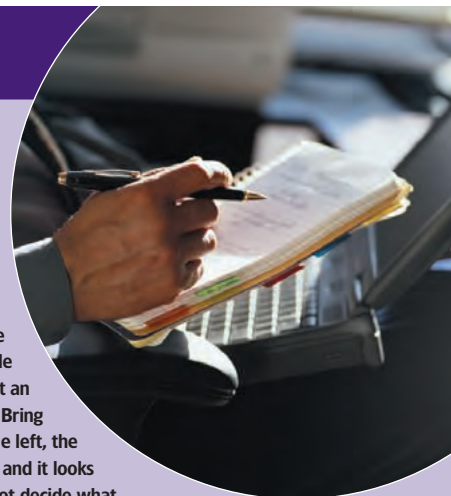
Yes, they can take it in quickly, but they are not going to be getting ahead of you. You remain the focus, it remains the tool.

A final point: presentation software often comes with flashy slide transitions. You can make one slide dissolve, slide in from one side or another or include "amusing" noises.

These detract from what you are talking about. Is the intention to show what the software can do or to inform the audience? If you really must use them, consider how

a written document would look with lots of different fonts.

Make all your slide transitions come from the left side and it will give it an integrated feel. Bring some in from the left, the top or the right and it looks like you could not decide what to do. That's not the impression you want to make.



you practice saying what you need to say and figure out how it will all fit into the time you have available. Unfortunately, it doesn't give you a summary. What you need is something that says "Total time 8 minutes" or similar. Both *PowerPoint* and *Hancom Office's* presentation software do it that way. *OOo's* version displays the timer so small that you'll have to be sitting right next to your screen. You'll need to be next to a flat surface anyway as you'll need a pen and paper to write down the time each slide needed. Then you'll have to add it all up to find the time it took. It would be much easier to find a clock and note the start and end times when you practise. The tool needs more work before it is any use.

If you've ever done a loop in a plane, the worst part is not being vertical or inverted, it's the bottom of the power dive you need to do to build up speed before you pull up. All the blood flows to your stomach and legs and the G-forces make it uncomfortable to breathe. I've drawn a loop and labelled parts of it. I intend to ask the displayed question, get responses from the audience, then display the answer. I used Insert> Duplicate Slide to make an exact copy of the previous slide. I then deleted everything except the answer. Click to highlight the text box and the hatched area. Click Slide Show> Effects will allow you to assign an effect to your text and objects. I made the text appear on a mouse click, shooting in from the left one letter at a time.

Slide Show> Interaction is a nice tool. It allows you to set special behaviours when you click an object on the page. You can set it to go to a different slide, make a noise or run a program. It's almost like having hypertext links in your presentation. If you wanted, you could use a presentation as an interactive teaching aid.

## Conclusions

Most of the tools in *PowerPoint* are present, but there are some things are missing. For example, you don't get the ability to convert files to Adobe PDF, probably for copyright reasons. Impress is a good piece of software in its own right. It has a good set of tools and while it is not a clone of the Microsoft program, you should have no trouble making the transition. There are a few things that could be better (the timer, as an example) but there's nothing left out that you would need.



**Notes View.** If your mind goes blank when you see who's in your audience, have it written down.

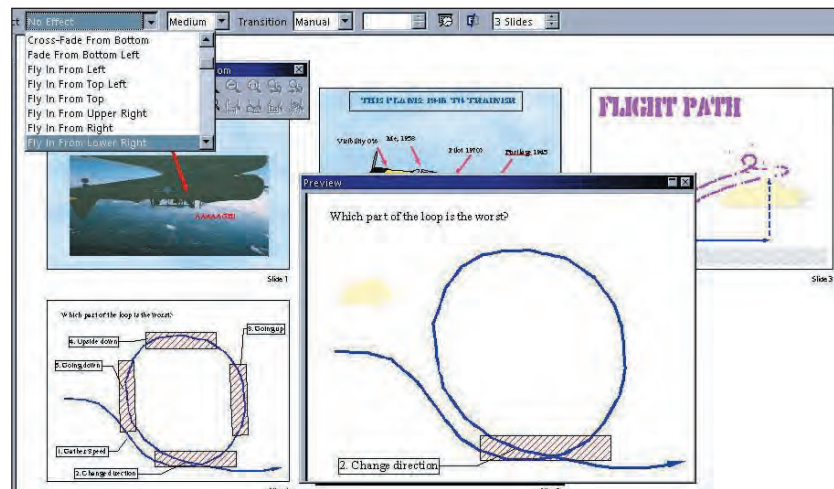
Perhaps the best argument is the cost. Linux is a free download or a cheap CD. *OOo* is the same. If your business has just ten registered copies of *MS Office*, running on ten Windows PCs, you are probably looking at £3000 to £4000 in software that will be obsolete in a couple of years. You'll be asked to upgrade, and it will cost. *OOo* will still be free. Buy one CD (not more than £5 from the place you get your

## PowerPoint Import and Comparisons

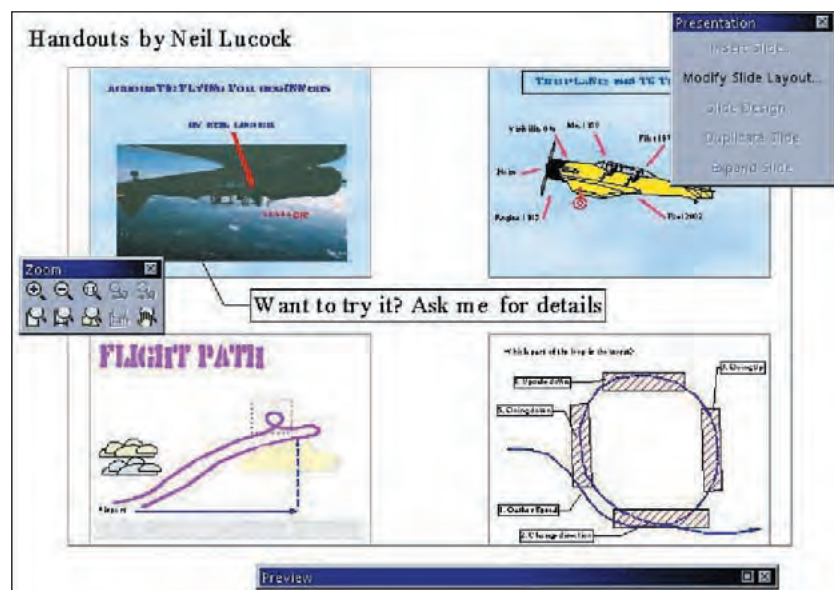
### Living with the opposition

When *LXF* did their reviews of Linux office suites *OOo* and *SO* showed that there was a viable alternative to *MS Office*. Impress will usually import *PowerPoint* materials created with anything up to *Office 2000*. Unless the particular font isn't on your machine, you will not tell the difference. It exports reliably too, I've drawn maps in *Impress* and used them with *PowerPoint* at work.

However, as with any Open Source program, don't expect it to be 100%, particularly when it comes to importing files from new software releases. I was sent some *PowerPoint* Safety Training presentations. I'm not sure what version of *Office* they were made with, but while *Office 2000* will open them, *OOo* failed. It takes a while for someone to write import filters for the latest version.



**Slides View.** If you did not set the way your presentation displays when the program started, you can set various effects and hide slides from this display.



**The handouts display** allows you to insert notes and captions.

Linux CDs from) and you can install it on all your PCs and you don't have to worry about record keeping or the the BSA knocking on your door. How much does the licence record keeping, (to prove that you have bought licences for all the copies of your software) cost you? How much would a prosecution harm your business? *OOo* and Linux have benefits that go beyond the cost of the software itself, and I didn't even mention Linux reliability once. **LXF**



## IMAGE GENERATION

# Practical PHP Programming

This month, **Paul Hudson** begins a three-part mini-series on multimedia creation, starting with image generation

**A**fter databases, the PHP topic I am most questioned on is media generation – how to create pictures, Flash movies, and the like just by using code. PHP has a lot to offer programmers wishing to dynamically generate multimedia content, and it really is a big topic to cover. In order to feature as much as possible in good depth, I'll be covering media generation in three separate parts, of which this the first – part two will focus on Flash movies, with part three covering PDF generation.

For now, though, the topic is image generation, by which I mean the creation of static bitmap content like JPEG and PNG pictures. If you're looking for information on using PHP to create GIF images, I'm sorry to say you're looking in the wrong place! Read the box, *Why are GIFs so bad?*, for more information. One warning: be prepared for lots of code, as media generation isn't something that's done with magic one-liners!

## LXF Tip

**Warning:** You must not forget to call **imagedestroy()** otherwise your web server will slowly chew up more and more system resources until your server all but locks up. Always keep your scripts in order – don't forget to clean up after yourself!

## Working with image data

If you've previously only used PHP to work with HTML content, it'll come as a nice surprise that working with image data uses almost exactly the same process. It seems many people just don't realise PHP is capable of handling much more than non-HTML data, this article should help shed light on this exciting and challenging topic. In order to work with images, you must first understand that each type of image, whether it be JPEG, PNG, or WBMP, all have their own unique file formats that need to be absolutely precise – you must be careful not to use any text in your image PHP scripts, not even extra blank lines before or after your `<?php` and `?>` tags.

To start the journey into the world of media, we'll begin by drawing some simple shapes. If this seems a little easy for you, relax – this is only the beginning! An important PHP function here is **header()**. **header()** outputs a HTTP header of your choice, and in this situation we'll be sending the content-type header, which tells web browsers what kind of content they can expect through the connection. Popular content types include text/plain for plain text documents, text/html for most web pages, and image/\*, where the \* is PNG, JPEG, GIF, or other picture formats.

As **header()** sends HTTP headers, it must be used before you send any content through. This is a core HTTP rule – no headers can be sent after content. This is the same thing that stops you from using cookies after you've sent content.

Creating a new image to work with is as simple as calling the **imagecreate()** function, which has just two parameters: the height and width of the image you wish to create. **imagecreate()** will return **false** if it failed to create an image, otherwise it will return the image as a resource for you to use in other image functions.

**imagecreate()** is of course complemented by **imagedestroy()**, which takes an image resource as its only parameter, and frees up the memory assigned to the image. Images are special in PHP as one of the few resources not automatically cleaned up when your script ends, so it is absolutely imperative that you call **imagedestroy()** on all your image resources once you are finished with them.

PHP provides a fantastic array of functions for you to use to manipulate an image and, when you're done, choose your output format and the pic is finished. To output the picture, you call one of several functions. If you want to convert it to PNG format, you call **imagepng()**, which takes two parameters: the image resource to use, and a filename to save the picture as, which is optional. If you don't provide the second parameter, **imagepng()** sends the PNG-format picture straight to output, which is usually a visitor to your site.

To choose JPEG, you call the **imagejpeg()** function, which takes three parameters – the same two as **imagedestroy()** plus the quality you wish to use for the picture. The quality, a number between 0 (lowest quality, smallest file) and 100 (highest quality, largest file) is optional, as is the filename parameter. If you want to set the quality without specifying a filename, just provide an empty string (") as the filename.

## First steps

Let's go from theory to practice and get stuck in with image creation!

```
<?php
$image = imagecreate(400,300);
// do stuff to the image
imagejpeg($image, "", 75);
imagedestroy($image);
?>
```

Go ahead and save that in your public HTML directory as *picture1.php*. Most of your pictures will probably be referenced from a webpage,

## Choosing a format

### More alphabet soup

PHP gives you a good selection of options when choosing an output format: PNG, GIF, JPEG, and WBMP to name a few. Normally, though, one is best for a certain type of image. Generally speaking, for high-quality images with many colours or a lot of detail, the JPEG format should be preferred. JPEG saves in true colour and allows you to set the compression ratio in order to get the best trade-off between

size and quality. PNGs, on the other hand, work best as a replacement for GIFs, and as such work well using limited colours. They also offer alpha transparency, and really teensy file sizes.

So, put as simply as possible: for photographs, prefer JPEGs, and for everything else, prefer PNGs. Just as an aside, and at the risk of starting a flame war, the correct pronunciations are: "ping", "jay-peg", and "jif" :)



we'll make this and save it as *phppicture.html*:

```
<HTML>
<BODY>
PHP woz 'ere:<BR><BR>
<IMG SRC="picture1.php">
</BODY>
</HTML>
```

That's all you need to get started – open up your web browser and load in *phppicture.html*. Note that I've used **imagejpeg()** here merely to demonstrate how to use the quality variable when no filename is specified – be sure to read the box *Choosing a format* for more information on which image format is best for your needs.

If you've made it this far, you will see a large black box for our image. Perhaps it's not in the same league as a Picasso, but it's a good start. The next step is to add a little colour in place of the “do stuff to the image” comment, and for this take we'll enlist the help of **imagecolorallocate()**. This new function takes four parameters – the image resource you're choosing a colour for, then three integers between **0** and **255**, one each for the red value, then green value, and the blue value of the colour. Note that more recent versions of PHP (4.2.x) also allow you to use hexadecimal (e.g. **0xff**).

The first colour you allocate is automatically used as the background colour for your image, so this next piece of code is a minor modification of the last script to include colour info...

```
<?php
$image = imagecreate(400,300);
$gold = imagecolorallocate($image, 255, 240, 00);
imagepng($image);
imagedestroy($image);
?>
```

Save that over *picture1.php*, and refresh *phppicture.html* – you should see the black square turn replaced by a golden yellow square. Not a big step forward, but a step forward nonetheless!

## Getting arty

The **imagefilledrectangle()** function takes six parameters, which are, in order: an image resource to draw on, the top-left X coordinate, the top-left Y coordinate, the bottom-right X coordinate, the bottom-right Y coordinate, and a colour to use. There is a similar function called **imagerectangle()**, which takes exactly the same parameters with the difference being that it only draws the *outline* of the rectangle, whereas **imagefilledrectangle()** fills the shape in with colour.

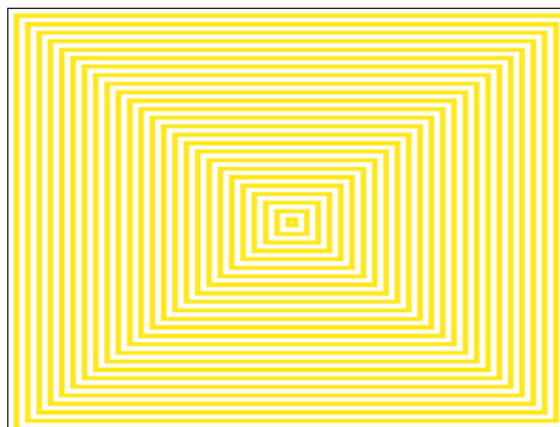
In order to draw a rectangle in such a way as to make it stand out, we need to allocate another colour, then draw the rectangle:

```
$white = imagecolorallocate($image, 255, 255, 255);
imagefilledrectangle($image, 10, 10, 390, 290, $white);
```

Put those two lines just after the definition of **\$gold**, then save the modified script and refresh *phppicture.html*. Not surprisingly, we now have a rectangle in there.

In case you're getting bored with simple stuff, let's take a leap forward with **imagefilledrectangle()** and draw a pattern using a loop. Take a copy of *picture1.php* and call it *picture2.php* – be sure to modify your HTML so that the image **SRC** points to *picture2.php*. In your favourite editor modify *picture2.php* to this:

```
<?php
$image = imagecreate(400,300);
$gold = imagecolorallocate($image, 255, 240, 00);
$white = imagecolorallocate($image, 255, 255, 255);
for ($i = 400, $j = 300; $i > 0; $i -= 4, $j -= 3) {
    if ($color == $white) {
        $color = $gold;
```



**Lots of rectangles. Don't stare at it too long otherwise you may get a headache!**

```
} else {
    $color = $white;
}
imagefilledrectangle($image, 400 - $i, 300 - $j, $i, $j, $color);
}
imagepng($image);
imagedestroy($image);
?>
```

The code is pretty much the same as before, with the addition of a simple **for** loop. Note that our loop sets the starting value for **\$i** and **\$j**, and also decrements **\$i** and **\$j** with each iteration.

As you can see in the code, we call **imagefilledrectangle()** each iteration of the loop, slowly making the rectangle smaller and smaller as **\$i** and **\$j** decrease in value. Save the script, and see how it looks in your web browser. If it's not the same as the screenshot, *above*, check your code over and try again.

## More shapes

Take a look at this next chunk of code. All being well, you should be able to spot the three new functions:

```
<?php
header("content-type: image/png");
$image = imagecreatetruecolor(400,300);
$blue = imagecolorallocate($image, 0, 0, 255);
$green = imagecolorallocate($image, 0, 255, 0);
$red = imagecolorallocate($image, 255, 0, 0);
imagefilledellipse($image, 200, 150, 200, 200, $red);
imagefilledellipse($image, 200, 150, 180, 180, $blue);
imagefilledellipse($image, 200, 150, 50, 50, $red);
imagefilledarc($image, 200, 150, 200, 200, 345, 15, $green,
IMG_ARC_PIE);
imagefilledarc($image, 200, 150, 200, 200, 255, 285,
$green, IMG_ARC_PIE);
imagefilledarc($image, 200, 150, 200, 200, 165, 195,
$green, IMG_ARC_PIE);
imagefilledarc($image, 200, 150, 200, 200, 75, 105, $green,
IMG_ARC_PIE);
imagepng($image);
imagedestroy($image);
?>
```

**imagefilledellipse()** and **imagefilledarc()** are pretty guessable once you know about **imagefilledrectangle()**, and the functions work much in the same way. **imagefilledellipse()** takes the following parameters: image resource, centre of ellipse (X coordinate), centre of ellipse (Y coordinate), height, width, and colour. **imagefilledarc()** takes the few extra parameters necessary to be able to draw an arc, >>

## Why are GIFs so bad?

Or, “Why are PNGs so good?” GIFs are bad for three reasons;

- They're patented by Unisys, and no one likes patents. PNG is a free alternative, and free is good.
- They're bigger. PNG uses superior compression to ensure the smallest files.
- They're old. The PNG format has advanced features including alpha transparency.



# TutorialPHP

◀◀ so its parameter list is: image resource, centre X, centre Y, height, width, then the start and end points of the arc specified in degrees, followed by colour, and finally type of arc to draw. The start and end points are specified from **0** to **359**, with **0** pointing directly to the right (3 o'clock). Specifying **0** and **359** as the start and end points would give you a complete circle. The final parameter, **IMG\_ARC\_PIE** here, dictates the style of arc to draw. Using **IMG\_ARC\_CHORD** for example, draws a straight line between the starting and ending angles as opposed to **IMG\_ARC\_PIE**'s curved line. The other possibilities are **IMG\_ARC\_NOFILL** and **IMG\_ARC\_EDGED** – you can combine them together using the **|** symbol. The only exception here is **IMG\_ARC\_CHORD** and **IMG\_ARC\_PIE**, which cannot be combined together. Here's an example:

```
imagefilledarc($image, 200, 150, 200, 200, 345, 15, $green,
IMG_ARC_EDGED | IMG_ARC_NOFILL);
```

The other new function I slipped in there was **imagecreatetruecolor()**, which is used in exactly the same way as **imagecreate()**. The differences between the two are that **imagecreatetruecolor()** returns an image with a true-colour palette, whereas **imagecreate()** only lets you use 256 colours at once. Furthermore, the image resource returned by **imagecreatetruecolor()** automatically has a black background, so you needn't worry about the first allocated colour being used as the image background colour.

## Outputting text

A great use for dynamic images is outputting text that is likely to change. You may be thinking "why use images for text when HTML does it so nicely?", and that's a valid point. However, what if you want to use a non-standard font? Or if you want to overlay the text onto an image? This is where using text in images becomes useful, and that's what we'll be looking at. One small hiccup when using PHP's textual image functions is the use of fonts. PHP allows you to use TrueType (TTF) fonts, PostScript Type 1 (PS) fonts, or FreeType 2 fonts, with TTF tending to be the most popular due to the availability of fonts. Some people don't have any TTF fonts installed on their system, and some don't have Windows CDs they can use to extract fonts from. In this situation you're stuck hunting around the 'Net for fonts to use – it shouldn't be too hard a search, though, as there are many freeware font sites out there.

In this example, I'll be using the font Arial, which is stored in the same directory as my PHP script. Save this code as *addingtext.php*:

```
<?php
header("content-type: image/png");
$image = imagecreate(400,300);
$blue = imagecolorallocate($image, 0, 0, 255);
$red = imagecolorallocate($image, 255, 0, 0);
$white = ImageColorAllocate($image, 255,255,255);
if(!isset($_GET['size'])) {
    $size=26;
} else {
    $size = $_GET['size'];
}
if(!isset($_GET['text'])) {
    $text = "Hello, world!";
} else {
    $text = strtoupper($_GET['text']);
}
imagefttext($image, $size, 0, 60, 140, $white, "ARIAL", $text);
imagepng($image);
imagedestroy($image);
```

## Help

There's now a messageboard on the LXF homepage dedicated to the discussion of PHP. I for one read it every day, so feel free to post questions, comments and requests there – you can be guaranteed many others will read it too and help if possible!

## GD and PHP

### Installation notes

PHP uses either *GD* or *GD2* to draw its graphics. Many of the examples above require *GD2* to work (check the PHP manual for specific information) and I recommend you use *GD2*. As of PHP 4.3, the *GD* code inside PHP has been forked in order to end the *GD* misery experienced by many PHP developers while trying to compile PHP.

Note that *GD* versions prior to 1.6 support GIF and not PNG, whereas versions after 1.6 support PNG and not GIF. You will also need to obtain and install *jpeg-6b* and recompile

*GD* to make use of this library in order to use JPEG images. To use the Type 1 fonts, you'll need *t1lib*.

Rasmus Lerdorf, the creator of PHP, has a short guide on how to get PHP working with *GD2*, and it's available at [www.php.net/~rasmus/gd.html](http://www.php.net/~rasmus/gd.html). Configuring PHP for use with *GD* has been a nightmare for a long time now, which is why they're branching it in 4.3. Don't be afraid to post to mailing lists if you have a PHP/*GD* problem – you won't be the first, and you certainly won't be the last!

?>

You should by now be familiar with the first few lines, so I'll skip them. The first *new* lines are where we set a **\$size** variable based upon the **GET** variable **size**. This allows users to set the variable directly via the `script.php?size=26`, or, if they don't set a size, we set a default of 26. The same is then done with **\$text**, except that we uppercase the user's value.

Next comes the important function, **imagefttext()**. This function takes the following parameters: the image resource to draw on, font size to use, angle to draw at, X coordinate, Y coordinate, colour, font file, and the text to write. A few of those parameters are exactly the same as parameters we've used in other functions – the new ones, though, are font size in points, angle, name of font, and the text to print.

The angle parameter works somewhat in the same manner as the angle parameters used in **imagefilledarc()** with the difference being that it works in the opposite direction – the angle in **imagefilledarc()** works in a clockwise direction from 3 o'clock, whereas **imagefttext()** works anti-clockwise. The parameter used in naming the font you wish to use is not really obvious, and the PHP manual explains it best: "Depending on which version of the *GD* library that PHP is using, when fontfile does not begin with a leading '/'; '.ttf' will be appended to the filename and the library will attempt to search for that filename along a library-defined font path." Phew! In translation, that



A post-modern neo-Raphaelite work of art. Perhaps not.

means “if you don’t specify an absolute path when using GD2, it will search for the file.” If you leave off the .ttf extension in the function and put the file in the same directory as your script, it will work just fine. You can see this in action above, where I pass the string **ARIAL** to use the font ARIAL.TTF.

The final parameter is the text to print, which is obvious enough. Note that you have to use **\n** to start a new line, not one or the other. Try changing around the parameters of **imagefttext()** to see what kind of results can be achieved with the minimum of effort – changing the angle looks particularly cool.

## Finishing up with fills

Compared to everything you’ve seen so far, using fills is remarkably easy. The function **imagefill()** takes just four parameters: an image resource, the X and Y coordinates to start the fill at, and the colour with which to fill.

The fill will automatically flood your image with colour outwards from the point specified by your X and Y parameters until it encounters any other colour. There is another similar function, **imagefilltoborder()**, where the colour to fill is the fifth parameter, and the new fourth parameter is the colour at which the fill should stop “flowing”. That is, the fill will keep flooding outwards until it hits the border colour.

First, try putting this **imagefill()** function call into the previous script, just after **imagefttext()**:

```
imagefill($image, 0, 0, $red);
```

With that function, our red colour is used to fill in the image starting from (0,0), which is the top-left corner. If you load the script into your web browser, you’ll see the fill has left some parts of the blue behind – the parts it couldn’t “reach” inside the text. Also, you’ll notice there’s a blue-ish fringe around the text, where the white text was smoothed against the blue background, producing a blue-white edge to the text.

Now try replacing the **imagefill()** with this:

```
imagefilltoborder($image, 0, 0, $white, $red);
```

Whereas the **imagefill()** function will fill the image with colour until it encounters any other colour, the **imagefilltoborder()** function call shown above will fill the image with colour and not stop till it finds pixels coloured with **\$white**. When you look at it in your browser, you’ll notice the text has become very jagged because our red fill has taken away all the blue-white smoothing.

With an arsenal of rectangles, ellipses, arcs, text, and fills, you’ve got yourself a good headstart in creating images from scratch. However, some of the best bits are yet to come...

## Working with existing images

While basic shapes look fairly good, especially when you combine them in clever ways, one of the most common tasks is to load an existing image and modify it in some way.

I hope you’re not entirely surprised to hear that, through the image library it uses, PHP makes loading existing images as easy as is possible. It has a collection of functions that all begin with “imagecreatefrom”, for example **imagecreatefrompng()** takes one parameter – the PNG file to load – and creates an image resource from it for you to use, whereas **imagecreatefromjpeg()** takes the same one parameter and loads a JPEG image.

To demonstrate working with existing images, I’ll also be using **imagecopyresampled()** and **imagecopyresized()** – two functions that resize a source image (our input picture, loaded from disk), with the difference being that **imagecopyresample()** smoothly

interpolates the pixels when resizing in order to retain clarity.

Naturally, this makes it slightly slower than **imagecopyresized()**, but there’s a noticeable difference in quality.

```
<?php
header("content-type: image/png");
$src_img = imagecreatefrompng("lxfshot.png");
$src_x = imagesx($src_img);
$src_y = imagesy($src_img);
$dest_x = $src_x / 1.5;
$dest_y = $src_y / 1.5;
$dstd_img = imagecreatetruecolor($dest_x, $dest_y);
imagecopyresampled($dstd_img, $src_img, 0, 0, 0, 0,
$dest_x, $dest_y, $src_x, $src_y);
imagejpeg($dstd_img, "", 100);
imagedestroy($src_img);
imagedestroy($dstd_img);
?>
```

As you can see, I load in a PNG file from the same location as my script. In my example, it’s a screenshot taken from the front page of the excellent *LXF* web site. Like **imagecreate()** and **imagecreatetruecolor()** before it, **imagecreatefrompng()** returns an image resource, which we file away in **\$src\_img**.

Then two new functions are called, **imagesx()** and **imagesy()**, which return the width and height of the image resource passed as their parameter respectively. From these values we set **\$dest\_x** and **\$dest\_y** to be the original size of the script divided by 1.5. This should create a picture that’s 66% the size of the original, which is perfect for the test. Next we use **imagecreatetruecolor()** to prepare a destination image resource to use for the upcoming resampling. The resampling itself is accomplished through **imagecopyresampled()** which takes a big heap of parameters: source image resource, destination image resource, four zeroes (I’ll explain in a moment), the width of the destination and the height of the destination, then finally the width and height of the source.

The four zeroes allow you to specify regions within the source and destination image. This allows you to take only a part of an image, resample it, and place it into only a part of another image – a little too advanced for this introduction! However, there is full information in the PHP manual online.

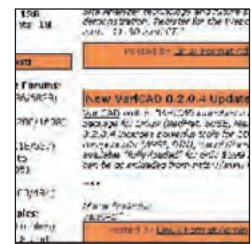
Finally, the image is outputted as a top-quality JPEG and the two image resources are saved. You will, of course, need to change the parameter passed to **imagecreatefrompng()**, but other than that the script is complete – save it as “*imagemanip.php*” and see how it looks in your browser.

All being well, you should see your input image shrunk to 66% of its size, with smooth pixels. Now try replacing **imagecopyresampled()** with this...

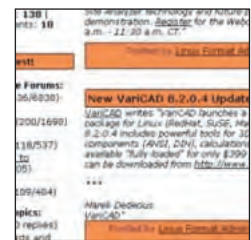
```
imagecopyresized($dstd_img, $src_img, 0, 0, 0, 0, $dest_x,
$dest_y, $src_x, $src_y);
```

When viewing that in the browser, you should see a substantial difference in quality. As you can see in the screenshots *above right*, the *LXF* website is entirely unreadable when **imagecopyresized()**, with **imagecopyresampled()** yielding much higher quality results.

Dynamic resizing as shown above is great to create thumbnail scripts, although to save time you will probably want to save the thumbnails as they are generated to avoid major slow-down. Perhaps the easiest way to implement this is to count the number of files in your images directory, compare it against the number that were last time the script was run, and if the number is different then regenerate the thumbnails. [LXF](#)



The *LXF* front page resized down to 66%. Compare that to...



...the *LXF* front page resampled down. Much smoother, and actually legible.

### About Paul Hudson

Paul Hudson is a London-based web developer specialising in PHP and Perl. He can be emailed at [hudzilla@php.net](mailto:hudzilla@php.net).

## NEXT MONTH

Next month I’ll be continuing the multimedia creation mini-series with an introduction to Flash movie generation. If you thought image creation was tricky, you’re in for a surprise! Of course, that’s more than balanced by the fact that Flash looks much more impressive :) If you have any comments or suggestions about this series, please be sure to write in!



# Answers

If you are really stuck and the HOWTOs yield no good result, why not write in? Our resident experts will answer even your most complicated problems!

## Our experts

Whatever your question is, we can find an expert to answer it – from installation and modem woes to network administrations, we can find the answer for you – just fire off a letter or email and it'll all be taken care of.

LXF answers guy  
**David Coulson** is a networking and security guru with plenty of sysadmin experience to boot.



**Nick Veitch** is the editor of the magazine, and answers your easy questions! Or indeed anything to do with *Grub*, *LILO*, *netatalk*, vi...



## Missing headers

**Q** I am quite new to Linux and was wondering if you could please help me. A few months ago I purchased Mandrake 8.2 Standard Edition. Having read *Linux Format* for sometime before that, the first thing I did after installation was to try some of the programs on the DVDs. I managed to install things like *OpenOffice.org*, but when it came to installing from source code I got errors about *gcc*. I thought maybe I hadn't installed it, so re-installed Mandrake making sure I selected *gcc* and it's dependencies. So once again I went back to the LXF DVDs and tried the usual unpack, *./configure*, *make*, *make install*, and still I got the same error:

```

creating cache ./config.cache
checking for non-GNU ld... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... found
checking for working autoconf... found
checking for working automake... found
checking for working autoheader... found
checking for working makeinfo... found
checking host system type... i686-pc-linux-gnu
checking whether to enable maintainer-specific portions of Makefiles... no
checking for a BSD compatible install... /usr/bin/install -c
checking for gcc... gcc
checking whether the C compiler (gcc) works... no

```

configure: error: installation or configuration problem: C compiler cannot create executables.

Please, please, please can you help me? Thanks in advance.

**Adam Forster**

**A** You need to ensure that you have all of the *gcc* RPMs installed, along with the *glibc-devel* packages. Most *configure* scripts produce a *config.log* file in the directory you execute it from, so you may wish to check that to see what exactly it is complaining about. The *configure* script will attempt to compile a test program to figure out if the compiler works properly or not. There are a number of reasons why this may not compile, such as unavailable C header files, incorrect *gcc* installation or something wrong with the *configure* script.

There is no need to completely reinstall each time you want to add packages. You can simply use the *rpm* command to install RPMs from the distribution CDs in order to add packages to the system.

## DHCP

**Q** I have installed a Red Hat Linux 7.3 Professional on a K6 500MHz computer with

256MB RAM and a 40GB harddisk. In this computer there is an ethernet card of this type:

D-LINK DFE-530TX PCI Fast Ethernet Adapter (Rev A)

This ethernet card is connected to a Zywall 1 router that give me the IP address and other TCP/IP information. The Zywall 1 router is connected to an ADSL modem. On the Zywall 1 router there is also connected a computer with Windows 2000 Professional. I use this Windows computer to connect to my company (my office at home). My question is now how I get connected to the Internet from Linux? I have tried to read the enclosed documentation, but I cannot find a useful answer to this question. I hope that you can help me with an answer of this question.

I hope to hear from you.

**Jan Christensen**

**A** The DFE-530TX NIC uses the *via-rhine.o* kernel module, so you will need to ensure that is either compiled into the kernel, or available as a module. Once the module is loaded, you should have an *eth0* device, which you can configure with *ifconfig*. Depending how your

```

david@macha:~ (pts/27)
Linux Administrator's Manual
PUMP(8)
NAME
    pump - configure network interface via BOOTP or DHCP protocol
SYNOPSIS
    /sbin/pump [-krBsd?] [-c ARG] [-h hostname] [-i iface] [-l hours]
    [--lookup-hostname] [--usage]
DESCRIPTION
    pump is a daemon that manages network interfaces that are controlled by
    either the DHCP or BOOTP protocol.
    While pump may be started manually, it is normally started automati-
    cally by the /sbin/ifup script for devices configured via BOOTP or
    DHCP.
    Once pump is managing an interface, you can run pump to query the sta-
    tus of that interface. For example,
    /sbin/pump -i eth0 --status
    will print the current status of device eth0.
CONTEND-LINE OPTIONS
    Manual page pump(8) line 11

```

*pump* is a very powerful DHCP client, which is useful if you don't know your network configuration.

router is setup, you will need to set the eth0 interface up in Linux with an appropriate IP address and gateway.

You will need to use a DHCP client, such as *dhclient* or *pump* to do a DHCP request and get an IP from your router. If you use *pump*, you just need to do:

```
# pump -i eth0
```

This will automatically set everything up, including the default gateway and DNS information. You should not need to do anything else to get Internet access and you should be able to set Red Hat up to automatically perform a DHCP request by using the network configuration tools.

## Keeping (fs)tabs

**Q** The best distro you have given out to try yet is Debian 3.0. I just love the way APT works. I did not find the system that difficult to get installed, but I don't have everything running correctly, yet. The main problem is the mouse has not installed properly. Although I put it in as a NetScroll PS/2, and looking through the files, that is what it has recognised. I used *mdetect*, but it made no difference. It will not work, which means keyboard only. Not the best way in 'X'.

Also whilst the system showed my DVD (IDE) in *fstab*, as */cdrom*. It did not put my SCSI CDwriter and Zip drives in. Any idea why? I have *cdrom*, */cdrom0* and */cdrom1*, but I only have two CDs? I will probably manage to put the writer in, but I have no idea what to do to get the Zip drive recognised. I do know that if *fstab* is not correct, I will not be able to boot, and I am not proficient enough to replace the file in rescue mode.

An article on doing a replacement for 'fstab' or similar, would maybe help others beside myself. I always await the magazine dropping onto themat each month. Keep up the good work.

John Bywater

**A** When you run X, it should create */var/log/XFree86.0.log*, which you can check to figure out why your mouse isn't working. If you check */etc/X11/XF86Config*, you can ensure that it is using */dev/psaux* and the correct driver is selected. You will need to use the IMPS/2 driver in order to use a mouse with a scroll wheel correctly. Usually X will fail completely if it can't find the mouse or can't open the */dev* file, so you're probably not using the correct driver, rather than not having the mouse set up correctly.

*fdformat* supports a number of difficult disk geometries, but you also need to create a filesystem.

The SCSI CD-RW drive should be available as */dev/scd0* and the Zip drive as */dev/sda4*. You can check the boot logs with the *dmesg* command to ensure that the specific device is being used. You'll need to be sure that the correct SCSI driver is compiled into your kernel, or available as a loadable module, although which one you use will depend upon the SCSI adaptor you have in your system. You can have Debian load the module at boot time by inserting the module name into */etc/modules*, along with any support modules required, such as *sr\_mod* for the CD-RW drive.

You may want to take a second look at */etc/fstab* and ensure that Debian did not put anything in there for your CD-RW drive or your Zip drive, although if it has not then you will need to do it yourself. */etc/fstab* is fairly straightforward, as you can simply cut and paste the other entries in for your CD-RW and Zip drives.

## Big floppy

**Q** You mentioned in the MiniDistro roundup, the ability to format floppy disks too 1.722MB.

Well tell me how. Many thanks

Phil Morgan

**A** The *fdformat* utility is used to format a floppy disk and you must use the appropriate */dev/fd0\** file, in this case */dev/fd0u1772*: *fdformat /dev/fd0u1772*

Once the disk has been formatted, you will also need to create a filesystem on it, such as *ext2*, *vfat* and so forth. This is usually done with *mke2fs*, in the case of an *ext2* filesystem. *mke2fs /dev/fd0*

## Fragmentation

**Q** It's been three months since I started to use Linux. In all the books, magazines, etc. I read, I cannot find any hint whether I should do a regular *scandisk* or defrag on my Mandrake box.

What are the regular tasks I should do to keep my system happy? How do I go about performing them? And if no such tasks are needed, why not? (would be one more good reason to dump Windows)

Bruno

**A** Ext2 is generally smarter than FAT32, so there is no need to defragment the filesystem

## A QUICK REFERENCE TO: qmail

*Sendmail* is certainly the most well known Open Source mailserver available, but it is by no means the only one, nor the best. Of course, picking a mail server is one's personal choice, but a particularly powerful alternative to *Sendmail* is the hugely popular *qmail*.

*qmail* has a very simple configuration system, which relies on having individual files within */var/qmail/control*, rather than one massive configuration file which is a nightmare to maintain. *qmail* was designed with scalability and virtual domain hosting in mind, so there is no need for ugly configuration options to make it handle more than one domain, plus users can handle their own mail aliases. This removes some administration control from the postmaster, but it does mean that their job is much easier as all they

have to do is point a domain to a particular user, rather than add a whole load of aliases for each domain.

dot-qmail files are a particularly important part of *qmail*, as these decide how email will be handled by the server once it has been accepted. For example, if we have a domain 'domain.tld' controlled by the user 'bob', bob will get all email for the domain sent through his *qmail* configuration. If we email 'test@domain.tld', then *qmail* will first check for *~bob/.qmail-test*, then for *~bob/.qmail-default*. We can, of course, create separate configurations for individual domains, so rather than having 'domain.tld' controlled by 'bob', we can have it controlled by 'bob-domain.tld'. Now *qmail* will check for *~bob/.qmail-domain.tld-test*, then for *~bob/.qmail-domain.tld-default* and finally for *~bob/.qmail-*

default. For local mailboxes, *qmail* will use *~bob/.qmail*, so if you send a mail to 'bob' on the local machine, *qmail* can handle it differently to a virtual domain.

These dot-qmail files can either forward email to another email account, or insert it into a mailbox. As well as regular mbox format files, *qmail* can also use Maildirs, which are generally considered to be more flexible and less prone to problems than mbox files. If an mbox file becomes corrupt, then all your email is gone. If a Maildir entry is corrupt, then you lose one email, which is naturally much better. It is also quite easy to index files in a Maildir, so it's much quicker to pick out an individual message from the Maildir. Not all mail clients support Maildirs, although *mutt* and the popular *Courier IMAP* client will use Maildirs on a *qmail* system.



# FREQUENTLY ASKED QUESTIONS: USER-MODE LINUX

## FAQ What the heck does User-mode Linux do?

User-mode Linux, otherwise known as UML, is a patch for the Linux kernel tree which allows the Linux kernel to be compiled as a stand alone ELF binary. The kernel can then be executed within user-space, hence the name User-mode Linux, as a regular process. The result is that you end up with one Linux system running within another Linux system.

Resources on the Linux system are mapped into the UML, depending upon what the user wants the UML kernel to have access to. Disk storage for the virtual machine is entirely contained inside a single file on your physical machine. You can assign your virtual machine only the hardware access you want it to have. With properly limited access, nothing you do on the virtual machine can change or damage your real computer, or its software.

## FAQ Er, OK. So what's the point?

As the UML kernel is entirely separate from the main Linux installation, then we can effectively secure the main host from problems which occur within the UML. The most obvious use is as a sand box or jail, allowing

a network service to run within a more powerful version of chroot.

Of course, as UML is a Linux kernel, it has also found its place for kernel development, so anyone writing kernel-level code for a non-hardware part of the kernel can debug it in user space before running it on a physical system. A number of individuals and organisations are using UML for this purpose, including the ReiserFS team, Alan Cox and the FreeSWAN developers.

## FAQ How do I build a UML kernel?

The UML kernel patch can be grabbed from <http://user-mode-linux.sf.net/dl-sf.html>, and it should be applied to a vanilla kernel source tree. It may apply to a -pre or -rc kernel, but it's not unlikely that it won't compile, or it won't work happily.

Once you have your UML kernel patch and your kernel tree, you simply:

```
tar xzf linux-2.4.19.tar.gz
cd linux-2.4.19
bzcat ../uml-patch-2.4.19-32.bz2 |
patch -p1
make mrproper ARCH=um
make menuconfig ARCH=um
make linux ARCH=um
```

Building UML should normally be done as a non-root user, so anyone can build a UML kernel in their home directory. Once the kernel has

User-mode Linux allows Linux to be executed as a regular binary, for security or development.

compiled, you can simply do

```
./linux
```

and hopefully it will boot up just as it would on a physical system, although it will complain about not having a root file system eventually.

## FAQ What about the rest of the distribution? A kernel isn't going to do much on it's own, is it?

Indeed it won't. A number of filesystems are distributed on SourceForge, so you can simply grab a Debian or Red Hat filesystem and run it under UML. It's not always best to run a downloaded file system directly, as it won't have

any free space available.

Creating an empty filesystem and copying everything over is quite easy:

```
dd if=/dev/zero of=rootfs bs=1k
count=1 seek=$[2*1024*1024]
/sbin/mke2fs rootfs
```

which will create a 2GB ext2 filesystem. This can then be mounted over the loopback device, along with the downloaded file system, and the contents can be copied, using **cp -fra** to ensure that permissions are saved, although this will have to be done as root.

## FAQ Can I do networking with UML?

Of course! Linux wouldn't be much use

◀ frequently. It will automatically rearrange things to improve throughput and to avoid huge gaps in the filesystem. The Linux version of *scandisk* is *fsck*, which is generally run once every ten or so reboots, meaning that you don't have to run it manually and you can usually just forget about it. Alternatively, you can run a journalled filesystem, such as *ext3*, *XFS* or *ReiserFS*, which is checked quickly at every mount so that anything which wasn't written correctly when the system was rebooted can be recovered, or at least it will try.

Every day Linux runs a number of scripts, which are located in `/etc/cron.daily/`, which includes logfile rotation scripts and a system which uses *updatedb* to index every file on the system so that *locate* will work

correctly. There is often little need to change these scripts, although if the system is frequently off during the night it may be a good idea to change `/etc/crontab` to run the scripts at a different time, as many distributions run their daily *cron* jobs somewhere between 2am and 6am.

More info on modifying your crontab can be found with **man 5 crontab**.

## Frog driver

**Q** I am currently running a WindowsXP driven system at this moment, but I am looking to go back to a dual boot system either using Debian 3.0r0 or SuSE 8.0 but what is my problem at this moment (Debian is my first choice out of the two Linux's mentioned): "how do I setup my Speedtouch

USB modem?" Before I switched to broadband I had SuSE 8.0 running with an old narrowband modem, but now I have not had any luck! Can you help Me? I have tried setting up SuSE before, but with no luck for my efforts! My modem works well on Windows, but who wants to be using Windows forever!

Preferably I would like to dump it in the bin, and go totally over to Linux, possibly running two different versions as a dual boot. Which flavour of Debian would you recommend for use with my modem?

Ivor Jones

**A** I'm surprised you've not been able to find information on this, as using the 'frog' is a popular topic with Linux. Both Debian and SuSE will work with the Alcatel

Speedtouch ADSL modem, as it requires kernel support and *pppoe* or *pppoea*, all of which both distros can provide.

There is a comprehensive HOWTO on the Speedtouch ADSL modem, at <http://linux-usb.sourceforge.net/SpeedTouch/howto.html>. You will want to run the current stable release of Debian, which at the time of writing is 3.0r0, otherwise known as Woody, although you may wish to ensure that you are running a recent 2.4 kernel with this install to ensure that you do not have any problems with drivers which may not function quite right in 2.2.

## Print run

**Q** I have a problem with OpenOffice.org 1.0.1 and printing which I can't seem to get around. I am running SuSE 8.0

without networking. UML supports a number of networking methods, including SLIP, multicast and TUN/TAP. The TUN/TAP transport is the most often used, as it provides a virtual Ethernet connection, so a variety of protocols can be run over the virtual 'patch cable'. The TUN/TAP interface can also be included as part of an Ethernet bridge on the host, meaning that the UML can get direct access to the LAN, so it looks like a regular host.

To use the TUN/TAP transport, the latest *uml-utils* package must be compiled and installed, as the *tunctl* or *uml\_net* programs are needed. We can use *tunctl* as root to create a tap device, and then run UML with the other end of this device as its ethernet device:

```
root:~# tunctl -u david
Set 'tap0' persistent and owned
by uid 500
david:~$ ./linux-2.4.19/linux
eth0=tuntap,tap0 mem=64M
ubd0=rootfs
```

### FAQ Where can I find more information on UML?

There are a number of sites which are useful for those interested in UML. There is the official UML kernel site, which can be found at <http://user-mode-linux.sf.net/>. There is also a community site, itself hosted on a UML system, at <http://usermodelinux.org/>.

on a Toshiba Satellite Pro 4600 laptop, 128MB RAM, 10GB HDD, with *OpenOffice.org* 1.0.1.

When I originally installed *OOo* I was running the *lprng* print spooler and could print from *OOo* with no problems. A few weeks ago I decided for various reasons to change my print spooler to *CUPS* using the *gimp-print* and *Foomatic* drivers. *CUPS* installed perfectly and all of the programs including *OOo* printed OK but *YaST2* warned me that the *CUPS* daemon and the *lprng* daemon would not work together, so I removed *lprng* just leaving *CUPS*.

Again everything appeared OK until I tried to print from *OOo* when it reported that *lpr* command did not exist which was not surprising

as I had removed *lprng* but I cannot figure out how to get *OOo* to print using *CUPS* alone as the *CUPS* documentation gave *lp* and *lpr* as their printing examples. Can you suggest how I can get around this apparent quandry.

Many thanks in advance

David M. Swain

**A** It sounds like you wiped out *lpr* when you removed *lprng*, so the quickest way to solve this issue is simply to reinstall *CUPS*, which should provide *lpr* for you. You can check this works by running *lpr* at the command line, as per the *CUPS* examples.

## Boot choice

**Q** I dual-boot Linux and Windows on a server that has no monitor. I use *Lilo*'s **LOCK** option to make the system boot whatever was run last. If I want to *change OS*, I use an attached Wyse serial terminal to intercept *Lilo* before it boots the default (i.e., the **LOCK**ed) OS.

What I'd *really* like, is to be able to nominate the OS to boot *before* rebooting. Of course, such a facility would have to work under Linux, Windows and whatever other OSs I might want to run.

Have you come across such an app? Would there be any benefit, with regard to the above requirement, in adopting *GRUB* instead of *Lilo*?

**A** The only option we can think of is to modify the **default=** line in *lilo.conf*, then rewrite

the *Lilo* block in the MBR. There is no nice way to do this via any OS other than Linux, so you're a little out of luck there. Of course, as you have the console available on a serial port, you could quite easily create a script to wait for the *Lilo* prompt and then feed it a new command line. Quite how you implement this would depend upon how you choose to select the OSs, although the *expect* utility could come in quite handy.

## Cape Perls

**Q** Linux started out being a week-end play-around with RH7.2 and somehow I have included a Linux course into the usual MS nonsensical stuff I teach.

I hope to get the course a little more filled out, out here in beautiful Cape Town, so that we can have 50% of the courses going on Linux!

Anyhow, the questions are fairly minor, but the reason for asking is important.

I'm finding ways to show people on these courses that Linux can do things cheaply, and as such, I love the program on the cover CDs called *YoSucker*. It's a mail 'sucker' program which sucks all Yahoo mail into one's *Kmail* client, say ... saves oodles of time online in a country where dial-up time is fairly expensive, so seeing banner-ads is particularly annoying...

First question: I'm looking for a Perl module called **Term::ReadKey** which does not seem to be there after installing Perl 5

Second question: Having given *OpenOffice.org* an installation run, it fires up and does ... *absolutely nothing!* The splash-screen starts then it just dies.

Did I miss something?

If I could get both of these working, there would be a very convincing reason for people to take a harder look at Linux as I definitely am taken with its capabilities. It would be nice for people not to believe it would take ten years of IT training to master and enjoy.

I seem unlucky with the programs I have chosen off the CDs so I was wondering if anyone checks these babies?

From the LXF forums

**A** Taking your first question. You can grab **Term::ReadKey** from

<http://search.cpan.org/dist/TermReadKey/>, although if you have installed packages of Perl 5 then you will likely want to install packages of **Term::ReadKey**, rather than compiling it, as it will make maintaining the system quite difficult. Of course, if you compiled Perl 5 yourself, then you will likely have two versions of Perl installed, as you will have one in */usr/bin* and the other in */usr/local/perl5/bin*.

For your second problem, you should check the console you have executed *OpenOffice.org* from, in order to find out if the installation program is segfaulting or exiting for some other reason. If you are running this through a GUI, then you will likely not notice any errors and it will quit without telling you anything.

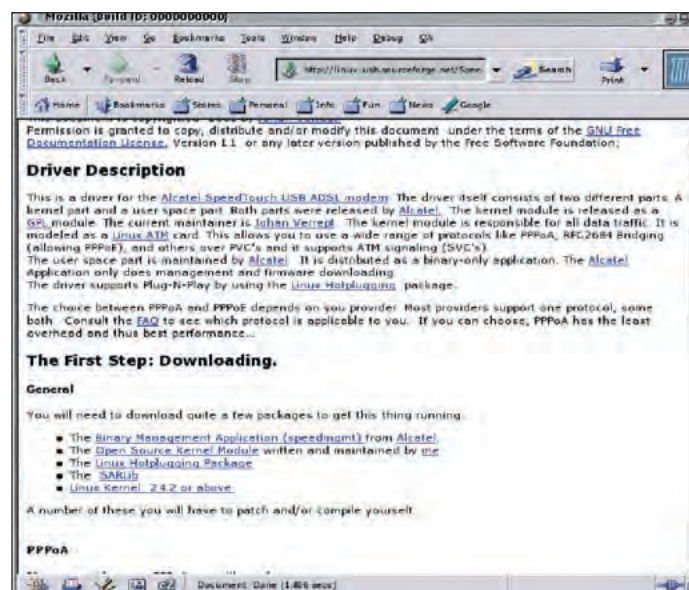
## Timed out

**Q** Hi I am trying to schedule a daily backup with *crontab* in Mandrake 8.2. I have edited my *crontab -e*. The job starts and 30 seconds later stops. I have tried writing a script and pointing *cron* at it but the same happens. I have checked Google for too long and have seen nothing relevant. The entry in *crontab* is:

```
*18 *** tar -gzip -cvf
/home/phil/backup/daily_working.tar
/home/phil/working
```

It is driving me to distraction – please can someone help, it seems so simple but I must be doing something wrong.

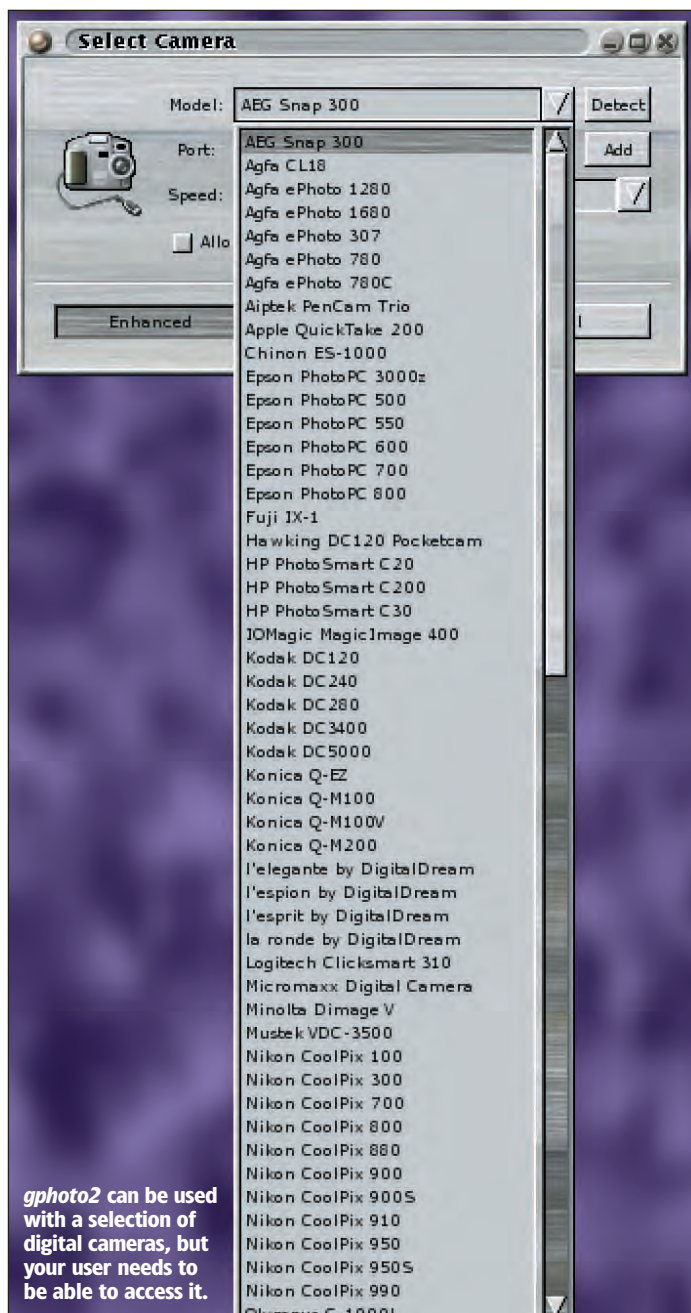
From the LXF forums



There is a ton of documentation available for the Speedtouch ADSL modem, for those who have the frog.



# Answers



*gphoto2 can be used with a selection of digital cameras, but your user needs to be able to access it.*

“**A** Your cronjob will run every minute when the hour is 6pm, so rather than running it once, you will infact be running it sixty times. You can modify your crontab just to run it once at 18:00 with:

```
0 18 *** tar cvfz /home/phil/backup/
daily_working.tar.gz
/home/phil/working
```

When the job has finished, it will email you the output to your system mailbox, usually in **\$MAIL**, so you can check the output from that. You will want to check that it creates /home/phil/backup/daily\_working.tar.gz, and also check upon the contents of this file to see how far it gets doing the backup, if it does anything at all.

## Photo smart

**Q** I have an HP Photosmart 318 digital camera. It is in the list when I start **gtkam** and try to make the setup. But when I try to connect to the camera it says, “could not initialise camera.” Whats wrong?

I have just installed Red Hat 8.0, so I'm quite a newbie. So please could the answer be something I can understand? Thanks.

*From the LXF forums*

**gtkam** is a front end for the **gphoto2** utility, so it's worth checking out if **gphoto2** works. We're assuming that the camera is USB, so you will want to try the following as root:

```
gphoto2 --auto-detect -P -R
```

which will search for the camera and automatically download the photos from it. However, as it says it could not initialise the camera, it may mean that the camera is not ready for a PC to connect to it. Hopefully, running **gphoto2** manually should help you.

If it works as root, then you will need to check out the **gphoto2** documentation and learn how to set hotplug up for your camera, as you will want the entry in **/proc/bus/usb** to change its UID to that of the user you will be fetching the photos as, otherwise it will not be able to connect to the camera.

## PAM problem?

**Q** I have set up a Linux server and workstation at the school I work, looking to expand to an entire classroom. I have set up NFS and NIS on the server, and set up the NIS client on the workstation, however when I come to login it says the the user or password is incorrect. However if I login as root and use **su** to change the user it works fine. Can you shed any light on this subject. Any help would be appreciated.

*From the LXF forums*

**A** NIS requires a few modifications to **/etc/passwd** and **/etc/nsswitch.conf** before you can authenticate with it. You said that you could **su** to the user, so it is likely able to read the UID and shell information from the server, but you may want to modify the **nsswitch.conf** file, as per the NIS documentation:

```
passwd: compat
group: compat
shadow: compat
```

```
passwd_compat: nis
group_compat: nis
shadow_compat: nis
```

However, as you are able to **su** to the UID, then it could also be a PAM problem, depending upon which distribution you are using on the client.

Most distributions handle this automatically, but you may want to check the NIS HOWTO to check upon anything you may need to modify for your own specific setup.

## Secure shelling

**Q** I use VNC over **OpenSSH**. The machine I connect to is also a gateway. I'm wondering if there is a way to tell

either **VNC** or **OpenSSH** just to serve on the **eth1** interface and not the **eth0** interface.

I use **iptables** to drop anyone trying to get in so I reckon the configuration is as safe as can be reasonably expected. I have blocked the port number I want to use on **eth0**.

I just wanted to know if this is the best solution or is there a better way?

*From the LXF forums*

**A** You can't modify which interfaces **SSH** listens on, but you can modify the IPs it will listen on, so they can only connect to it if they connect to the IP associated with a specific interface.

To do this, you use the **ListenAddress** option: **ListenAddress 192.168.1.1**

Of course, you can have as many entries as you like, so it can listen on more than one IP address. It's open to argument which is most secure, as you can easily log connections with **iptables**, plus you can more comprehensively limit who can connect to the server, depending on source IP and so forth. **LXF**

## Submission advice

We are happy to answer all sorts of Linux related questions. If we don't know the answer, we'll find out for you! But in order to give you the best service, it helps a lot if you read the following submission advice.

- Please be sure to include any relevant details of your system. "I can't get X to work" doesn't really mean anything to us if we don't know things like what version of X you are trying to run, what hardware you are running on.
- Be specific about your problem. Things like "it doesn't work" or "I get an error" aren't all that helpful. In what way does something not work? What were you expecting to happen? What does the error message actually say?
- Please remember that the people who write this magazine are NOT the authors or developers of Linux, any particular package or distro. Sometimes the people responsible for software have more information available on websites etc. Try reading the documentation!

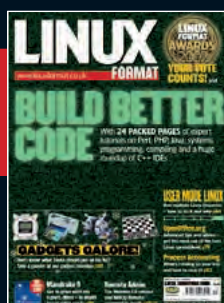
We will try and answer all questions. If we don't answer yours specifically, you'll probably find we've answered one just like it. We can't really give personal replies to all your questions.

**WRITE TO US AT:**  
Linux Format, Future Publishing, 30  
Monmouth Street, Bath BA1 2BW or  
email: [lxf.answers@futurenet.co.uk](mailto:lxf.answers@futurenet.co.uk)

# missed one?

## LINUX FORMAT BACK ISSUES

Every month *Linux Format* brings you the best tutorials, the essential reviews and the latest news. But if you've missed out on a must-read feature or a vital programme from our expertly compiled CDs and DVDs, order your back issue NOW! And remember, you need never miss an issue of your favourite Linux mag, if you subscribe to *Linux Format* (see overleaf for more details).



### Christmas 2002

Product code:  
**LXFB0035**(cd)  
**LXFD0035**(dvd)

**DVD HIGHLIGHTS:**  
Mandrake 9.0 – 3 CD  
download edition; jEdit;  
BastilleLinux; Phoenix;  
VideoLAN; KDE 3.0.4

**MAGAZINE FEATURING:**  
Huge C++ IDE RoundUp;  
LXF Awards nominations;  
gadgets for Xmas; User  
Mode Linux; compiling code;  
process accounting

**CDs HIGHLIGHTS:**  
3CDs, including 2CD  
Mandrake 9.0; SpamX;  
Stellarium; Krename;  
ImagePress; Nogger;  
DigitalMusicCentre



### December 2002

Product code:  
**LXFB0034**(cd)  
**LXFD0034**(dvd)

**DVD HIGHLIGHTS:**  
KDE 3.1 beta2, Scribus,  
Smoothwall2, Movix,  
GNUPrivacyGuard,  
GCompris, Drip

**MAGAZINE FEATURING:**  
Distro reviews of SuSE 8.1,  
Red Hat 8.0, Lycoris and a  
mini-distro RoundUp,  
scanning, PHP acceleration,  
Linux Expo UK 2002

**CDs HIGHLIGHTS:**  
Unreal Tournament  
2K3 (Demo),  
DemoLinux, Cardfile,  
RUTE, OOOictionaries,  
Mozilla binary packages



### November 2002

Product code:  
**LXFB0033**(cd)  
**LXFD0033**(dvd)

**DVD HIGHLIGHTS:**  
Debian 3.0, mFighter,  
OpenOffice.org (bugfix  
release), DVDrip,  
Cyrus IMAP Server

**MAGAZINE FEATURING:**  
Sun's move into the Linux  
server market – with LX50  
review, The Liberty Alliance,  
Systems programming, using  
OpenOffice.org, Homebase

**CDs HIGHLIGHTS:**  
Kylinx 3, GNUCash,  
BXPro, KDevelop,  
Opera, Vega Strike,  
Parted, AnjutaIDE,  
GTransferManager



### October 2002

Product code:  
**LXFB0032**(cd)  
**LXFD0032**(dvd)

**DVD HIGHLIGHTS:**  
Knoppix, Drip, Squeak,  
extra FlightGear maps,  
Ogg Vorbis 1.0, Knoppix

**MAGAZINE FEATURING:**  
Building better databases,  
'Trusted Computing' –  
beware Palladium, USB 2.0,  
firewall roundup, Amiga  
emulation, Gentoo review

**CDs HIGHLIGHTS:**  
FlightGear (runs from  
disc), Aglaophone, UAE,  
Clam Antivirus, Perl  
5.8, Quanta Plus,  
Netclipboard, Mah-  
Jong, HTML-Mason,  
WebSuck, Epsutil



### September 2002

Product code:  
**LXFB0031**(cd)  
**LXFD0031**(dvd)

**DVD HIGHLIGHTS:**  
Slackware 8.1, Cinelerra,  
Ogle & Zine, Gnome2,  
MultiCD, Phobia III,  
Grip, Zinf, TkVoice

**MAGAZINE FEATURING:**  
Linux goes to Hollywood,  
HTML Editors roundup,  
Internet security special,  
Ruby scripting language

**CDs HIGHLIGHTS:**  
Slackware 8.1, Fluxbox,  
Kallers, Torcs,  
SaveMyModem



### August 2002

Product code:  
**LXFB0030**(cd)  
**LXFD0030**(dvd)

**DVD HIGHLIGHTS:**  
Intel's C++ & Fortran  
compilers, CPAN,  
Gnome2, Gentoo  
Linux, Boson, Mozilla,  
Normalize, EarCandy

**MAGAZINE FEATURING:**  
Ultimate Office: Every current  
office solution on test, plus  
Quantum computing, Red  
Hat 7.3, PHP tips, Eden  
motherboard test

**CDs HIGHLIGHTS:**  
Intel's C++ & Fortran  
compilers, GNOME2,  
Transcode, Album  
Cover Grabber, Office  
Suites

Other back issues also available, please call for further information, giving the issue number and/or date required

# To order, call the back issues hotline on 0870 4448645

or +44 1458 271178 for overseas orders

Remember to quote the issue code shown above and  
have your credit card/switch card details ready.

**Prices (inc post and packing):** CD – UK£5.99,

US & Eur £6.99, rest of world £7.99

DVD – UK £6.99, US&Eur £7.99, rest of world £8.99.

# LINUX

www.linuxformat.co.uk **FORMAT**



## Benefits – why subscribe?

- **Subscribe by Direct Debit – it's easy! Fill in the form below or call 0870 444 8645\* and have your bank details ready.**
- **Huge savings on a year's supply of *Linux Format*.**
- **Subscription copies are sent hot off the press – be one of the first to get yours!**
- **Guarantee your copy – never miss an issue.**
- **All you need to do is fill in the form. If you live in the UK, you don't even need a stamp!**

\* or call the overseas number shown at the bottom of the coupon, below

### RISK-FREE GUARANTEE!

You can cancel your subscription at any time and receive a full refund on all unmailed issues, no questions asked

Title \_\_\_\_\_ Initials \_\_\_\_\_ Surname \_\_\_\_\_  
 Address \_\_\_\_\_  
 \_\_\_\_\_ Postcode \_\_\_\_\_ Country \_\_\_\_\_  
 Email address \_\_\_\_\_  
 Daytime telephone number \_\_\_\_\_



Media with passion  
 Future Publishing Ltd, Cary Court,  
 Somerton, Somerset, TA11 6BR



Originator's Identification Number

7 6 8 1 9 5

Please send this form to address shown

1. Name and full postal address of your Bank or Building Society branch

To: The Manager

\_\_\_\_\_ Bank/Building Society

Address \_\_\_\_\_

Postcode \_\_\_\_\_

2. Name(s) of account holder(s)

3. Branch sort code

(from the top right-hand corner of your cheque)

4. Bank or Building Society account number

5. Instruction to your Bank or Building Society

Please pay Future Publishing Direct Debits from the account detailed on this Instruction subject to the safeguards assured by the Direct Debit Guarantee. I understand that this instruction may remain with Future Publishing and if so, details will be passed electronically to my bank or building society.

Signature(s) \_\_\_\_\_ Date \_\_\_\_\_

Ref No (Office use only) \_\_\_\_\_

Banks and Building Societies may not accept Direct Debit Instructions for some types of account.

You may photocopy this blank form and then complete it. You cannot fax or e-mail Direct Debit forms.

QUOTE ORDER No. **LXFP36**



Please choose your method of payment **1** or **2**

**1.** ☐ Direct Debit (UK only – double CD Edition) – I wish to pay £27.45 every 6 months

or ☐ Direct Debit (UK only – DVD Edition) – I wish to pay £29.95 every 6 months

**Please complete the form on the left**

I understand that I will receive 13 issues during the next 12 months.

**2. Cheque/credit card 13 issues**

	CD	DVD
<b>UK</b>	<input type="checkbox"/> £54.90	<input type="checkbox"/> £59.90
<b>Europe (including Eire)</b>	<input type="checkbox"/> £78.90	<input type="checkbox"/> £83.90
<b>North America</b>	<input type="checkbox"/> £78.90	<input type="checkbox"/> £83.90
<b>Rest of the World</b>	<input type="checkbox"/> £91.90	<input type="checkbox"/> £96.90

☐ Cheque (payable to Linux Format. Sterling cheques drawn on a UK account.)

☐ Visa

☐ MasterCard

☐ American Express

☐ Switch

Issue No/Valid date \_\_\_\_\_

Card No \_\_\_\_\_ Expires \_\_\_\_\_

Signature \_\_\_\_\_ Date \_\_\_\_\_

**UK READERS RETURN THIS COUPON TO:**

*Linux Format*, Subscriptions, Future Publishing Ltd, Freepost BS4900, Somerton, Somerset, TA11 6BR.

**OVERSEAS READERS RETURN (POSTAGE PAYABLE) TO:**

*Linux Format*, Subscriptions, Future Publishing Ltd, Cary Court, Somerton, Somerset, TA11 6TB, UK

**WEB:** [www.futurenet.com/secure/subs1.asp?ID=92](http://www.futurenet.com/secure/subs1.asp?ID=92)

☐ Please tick this box if you would prefer not to receive information on other offers.

**ORDER HOTLINE ☎ 0870 444 8645**  
**OR FROM OVERSEAS ON +44 870 444 8645**

**OFFER ENDS: 31.01.03**



# EVERY ISSUE IS A SOFTWARE BONANZA!

Every issue *Linux Format* comes with either 2 CDs or a DVD crammed full of the latest Linux software. Get all this delivered direct to your door every month with a subscription costing from just £27.45 for six months!

In the last six months you would have received on DVD:

The latest **FULL LINUX DISTROS** including ● Mandrake 9.0 ● DemoLinux ● LinuxFromScratch ● Sorcerer Linux ● Knoppix ● Red Hat 7.3 ● Slackware 8.1 ● Debian 3.0 ● Gentoo  
**PLUS** Over **750** of the best Linux apps and utilities.

Don't miss these great software packages – save yourself hours of download time with every issue of *Linux Format*!

## « Subscribe from just £27.45

every 6 months by Direct Debit  
saving £22.97 a year (CD issue)

DVD issue from £29.95 for 6 months





# Essential disc info

Read this important information before you use your *Linux Format* coverdisc – CD or DVD.  
We've collated some helpful info to help you get the most from these jewels of data!

## Finding the essentials

### Missing something?

As many of the programs on our discs are the very latest releases, they are often built on the very latest libraries and may depend on other packages your current Linux setup does not contain. We try to provide you with as many of these important supporting files and libraries as possible, though obviously we don't have space to include absolutely everything.

In many cases the latest libraries and

other packages you might need will be included in the "essentials" folder on the disc, so if you are missing dependencies, this is the first place to look.

### Package formats

Wherever possible, we try to include as many different types of package for an installation as possible, whether that be distribution specific RPMs, debs or whatever. Please bear in mind that we can only do this where space permits and when the packages are available.

We will, apart from exceptional or legally restricted situations, include the source files for any package, so that you can build it yourself.

### Documentation

These pages provide helpful information on how to install and use some of the packages on the CD. Please note that many of the applications come with their own documentation, and there are additional notes and files in the relevant directories.

## What are all these files?

If you are new to Linux, you may find the profusion of different files and extensions confusing. As we try to give as many packages as possible for compatibility, there will often be two or three files in a directory covering different types of Linux, different architectures and usually source and binary versions – so which do you install? They can be identified by their filenames, and usually just by the file extensions.

Someap-1.0.i386.rpm – This is probably a binary rpm, designed to run on x86 systems.

Someap-1.0.i386.deb – The same, but a debian package.

Someap-1.0.tgz – This is usually source code.

Someap-1.0.tgz – Same as the above, tgz is abbreviated form of tar.gz

Someap-1.0.tar.bz2 – Same, but uses bzip2 compression instead of zip

Someap-1.0.src.rpm – This is also source code, but supplied as an rpm to make it easier to install

Someap-1.0.i386.RH7.RPM – A binary, x86 RPM designed specifically for Red Hat Linux

Someap-1.0.ppc.Suse7.rpm – A binary RPM designed specifically for SuSE7.x PPC Linux.

Someap-devel-1.0.i386.rpm – A development version.

## Installing from tarballs

A tar ball is a two stage archive. First the files are archived into a single file with **tar** and then compressed with **Gzip** or **Bzip2**. To unpack, **cd** to the directory you want to unpack it, usually your home directory and type one of the following two lines:

```
tar xzvf /mnt/cdrom/Desktop/progname/progname-2.1.0.tgz
```

```
tar xvf -bzip2 /mnt/cdrom/Desktop/progname/progname-2.1.0.tar.bz2
```

Use the first for Gzipped files, those ending in .tar.gz or .tgz, and the second for Bzipped files, ending in .tar.bz2 or .tbz2. Naturally, you change the paths to suit the location and name of the archive. and replace /mnt/cdrom with whatever is applicable to your system (e.g. /cdrom). This normally unpacks the archive into a directory of the same name, enter that directory with:

```
cd progname-2.1.0
```

To compile and install the software, type the following three commands:

```
./configure
```

```
make
```

```
su -c "make install"
```

The last line will prompt you for the root password, as this stage must be run as root. If you are already logged in as root, just type **make install**. This will give you a default installation. If you want to change any aspect of the install, type **./configure --help** to see the options available. For example, you are usually able to change the default location with the **PREFIX** argument. When you have finished installing, you may remove the source files with:

```
cd ..
```

```
rm -fr progname-2.1.0
```

You should also log out as root, before you do anything you may later regret.

## Defective CDs

In the unlikely event of your CD/DVD being physically damaged we'll send you a new, working version within 28 days. Send your defective disc – complete with your name, address, and a description of the fault – to:

**Linux Format, Future Publishing Disc Department, 3B Athena Avenue, Elgin Industrial Estate, Swindon, SN2 8HF.**

## Creating install CDs with cdrecord

The quickest way to burn an ISO image to CD is with **cdrecord**. You need to be root to do this. First find the address of your CD-writer with

```
cdrecord -scanbus
```

This will show the devices connected to your system. The SCSI address of each device is the three numbers in the leftmost column, say 0,3,0. Now you can burn a CD with

```
cdrecord dev=0,3,0 -v  
/path/to/image.iso
```

You can simplify the command by saving some default settings in /etc/default/cdrecord. Add a line for each CD writer on your system (usually one) like this

```
Plextor= 0,3,0 12 16M
```

The first item is a label, after the SCSI address you put the speed and the buffer size to use. You can now replace the SCSI address in the command line with the label, but it gets even easier if you add

```
CDR_DEVICE=Plextor
```

Now you can burn an ISO image to disc with

```
cdrecord -v/path/to/image.iso
```

If you really don't want to use the command line, **gcombust** will do the job for you. Start it as root, select the "Burn" tab and the "ISO 9660 Image" gadget near the top of the window. Put the path to the image file in the gadget and press "Combust!". Now put on the kettle while the CD is created for you.

## Other OS?

You do not have to use Linux to burn the ISO to a disc. All the Linux-specific bits are already built into the image file. Programs like **cdrecord** simply dump it to the disk. If you don't have a CD-writer, find someone who does have one, and a DVD drive, and use the CD burning software on their computer. It can be Windows, MacOS, AmigaOS whatever.

## No CD burner?

What if you have no CD writer? Do you know someone else with one? You don't have to use Linux to burn the CDs, any operating system that can run a CD-writer will do the job (see above).

With some distributions it is also possible to mount the images and do a network install, or even a local install from another disk partition. The methods often vary between distributions, so check on the distro vendors website for more info.

# Coverdisc



**Neil Bothwick** is your guide through the wonders of this month's jam-packed *Linux Format DVD*, as we take the easy approach to source-based distros...

Lucky CD users get another bonus CD this month, that's nearly 2GB of material. Everything that is on the three CDs is also on the DVD, so have a look at those pages to see part of what you've got. In addition, DVD users get a full 2GB of extra material. So if you have the DVD issue, here's the second part of what's on your coverdisc. If you don't have a DVD drive, here's 2GB of reasons why it's worth upgrading your hardware.

## DISTROS LRS-LINUX

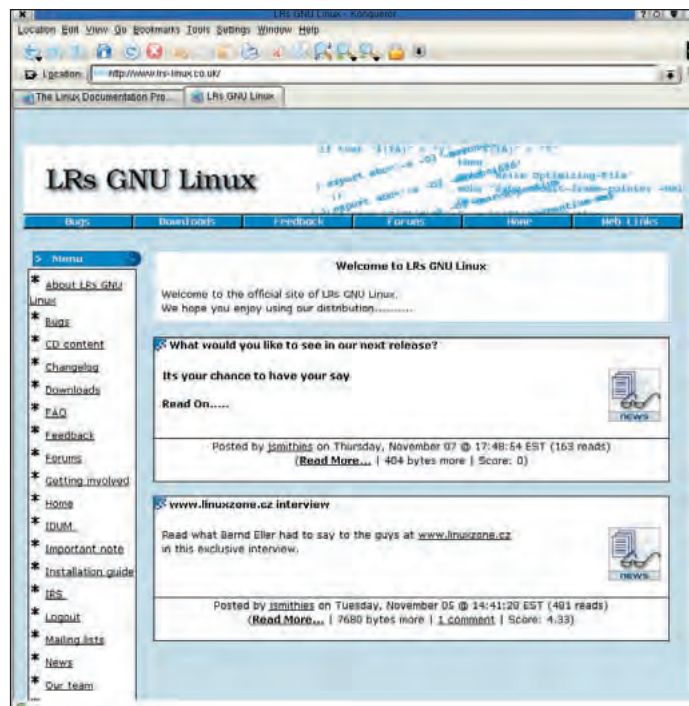
The extra space on the DVD gives us the opportunity to include extra Linux distributions for you to try. In recent

months we have included a number of the big name distros, such as Mandrake, Debian and Slackware. The recent upgrade frenzy, where just about every major distro released a new version, is over for now. We have given you the latest releases of all the major distributions we are allowed to put on the coverdiscs. This gives us the chance to include some of the smaller – both in size and market share – distros.

If you've read about Linux From Scratch on the CD pages, you may like the idea of a distro compiled from source specifically for your hardware, but be less keen on the time and effort involved to do so. LRs Linux is a source-based distro, based on Linux From Scratch. LRs is a relatively new distribution, and a fairly small one for general purpose use. This is good for those wanting a lightweight and efficient Linux system. The installer is text based. While being a lot easier to set up than a raw Linux From Scratch system, it still requires more user knowledge and input than the point and click installers of the major RPM based distros.

## GRAPHICS DVDRIP

If you are reading this you probably have a DVD drive, and if you have a DVD drive you probably use it to watch DVD movies. *DVD-Rip*, as the name suggests, is a program to rip movies from DVDs and store them on your hard disk, but it does a fair bit more than that. Before anyone starts jumping up and down over piracy issues, there are legitimate reasons for ripping movies from DVDs you own. One is that hard drive space is cheap, many of us have far more than we use. Ripping movies to disk means we can watch what we want without rummaging in the DVD box to find the right disc first. Although the discs themselves hold over 4GB, the movies



## Linux From Scratch made easier, with LRs Linux.

themselves are often much smaller, so disk storage requirements aren't as great as you may think. This is reduced even more because *DVD-Rip* does a lot more than copy the movies from DVD to hard disk, it uses *transcode* as its back end to process the data before saving it. So you can save the file in a more compact format, such as DivX;-).

The use of *transcode* opens up other options too. Some PDAs now have video players. You could use *DVD-Rip* to reduce the resolution of a movie to suit a PDA screen, and reduce the bitrate to fit a full feature film onto a single compact flash card. Let others listen to their Walkmans on the train, you could be watching the latest blockbuster.

## MOBILE OPENZAURUS

We tend to concentrate on desktop and server software, because that's

what is most commonly available. However, Linux works on many other devices too, like Sharp's Zaurus PDA.

*Open Zaurus* is a replacement ROM project for the Zaurus. Originally based on the standard Zaurus ROM with a few enhancements, it has gradually moved away, adding its own functionality. However, it is also compatible with the vast majority of Zaurus applications.

Installation consists of copying the *initrd.bin* images and one of the kernel images to a compact flash memory card and performing the standard ROM flash actions, just the same as if you were installing a Sharp ROM update. Looking on the DVD, you'll immediately notice that there are a choice of kernel images, these differ in the way that memory is allocated.

The standard ROM splits the 64MB into two equal portions, one for system memory and one as a storage

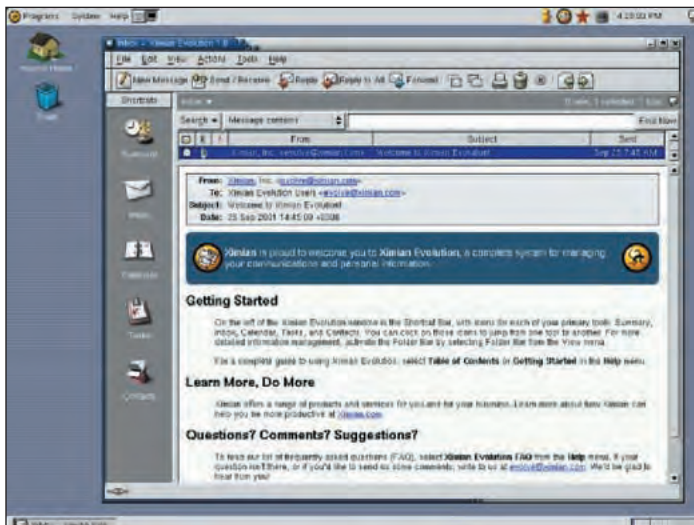


Wherever you see this logo it means there's related stuff on the DVD

## IMPORTANT NOTICE

Before you even put the DVD in your drive, please make sure you read, understand and agree to the following: The *Linux Format DVD* is thoroughly tested for all known viruses, and is independently certified virus-free before duplication. We recommend that you always run a reliable and up-to-date virus-checker on ANY new software. While every care is taken in the selection, testing and installation of DVD software, Future Publishing can accept no responsibility for disruption and/or loss to your data or your computer system which may occur while using this disc, the programs or the data on it. You are strongly advised to have up-to-date, verified backups of all important files. Please read individual licences for usage terms.





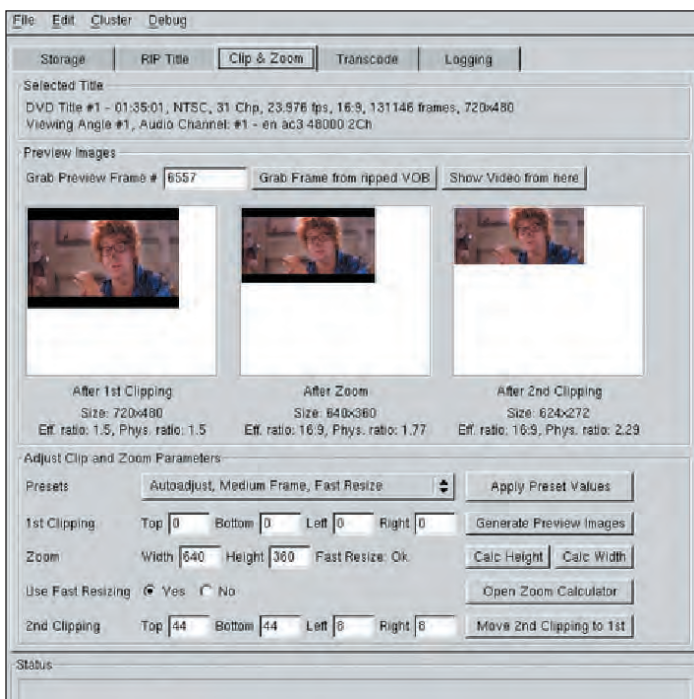
The all-in-one mailer, contact and information manager just got updated. Try the new *Evolution* here.

device on which /home is mounted. This is fine if you have no other storage on the PDA, but wasteful of memory if you have a CF or SD card mounted as a storage device. In that case, you should use one of the kernels that gives more of the on board RAM to the system.

There are two important points to consider before installing *Open Zaurus*. The first is to read the instructions thoroughly before you try reflashing the device. The second is to make sure you have one of Sharp's ROM update files available, so you can go back to the old ROM if you decide you don't like *Open Zaurus*.

## INTERNET EVOLUTION

The Unix way is generally to use a number of small tools, each one specifically designed for one particular task. However, many people prefer their Internet packages to have a more all-in-one approach. *Mozilla* is a good example of this, combining web, mail, news, IRC and other functions in one large package. *Evolution* is another such program, being a comprehensive email and personal information management program. While this approach may not be everyone's preference, there's no denying that



Copy films from DVD to HDD, processing them on the way, with *DVD-Rip*.

## Roll your own CDs

### Missing file workaround

DVD users get Freeduc bootable from the DVD, but what if you want to make copies to show off the capabilities of Linux to your friends? We have included *jigdo* again, this time in the Distros/Freeduc directory, to create ISO images from the DVD. However, the day after the DVD was mastered, a problem showed up which prevents the Linux version of *jigdo* from working as supplied. This is caused by a missing file but, fortunately, the solution is simple.

Copy the Freeduc directory to your hard disk, as you would have to do anyway. Make sure the partition you copy to has at least 650MB of free space. Now *cd* to this directory in a terminal window and type:

```
tar xjf jigdo/jigdo-bin-0.6.8.tar.bz2
mv jigdo-bin-0.6.8/jigdo-file jigdo/
```

This extracts the necessary file from the archive already on the disc and puts it where it is needed. Now continue with the normal procedure by typing:

```
sh mkiso
```

When you see the message that begins with:

If you already have a previous version of the CD...

You should enter:

```
/mnt/cdrom/KNOPPIX
```

This tells *jigdo* where to find the files on the DVD. The process is much faster than on previous LXF cover DVDs because all of the files *jigdo* needs are in their own directory, so it doesn't have to scan the whole DVD.

## Building Mandrake CDs from last month's DVD

The same problem with *jigdo* was present on last month's DVD. The solution is almost identical, copy the *jigdo* directory to your hard disk and type:

```
tar xjf jigdo/jigdo-bin-0.6.8.tar.bz2
mv jigdo-bin-0.6.8/jigdo-file jigdo/
```

Apologies for this hiccup, but this is a new system for us and, like most new systems, it has its glitches. This particular problem won't affect any subsequent DVDs.

```
./configure
make
make install
```

There are no dependencies on external libraries, so you should be able to install like this on any system, with no other installations needed beforehand.

## ESSENTIALS LDP

If you have a Linux question, there is a very good chance that the Linux Documentation Project will contain the answer. This is a vast and comprehensive collection of HOWTOs, books, FAQs, man pages and two online magazines, *Linux Gazette* and *Linux Focus*. The main site is at [www.tldp.org](http://www.tldp.org) with various mirrors around the World for faster access, but what if you don't have a fast Internet connection, or the very problem you need help on is preventing you from using the Internet? Every month, the DVD contains a full mirror of the Linux Documentation Project. This is as up to date as our production schedules allow, usually being updated only a few hours before the DVD is mastered. Some distro-specific questions are best asked on that distro's support forums, but just about everything else is covered in here. [LXF](#)

## DESKTOP OPENBOX

Which window manager is best? I doubt that question will ever be resolved, with KDE3 and GNOME2 being the main protagonists. One side-effect of the contest to make each window manager more useful and add more features is that they tend to rely more and more on the user having reasonably powerful hardware. That is why there will always be a place for lighter and faster window managers, environments that may not have quite so many features, but which run more efficiently. *Openbox* is one such window manager (another is *Fluxbox*, which we have featured on previous coverdiscs). In keeping with the small and efficient approach, *Openbox* is supplied as a source code tarball, installed with the usual method, run as root.

# CoverdiscDVD

## DVD CONTENTS AT A GLANCE

### Desktop

<b>Arson</b>	KDE frontend for CD burning and ripping
<b>BMI</b>	A program that calculates your BMI (Body Mass Index)
<b>GNUPrivacyGuard</b>	GPLed PGP replacement tool
<b>I-Man</b>	PHP/Interbase contact and information database
<b>IOzone</b>	Filesystem benchmark utility
<b>Kmplot</b>	Mathematical function plotter for the KDE desktop
<b>Kmuser</b>	User-Administration-Tool for the KDE-Desktop
<b>MySQLControlCenter</b>	GUI client for MySQL databases
<b>Openbox</b>	Fast, slim window manager for X11 written in C++
<b>ParallelPortPowerSwitch</b>	TCP/IP server to control a parallel port power switch
<b>ReallySlickScreenSavers</b>	collection of OpenGL screensavers
<b>Skylander</b>	Modern astrology software for Linux/KDE3
<b>Twin</b>	Text-mode window manager and terminal emulator
<b>Unarc</b>	Unpack archives into their own directory
<b>Workar</b>	Highly configurable graphical Filemanager for X
<b>X11DiskActivityFeedback</b>	Shows disk activity by animating the X11 cursor

### Development

<b>Autoconf</b>	Produce scripts that automatically configure sourcecode
<b>CVSSpam</b>	Notification of CVS commits, by email
<b>CVSToys</b>	CVS enhancement that provides commit notifications
<b>GNOMEStructuredFileLibrary</b>	
	Read and write OLE and Zip files
<b>GTK+-DirectFB</b>	Port of GTK+ to the DirectFB graphics system
<b>IDEStudio</b>	Enhanced version of Python Idle Editor
<b>Jaxor</b>	An object/relational code generation tool
<b>libwebserver</b>	Library for adding Web interfaces to your programs
<b>NMM</b>	Multimedia middleware package
<b>PHPEclipse</b>	PHP support for the Eclipse IDE Framework
<b>PicoGUI</b>	Complete GUI system for embedded systems
<b>RubyInline</b>	Embedded C code in Ruby scripts
<b>Self</b>	Port of the Self system to Linux on x86 PCs
<b>SmartCardToolkit</b>	Smart card library and tools
<b>TCLpython</b>	Python interpreter which can be embedded into TCL
<b>XBasic</b>	Cross Platform BASIC programming language

### Distros

<b>LRs-Linux</b>	Distribution
<b>PXES</b>	Distro for thin clients or converting PCs into thin clients
<b>TrinityOS</b>	Guide to securing, tuning and enabling services
<b>TrustixSecureLinux</b>	Secure Linux distribution for servers
<b>Warewulf</b>	Distributed Linux distribution

### Games

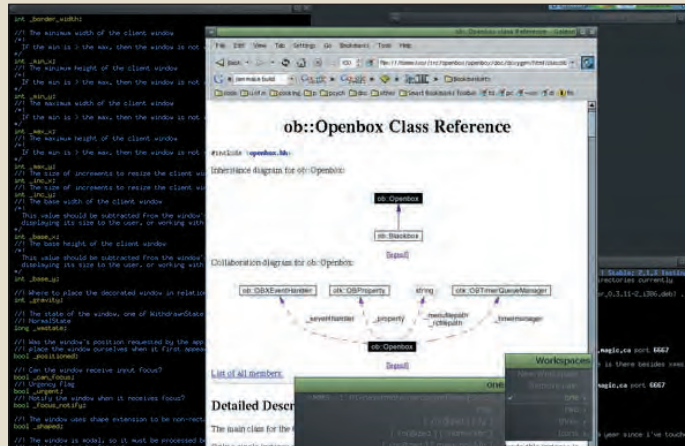
<b>Boson</b>	Network real-time strategy game like warcraft, without AI
<b>BZFlag</b>	3D multiplayer tank battle game
<b>Frotz</b>	Portable Z-Machine interpreter
<b>glAnts</b>	An open source mech game
<b>HL-webadmin</b>	Web-based controller for a dedicated Half-Life server
<b>KrystalDrop</b>	Free clone of NeoGeo's Magical Drop
<b>LabyrinthOfWorlds</b>	Ultima Underworld II rewrite

### Graphics

<b>AutoTrace</b>	Converts bitmap to vector graphics
<b>AVconvert</b>	Record, convert, and edit video files using a C++ interface
<b>DVDrip</b>	Full featured DVD Ripper GUI
<b>Freevo</b>	Standalone Linux digital VCR/PVR
<b>Goggles</b>	An interface for the Ogle DVD player
<b>MplayerXP</b>	mplayer with extra performance
<b>TheGimp</b>	The GNU Image Manipulation Program
<b>Xine</b>	Unix video player

### Internet

<b>AnnoyanceFilter</b>	Bayesian junk mail filter
<b>DownloaderForX</b>	Downloads files from the Internet via both FTP and HTTP
<b>Evolution</b>	GNOME mail client and PIM
<b>Galeon</b>	GNOME Web browser
<b>LimeWire</b>	Powerful Gnutella file sharing client with great features
<b>Linphone</b>	An SIP-compatible Web phone with a GNOME interface
<b>MasqMail</b>	Offline Mail Transfer Agent
<b>MSNlib</b>	An MSN Messenger library and text-mode client
<b>PictureCacher</b>	Downloads and archives pictures to display them on the Web
<b>SILC</b>	Secure Internet Live Conferencing



Want a lighter, faster alternative to KDE and GNOME? then try Openbox.

### Tneclean

A script to clean up Outlook attachments in incoming email

### Mobile

**Affix**  
**Backgammon**  
**gnuPod**  
**GPSdrive**  
**MadBomber**

Open source Bluetooth protocol stack for Linux  
Backgammon game  
Collection of tools for accessing the iPod under UNIX  
Displays your GPS position on a zoomable map  
Clone of the excellent 1981 Atari 2600 video game "Kaboom!"  
Alternative ROM image for the Sharp Zaurus PDA  
Suite of tools for connecting to PalmOS handheld devices  
Applet that allows you to set screenshots  
Windows CE and Pocket PC connectivity for Linux  
Record time spent on projects  
GTK frontend to Kismet  
Configures Linux' Wireless Extension

### Office

**Ledger**  
**Moodle**  
**ProjectButler**

An income and expense tracker  
Course management system for distance education  
Distributed multi-user project management software

### Server

**Back-End**  
**CyrusIMAPServer**  
**DoSEvasiveManeuversModule**

Simple template driven PHP/MySQL CMS  
Full featured IMAP server  
Apache DoS Evasive Maneuvers Module  
Simple HTML-compressor  
Helps to manage the Web bookmarks of an organization  
HTML menu building wizard  
PHP-Nuke module for genealogy  
intranet system designed to be used at LAN parties  
Extensible content management framework with object reuse  
The database backend for MySAP

**HTMLcrunch**  
**LinkBase**  
**MenuWorkshop**  
**MyNukeGenealogy**  
**phpCrystal**  
**PHPortal**

### SAPDatabase

### Sound

**Autoplay**  
**BlueXMMS**  
**Cecilia**

Audio CD autoplaser  
Bluetooth XMMS remote control  
Graphic user interface for CSound

### System

**ArnosIPTABLEScript**  
**BusyBox**  
**FastOnlineUpdateForSuSE**

Iptables firewall script with support for ADSL modems  
Tiny Unix utilities, for rescue disks and embedded systems

### Logwatch

**MultiCD**  
**PamMount**  
**login**

Lightweight alternative for YaST OnlineUpdate  
Script to summarize system logs  
Backup your data to CD-R/CD-RW  
SMB, NCP, or loopback-encrypted volume mounting upon  
Detects, alerts and responds to port scans  
Update Red Hat Linux systems  
Utilities to monitor S.M.A.R.T. disks and devices  
Collection of boot loaders for Linux  
WOLK with many useful patches - Base: 2.4.18  
Tools for creating a compressed ISO9660 filesystems



# User Groups

LUGs worldwide are full of members keen to help with your problems, discuss ideas, and generally natter about all things Linux. You can find lots more information online at: [www.lug.org.uk](http://www.lug.org.uk)

## 1 Hampshire

URL [www.hants.lug.org.uk](http://www.hants.lug.org.uk)  
Contact Hugo Mills

## 2 Bristol & Bath

URL [www.bristol.lug.org.uk](http://www.bristol.lug.org.uk)

## 3 Scottish

URL [www.scottish.lug.org.uk](http://www.scottish.lug.org.uk)

## 4 Oxford

URL [www.oxford.lug.org.uk](http://www.oxford.lug.org.uk)  
Contact Alasdair G Keron

## 5 Kent

URL [www.kent.lug.org.uk](http://www.kent.lug.org.uk)  
Contact John Mills

## 6 Brighton

URL [www.brighton.lug.org.uk](http://www.brighton.lug.org.uk)  
Contact Johnathan Swan

## 7 Worcestershire

URL [www.worcs.lug.org.uk](http://www.worcs.lug.org.uk)  
Email [info@thirdeyeddevelopment.com](mailto:info@thirdeyeddevelopment.com)

## 8 Northants

URL [www.northants.lug.org.uk](http://www.northants.lug.org.uk)  
Contact Kevin Taylor

## 9 Anglian

URL [www.anglian.lug.org.uk](http://www.anglian.lug.org.uk)  
Contact Martyn Drake

## 10 Milton Keynes

URL [www.mk.lug.org.uk](http://www.mk.lug.org.uk)  
Contact Denny De La Haye

## 11 Doncaster

URL [www.doncasterlug.org.uk](http://www.doncasterlug.org.uk)  
Contact Andy Smith

## 12 Moray

URL [www.moray.lug.org.uk](http://www.moray.lug.org.uk)  
Contact Stewart Watson

## 13 West Wales

URL [www.westwales.lug.org.uk](http://www.westwales.lug.org.uk)  
Contact Dan Field

## 14 Wolves

URL [www.wolveslug.org.uk](http://www.wolveslug.org.uk)  
Contact Jono Bacon

## 15 Peterborough

URL [www.peterboro.lug.org.uk](http://www.peterboro.lug.org.uk)  
Contact Steve Gallagher

## 16 Edinburgh

URL [www.edinburgh.lug.org.uk](http://www.edinburgh.lug.org.uk)  
Contact Alistair Murray

## 17 Tyneside

URL [www.tyneside.lug.org.uk](http://www.tyneside.lug.org.uk)  
Contact Brian Ronald

## 18 Leicester

URL [www.leicester.lug.org.uk](http://www.leicester.lug.org.uk)  
Contact Clive Jones

## 19 Greater London

URL <http://glug.linux.co.uk/>  
Contact John Southern

## 20 Surrey

URL [www.surreylug.org.uk](http://www.surreylug.org.uk)  
Contact Jay Bennie

## 21 Cambridge

URL [www.cam-lug.org.uk](http://www.cam-lug.org.uk)

REVISED  
DETAILS

## 22 Devon & Cornwall

URL [www.dclug.org.uk](http://www.dclug.org.uk)  
Contact Simon Waters

## 23 Falkirk

URL [www.falkirk.lug.org.uk](http://www.falkirk.lug.org.uk)

## 24 Manchester

URL [www.manlug.mcc.ac.uk](http://www.manlug.mcc.ac.uk)  
Contact John Heaton, Owen Le Blanc

## 25 Hertfordshire

URL [www.herts.lug.org.uk](http://www.herts.lug.org.uk)  
Contact Nicolas Pike

## 26 West Yorkshire

URL [www.wyylug.lug.org.uk](http://www.wyylug.lug.org.uk)  
Contact Jim Jackson

## 27 Sheffield

URL [www.shefflug.co.uk](http://www.shefflug.co.uk)  
Contact Richard Ibbotson

## 28 Staffordshire

URL [www.staffslug.org.uk](http://www.staffslug.org.uk)

## 29 North East

URL [www.shofar.uklinux.net/NELUG](http://www.shofar.uklinux.net/NELUG)

## 30 London

URL [www.lonix.org.uk](http://www.lonix.org.uk)

## 31 Thames Valley

URL [www.sclug.org.uk](http://www.sclug.org.uk)

## 32 Liverpool OpenSource

URL [http://linux.liv.ac.uk/\\_liv\\_linux\\_ug/](http://linux.liv.ac.uk/_liv_linux_ug/)  
Contact Simon Hood

## 33 Deal Amiga Club

Email [superhighwayman@hotmail.com](mailto:superhighwayman@hotmail.com)  
Contact John Worthington

## 34 Chesterfield

Email [spirelug@yahoo.co.uk](mailto:spirelug@yahoo.co.uk)  
Contact Robin Needham

## 35 South Derbyshire

URL [www.sderbylug.org.uk](http://www.sderbylug.org.uk)  
Contact Dominic Knight

## 36 Belfast (BLUG)

URL [www.belfastlinux.cx](http://www.belfastlinux.cx)  
Email [russell@belfastlinux.org](mailto:russell@belfastlinux.org)

## 37 Wiltshire

URL [www.wiltshire.lug.org.uk](http://www.wiltshire.lug.org.uk)  
Contact Jason Rudgard

## 38 South London

URL [www.sl.lug.org.uk](http://www.sl.lug.org.uk)  
Email [ben@ilovephilosophy.com](mailto:ben@ilovephilosophy.com)

## 39 Cheshire

URL [www.sc.lug.org.uk](http://www.sc.lug.org.uk)  
Contact Anthony Prime – [enquiry@sc.lug.org.uk](mailto:enquiry@sc.lug.org.uk)

## 40 North Wales

URL [www.northwales.lug.org.uk](http://www.northwales.lug.org.uk)  
Contact Jonathan Cole

## 41 Midlands

URL <http://midlandslug.port5.com/>  
Contact Pete Thompson

## 42 Cumbria

URL [www.cumbria.lug.org.uk](http://www.cumbria.lug.org.uk)  
Contact Jamie Dainton

## 43 Dorset

URL [www.dorset.lug.org.uk](http://www.dorset.lug.org.uk)  
Contact John and Mat

## 44 Shropshire

URL [www.shropshire.lug.org.uk](http://www.shropshire.lug.org.uk)  
Email [shropshire@lug.org.uk](mailto:shropshire@lug.org.uk)

## 45 South West

URL [www.southwestlug.uklinux.net](http://www.southwestlug.uklinux.net)  
Email [southwest@lug.org.uk](mailto:southwest@lug.org.uk)

## 46 South Wales

URL [www.sw.lug.org.uk](http://www.sw.lug.org.uk)  
Contact Tim Bonnell

## 47 North London – see 87

URL [www.kemputing.net/lug/anlug-aims.html](http://www.kemputing.net/lug/anlug-aims.html)

## 48 Malvern

URL [www.malvern.lug.org.uk](http://www.malvern.lug.org.uk)  
Contact Greg Wright

## 49 Huddersfield

URL [www.hud.lug.org.uk](http://www.hud.lug.org.uk)  
Contact Adam Brookes

## 50 Nottingham

URL [www.nottingham.lug.org.uk](http://www.nottingham.lug.org.uk)  
Contact Godfrey Nix

## 51 St Albans & Luton

URL [www.lust.lug.org.uk](http://www.lust.lug.org.uk)  
Contact Michael Culverhouse – [mike@easily.co.uk](mailto:mike@easily.co.uk)

## 52 Wrexham

Contact Paul Kersey-Smith  
Email [paul@pkls.fsnet.co.uk](mailto:paul@pkls.fsnet.co.uk)

## 53 Preston & Lancs

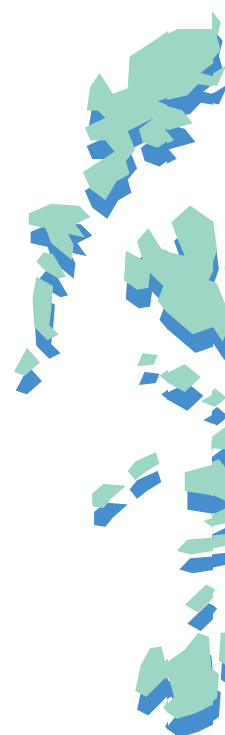
URL [www.preston.lug.org.uk](http://www.preston.lug.org.uk)  
Contact Phil Robinson

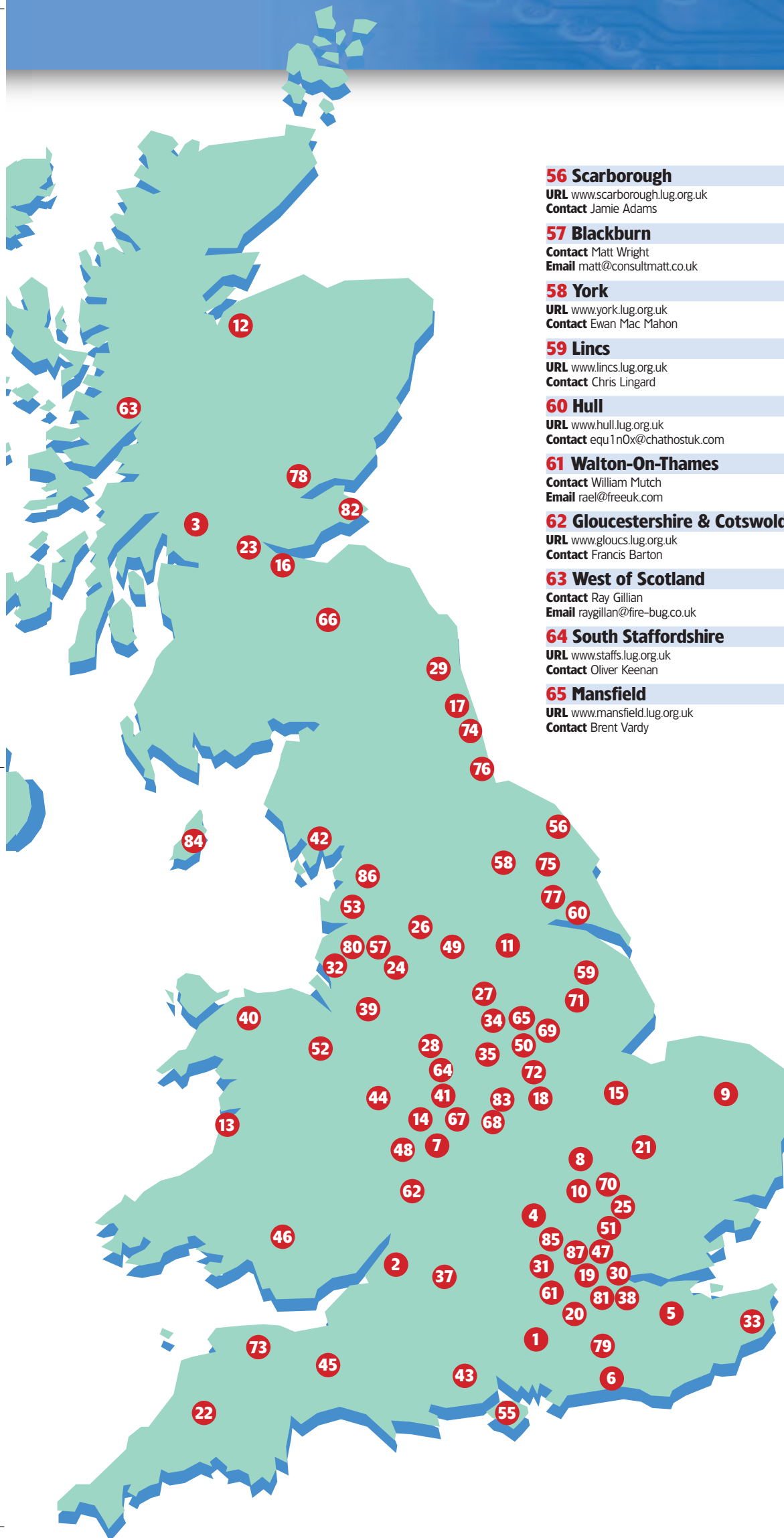
## 54 Derry

URL [www.derry.lug.org.uk](http://www.derry.lug.org.uk)

## 55 Isle of Wight

URL [www.iow.lug.org.uk](http://www.iow.lug.org.uk)  
Contact David Groom – [info@iow.lug.org.uk](mailto:info@iow.lug.org.uk)



**56 Scarborough**

**URL** [www.scarborough.lug.org.uk](http://www.scarborough.lug.org.uk)  
**Contact** Jamie Adams

**57 Blackburn**

**Contact** Matt Wright  
**Email** [matt@consultmatt.co.uk](mailto:matt@consultmatt.co.uk)

**58 York**

**URL** [www.york.lug.org.uk](http://www.york.lug.org.uk)  
**Contact** Ewan Mac Mahon

**59 Lincs**

**URL** [www.lincs.lug.org.uk](http://www.lincs.lug.org.uk)  
**Contact** Chris Lingard

**60 Hull**

**URL** [www.hull.lug.org.uk](http://www.hull.lug.org.uk)  
**Contact** [equ1n0x@chathostuk.com](mailto:equ1n0x@chathostuk.com)

**61 Walton-On-Thames**

**Contact** William Mutch  
**Email** [rael@freeuk.com](mailto:rael@freeuk.com)

**62 Gloucestershire & Cotswolds**

**URL** [www.gloucs.lug.org.uk](http://www.gloucs.lug.org.uk)  
**Contact** Francis Barton

**63 West of Scotland**

**Contact** Ray Gillian  
**Email** [raygillian@fire-bug.co.uk](mailto:raygillian@fire-bug.co.uk)

**64 South Staffordshire**

**URL** [www.staffs.lug.org.uk](http://www.staffs.lug.org.uk)  
**Contact** Oliver Keenan

**65 Mansfield**

**URL** [www.mansfield.lug.org.uk](http://www.mansfield.lug.org.uk)  
**Contact** Brent Vardy

**66 Borders**

**URL** [www.linux.bordnet.co.uk](http://www.linux.bordnet.co.uk)  
**Contact** Welby McRoberts

**67 South Birmingham**

**URL** [www.sb.lug.org.uk](http://www.sb.lug.org.uk)  
**Contact** Tim Williams

**68 Coventry**

**Contact** Darren Austin  
**Email** [info@coventry.lug.org.uk](mailto:info@coventry.lug.org.uk)

**69 Newark & Lincoln**

**URL** [www.newlinc.lug.org.uk](http://www.newlinc.lug.org.uk)

**70 Bedfordshire**

**URL** [www.beds.lug.org.uk](http://www.beds.lug.org.uk)  
**Contact** Neil Darlow

**71 Lincoln**

**URL** [www.lincoln.lug.org.uk](http://www.lincoln.lug.org.uk)  
**Contact** Jon Shamash

**72 Loughborough**

**URL** [www.loughborough.lug.org.uk](http://www.loughborough.lug.org.uk)  
**Contact** Martin Hamilton

**73 Exeter University**

**Contact** Nicholas Murison  
**Email** [N.J.Murison@exeter.ac.uk](mailto:N.J.Murison@exeter.ac.uk)

**74 Sunderland**

**Contact** Thomas Croucher  
**Email** [thomas.croucher@sunderland.ac.uk](mailto:thomas.croucher@sunderland.ac.uk)

**75 East Yorkshire**

**Contact** Daniel Gallacher  
**Email** [sharkonline@whatemail.com](mailto:sharkonline@whatemail.com)

**76 Cleveland Open Source Group**

**Contact** Haniff Din  
**Email** [openlug@digitalmedia.co.uk](mailto:openlug@digitalmedia.co.uk)

**77 Beverley**

**Contact** Vladimir Lukyanov  
**Email** [vladimir\\_lukyanov@hotmail.com](mailto:vladimir_lukyanov@hotmail.com)

**78 Dundee & Tayside**

**URL** [www.dundee.lug.org.uk](http://www.dundee.lug.org.uk)  
**Contact** Duncan Gauld

**79 Sussex**

**URL** [www.phpworld.co.uk/~swlug](http://www.phpworld.co.uk/~swlug)  
**Contact** Gareth Ablett

**80 Wigan & St Helens**

**Contact** Paul F Johnson  
**Email** [paulf.johnson@ukonline.co.uk](mailto:paulf.johnson@ukonline.co.uk)

**81 Brixton**

**URL** [www.communitytechnology.org.uk/~linuxhome](http://www.communitytechnology.org.uk/~linuxhome)  
**Contact** R M Sanchez

**82 St.Andrews, Fife**

**URL** [www.standrews.lug.org.uk](http://www.standrews.lug.org.uk)  
**Email** [stuart@nx14.com](mailto:stuart@nx14.com)

**83 Nuneaton**

**URL** [www.nuneaton.lug.org.uk](http://www.nuneaton.lug.org.uk)  
**Contact** S Prosser

**84 Isle of Man**

**URL** [www.iom.lug.org.uk](http://www.iom.lug.org.uk)  
**Email** [helix@manx.net](mailto:helix@manx.net)

**85 Aylesbury**

**URL** [www.aylesbury.lug.org.uk](http://www.aylesbury.lug.org.uk)  
**Email** [drbond@educational-computing.co.uk](mailto:drbond@educational-computing.co.uk)

**86 Lancashire**

**URL** [www.lancasterlug.org.uk](http://www.lancasterlug.org.uk)  
**Contact** Jonathan Hill

**87 North London – see 47**

**Contact** Jason Hodson  
**Email** [jason@voyagercomputers.co.uk](mailto:jason@voyagercomputers.co.uk)



# LinuxUserGroups

## LUG OF THE MONTH

### South Cheshire

South Cheshire LUG actually covers the whole of the county (and has members in Staffordshire and North Wales). The modest title comes from the founders' belief that the larger towns of North Cheshire would soon form their own LUGs (we would still like to encourage this).

Three years down the line from the first idea for the LUG, and we are emerging from a quiet period. Meetings have restarted: we've just

had a terrific talk by Anthony Holloway from Jodrell Bank, speaking about their use of Linux, and their 180-CPU Beowulf cluster. This drew a good turnout, and was well received. Now plans are under way for a recruitment drive, and to find a permanent meeting place, where we can demonstrate Free Software, as an alternative to the current tour of real ale pubs.

We hope to collaborate with neighbouring LUGs on regional

events in the near future.

Members range from business users and I.T. professionals, through home users, to those working to bring Free Software to local schools and charities. All are welcome, regardless of knowledge, age or experience. Get involved and make a difference in your local community. The mailing list is linked from:

[www.sc.lug.org.uk](http://www.sc.lug.org.uk)



## Worldwide Linux User Groups

Free Software users across the globe

### Africa

#### EGYPT

URL [www.linux-egypt.org](http://www.linux-egypt.org)

Contact Hesham Bahram

#### GAUTENG, SOUTH AFRICA

URL [www.glug.org.za](http://www.glug.org.za)

Email [glugmin@revolution.org.za](mailto:glugmin@revolution.org.za)

### Australia

#### ADELAIDE

URL [www.linuxsa.org.au](http://www.linuxsa.org.au)

Email [mtippet@anu.edu.au](mailto:mtippet@anu.edu.au)

#### ALICE SPRINGS

URL [www.aslug.org.au](http://www.aslug.org.au)

#### MELBOURNE, VICTORIA

URL [www.luv.asn.au](http://www.luv.asn.au)

Contact [luv-committee@luvasn.au](mailto:luv-committee@luvasn.au)

#### PERTH

URL <http://plug.linux.org.au/>

#### SYDNEY

URL [www.slug.org.au](http://www.slug.org.au)

### Europe

#### AUVERGNE

URL [www.linux-arverne.org](http://www.linux-arverne.org)

Email [Cyril.Hansen@wanadoo.fr](mailto:Cyril.Hansen@wanadoo.fr)

#### COSTA DEL SOL (English speaking)

URL [www.fuengirola.lug.org.uk](http://www.fuengirola.lug.org.uk)

#### DENMARK

[Aissund](mailto:Aissund) [www.aislug.dk](http://www.aislug.dk)

[Esbjerg](mailto:Esbjerg) [www.eslug.dk](http://www.eslug.dk)

[Fyns](mailto:Fyns) [www.flug.dk](http://www.flug.dk)

[Midt-og Vestjylland](http://www.mvjlug.dk) [www.mvjlug.dk](http://www.mvjlug.dk)

[Nordjylland](http://www.njlug.dk) [www.njlug.dk](http://www.njlug.dk)

[Skåne Sjælland](http://www.sslug.dk) [www.sslug.dk](http://www.sslug.dk)

[Trekantsområdet](http://www.tlug.dk) [www.tlug.dk](http://www.tlug.dk)

[Vest-fyn](http://www.haarby-net.dk/vflug) [www.haarby-net.dk/vflug](http://www.haarby-net.dk/vflug)

[Århus](http://www.aalug.dk) [www.aalug.dk](http://www.aalug.dk)

#### EIRE

URL [www.linux.ie](http://www.linux.ie)

Email [root@linux.ie](mailto:root@linux.ie)

URL [www.dilu.org](http://www.dilu.org)

Email [glossary@dilu.org](mailto:glossary@dilu.org)

#### GOTHENBURG

<http://nain.oso.chalmers.se/LUG/>

### India

URL [www.linux-india.org](http://www.linux-india.org)

Email [newsmaster@linux-india.org](mailto:newsmaster@linux-india.org)

#### TRIVANDRUM

URL [www.river-valley.com/tux](http://www.river-valley.com/tux)

Email [anil@river-valley.com](mailto:anil@river-valley.com)

### North America

#### ALASKA

URL [www.aklug.org](http://www.aklug.org)

Email [deem@wdm.com](mailto:deem@wdm.com)

#### ALBERTA

URL <http://calgary.linux.ca/>

#### BATON ROUGE

URL [www.brlug.net](http://www.brlug.net)

Email [dpuryear@usa.net](mailto:dpuryear@usa.net)

#### BAY AREA

URL [www.balug.org](http://www.balug.org)

Email [aflyde@balug.org](mailto:aflyde@balug.org)

#### CLARKSVILLE, TN

URL [www.clug.org](http://www.clug.org)

Email [tux@clug.org](mailto:tux@clug.org)

#### DENVER

URL <http://clue.denver.co.us/>

#### FLORIDA

URL [www.flux.org](http://www.flux.org)

#### LOS ANGELES

URL [www.lalugs.org](http://www.lalugs.org)

Email [dank@alumni.caltech.edu](mailto:dank@alumni.caltech.edu)

#### NORTH COLORADO

Email [nclug@nclug.org](mailto:nclug@nclug.org)

#### OTTAWA CANADA

URL [www.oclug.on.ca](http://www.oclug.on.ca)

#### TAMPA

URL [www.suncoastlug.org](http://www.suncoastlug.org)

Email [president@suncoastlug.org](mailto:president@suncoastlug.org)

#### UHACC Normal, IL

URL [www.uhacc.org](http://www.uhacc.org)

Email [lug@uhacc.org](mailto:lug@uhacc.org)

#### VIRGINIA TECH

URL [www.vtluug.org](http://www.vtluug.org)

Email [nega@vt.edu](mailto:nega@vt.edu)

### South America

#### BUENOS AIRES

Email [dcoletti@impost.com.ar](mailto:dcoletti@impost.com.ar)

#### CHILE

URL [www.linux-chile.org](http://www.linux-chile.org)

#### MONTEVIDEO

URL [www.linux.org.uy](http://www.linux.org.uy)

#### PARAGUY/ ASUNCION

Email [rolgiati@conexion.com.py](mailto:rolgiati@conexion.com.py)

#### SAO PAULO

URL <http://gul.ime.usp.br/>

Email [gul@ime.usp.br](mailto:gul@ime.usp.br)

## Spreading the word

In the latest in this series on advocating GNU/Linux, **Jono Bacon** looks at marshalling your arguments.

So far in our little series of advocacy tips and tricks, we have looked at the aim of advocacy. We will now delve into the issues surrounding writing advocacy documents for the punters. Writing your advocacy ideas, and proposals and how you do it, is highly dependent on the people you are targeting. Home users, for example, will want a quick run down of the issues in a visually attractive manner. Business users will want a more formal report style document with emphasis on Total Cost of Ownership (we will discuss this next month). It is always important to format the document in the style of the audience, otherwise it may well be ignored.

Another important aspect to writing your documents is backing them up with research and evidence. We will be looking into this in more detail in future issues, but it is this evidence that will strengthen your claim. The important thing about writing your

proposal is that it must clearly identify how it can solve a problem (a) more efficiently, and (b) at a lower cost to the owner. This principle can be applied to both home and business users, but remember they will have radically differing needs.

Validity of information is the key in writing proposals. It is a wise move to quote other sources (remember to use full referencing), but always bear in mind who you are referencing and which documents you are suggesting the user reads. An example is that a white paper from RedHat is going to stand much more ground than a view posted on a discussion site such as Slashdot. Although Linux is community led, remember that the business world in particular favours the views and output of other businesses.

Next month we will take a look at the infamous Total Cost of Ownership issue that is pertinent to business and how we can build a case that Linux is cheaper to maintain. Have fun!

## Linux User Group organisers

If you're not listed here, or we have your details wrong, please contact us at: **LUGS!, Linux Format, 30 Monmouth Street, Bath, BA1 2BW** or email your details to: [linuxformat@futurenet.co.uk](mailto:linuxformat@futurenet.co.uk)

**EDITORIAL****Editor** Nick Veitch [nick.veitch@futurenet.co.uk](mailto:nick.veitch@futurenet.co.uk)**Art Editor** Julian Jefferson [julian.jefferson@futurenet.co.uk](mailto:julian.jefferson@futurenet.co.uk)**Production Editor** Richard Smedley  
[richard.smedley@futurenet.co.uk](mailto:richard.smedley@futurenet.co.uk)**Editorial Contributors**

Andrew Channelle, Charles Stross, Richard Drummond, Chris Brown, Paul Hudson, Robert Smith, Jonathan Wright, Marco Fioretti, Jono Bacon, Mike Saunders, David Coulson, Neil Lucock, Warren Brown, Nick Heron, Alain Buret. And not Paul Dunne

**COVER CD PRODUCTION****CD Editor** Neil Bothwick**ART CONTRIBUTORS****Photography** Louise Parker, James Wilson, Superstock, Getty Images**Illustration** Paul Bateman, Chris Winn, Shane Collinge**ADVERTISING SALES****Portfolio advertising manager** Matt Stanley 01225 442244  
[matt.stanley@futurenet.co.uk](mailto:matt.stanley@futurenet.co.uk)**Portfolio deputy ad manager** Diane Clydesdale**Business development manager** Damian Hughes**Account managers** Zoe Drew, Giles Crosthwaite Scott,  
**Ad sales executives** Jo Doyle, Stephen Hall**MARKETING AND PROMOTIONS****Marketing Manager** Fiona Tully**Subscriptions Executive** Marie Spicer**PRODUCTION****Production Co-ordinator** Diane Ross**Group Production Manager** Clare Tovey**Advertising Production Co-ordinator** Jo Crosby**MANAGEMENT****Publisher** Dave Taylor**Group Advertising Manager** Simon Moss**Group Publisher** John Weir**Circulation Director** Sue Hartley**Managing Director** Colin Morrison**DISTRIBUTION AND CIRCULATION****Circulation Manager** Jamie Malley[jamie.malley@futurenet.co.uk](mailto:jamie.malley@futurenet.co.uk)**Distributed by** Seymour Distribution, 86 Newman Street,  
London W1T 3EX**Tel** +44 (0)207 3968000**Overseas Distribution by** Future Publishing Ltd.**Tel** 01225 442244**Overseas Licences**Simon Wear [simon.wear@futurenet.co.uk](mailto:simon.wear@futurenet.co.uk)**Tel** +44(0)1225 442244 **Fax** +44 (0)1225 732361**Contact Details**

Linux Format, 30 Monmouth Street, Bath BA1 2BW

**Tel** +44 (0)1225 442244 **Email** [linuxformat@futurenet.co.uk](mailto:linuxformat@futurenet.co.uk)**Subscriptions and Mail Order****Phone** +44 (0)1458 271178. See page 98**Email** [linuxformat.subs@futurenet.co.uk](mailto:linuxformat.subs@futurenet.co.uk)

**Copyright** No part of this publication may be reproduced without written permission from our publisher. We assume all letters sent – by email, fax or post – are for publication unless otherwise stated, and reserve the right to edit contributions. All contributions to Linux Format are submitted and accepted on the basis of non-exclusive worldwide licence to publish or license others to do so unless otherwise agreed in advance in writing recognises all copyrights and trademarks. Where possible, we have acknowledged the copyright holder. Contact us if we haven't credited your copyright and we will always correct any oversight. **All CD-ROM** demos and reader submissions are supplied to us on the assumption they can be incorporated into a future covermounted CD-ROM, unless expressly stated to the contrary. We cannot be held responsible for mistakes or misprints. Linux Format recognises all copyrights in this issue. Where possible we have acknowledged the copyright holder. Please contact us if we have failed to credit copyright.

**Disclaimer** All tips in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your computer, peripherals or software through the use of any tips or advice.

**Printed** in the UK by Midway Clark (Holt) and  
© Future Publishing Ltd 2003

**LINUX** is a trademark of Linus Torvalds, GNU/Linux is abbreviated to Linux throughout for brevity.

All other trademarks are the property of their respective owners

**Future Publishing Ltd. is part of The Future Network plc.**

The Future Network produces carefully targeted specialist magazines and websites for groups of people who share a passion. We aim to satisfy their passion by creating titles that offer superb value for money, trustworthy information, multiple ways to save time and money, and are a pleasure to read or visit. Today we publish more than 80 magazines and over 30 magazine websites and networks from offices in five countries. The company also licenses 32 of its titles resulting in over 60 local editions in a further 23 countries.

The Future Network plc is a public company quoted on the London Stock Exchange (symbol: FNET).

**Non-executive Chairman:** Roger Parry**Chief Executive:** Greg Ingham**Chief Operating Officer & Managing Director, UK:** Colin Morrison**Group Finance Director:** John Bowman**Tel** +44 (0)1225 442244

BATH • LONDON • MILAN •  
NEW YORK • PARIS •  
SAN FRANCISCO • WROCLAW

[www.thefuturenetwork.plc.uk](http://www.thefuturenetwork.plc.uk)

Media with passion

# NEXT MONTH

## Issue 37 on sale Friday 31 January

# READY FOR THE DESKTOP?

The debate continues, but we look at what would constitute 'ready', how it varies depending on your perspective and how many are already using Linux as a desktop alternative. Plus we'll garner opinions from users and industry players on the future of Linux on your desktop

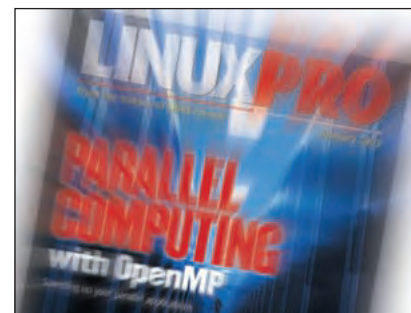
## ON TEST

Jool's "Kwartz" server, Arkeia 5 (we mean it this time) and the first outing for United Linux. And lots more

## LINUXPRO

### New size, more content

From Next Month, Linux Pro has got even bigger. Two extra pages might not seem a lot, but the whole mag will be supersized to A4 – twice the real estate of the current version. Check out the new Linux pro, with even more features, interviews and information for Linux professionals



## DON'T MISS YOUR COPY OF LINUX FORMAT!

Get it delivered to your door every month – subscribe on page 98

The exact contents of future issues are subject to change