

THE MAGAZINE FOR ADVANCED LINUX KNOW-HOW

# LINUX

## MAGAZINE

OCTOBER 2002

Apache on Linux

# WINNING TEAM

**LOWER  
PRICE**

Mod\_rewrite & mod\_gzip  
Modules overview  
Migration to 2.0

News & Report

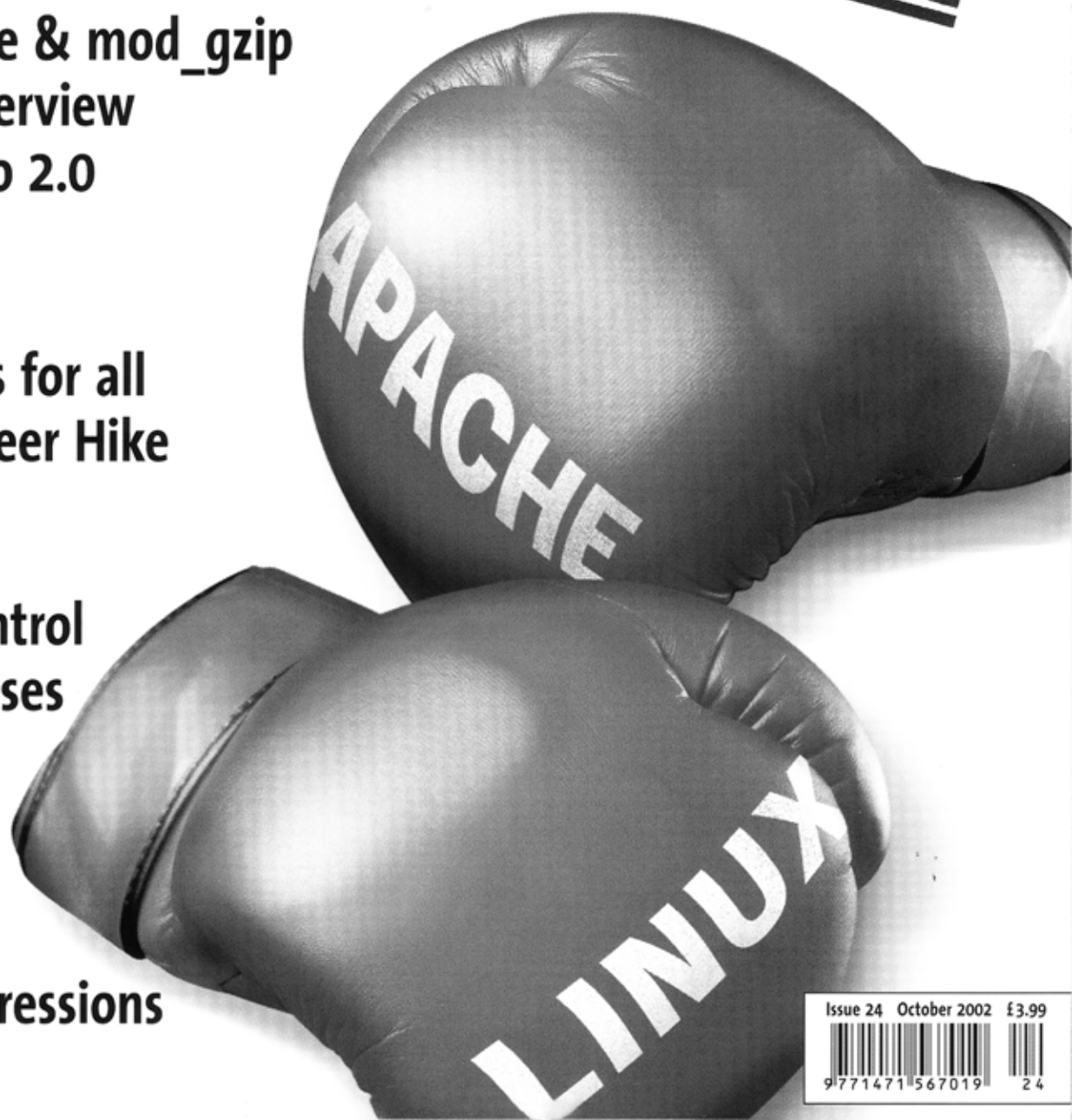
Mail Servers for all  
The Linux Beer Hike

Hands-on

PhotoPC control  
MP3 databases  
LeoCAD

Revealed

OpenSSH  
Regular Expressions



Issue 24 October 2002 £3.99





Figure 2: Gnumed summary screen

dangers of proprietary software in the health sector took place in March 2000. Physicians refused to base their decisions on non-transparent algorithms. Within this discussion Horst Herb was accused of unconstructive criticism, which he took as a trigger to start working on Gnumed.

After a first working alpha-release was presented at the MedInfo2001 in London, the international interest in Gnumed made an total redesign of the internal structure necessary. Implementing this new structure is currently the main task for the project co-ordinations Horst Herb and Karsten Hilbert, who work on this together with about 17 other developers and many volunteers.

After completing a minimal version, which they hope to already be useful, it is planned to make Gnumed a complete medical solution which should include decision support.

The problems that Gnumed faces on the way to this is a lack of free pharmaceutical databases, different health systems with different regulations, lack of data formats, transfer standards and standardised messaging protocols, as well as lack of a system to create a globally unique ID for a patient.

Programming languages used in this project are Python and C/C++ on the client side, PgSql, C and Python on the server side, with reliability and security being the most important paradigms; both of which are not adequately treated in proprietary solutions in the opinion of

the Gnumed team. In the Gnumed team there are many physicians from many different fields, who know what they want, but often not how to implement it.

Therefore some more experienced developers would be a very welcome addition to the Gnumed team.

Gnumed, which seeks to have an easy, ergonomic and highly configurable GUI, support for

different languages and health systems, as well as relative platform independence in the end, is published as Free Software under the GNU General Public License.

If you wish to get active in this sector, Gnumed is surely a project to help with.

## OIO

The "Open Infrastructure for Outcomes" (OIO) [9] is called the "Search for the holy grail" of data portability by its author, Andrew Ho. Nandlal Gunaratne, Alexander Chelcnokov and others accompany him on this quest.

OIO was used for production at the Harbor-UCLA Medical Center in March 2001 before being published as Free Software under the GNU General Public License in August. By September it was managing data of more than 1000 patients and since February 2002, it is being used as a hospital-wide information system. So it is safe to say that OIO has proven itself already in daily use.

The primary components of OIO include the server, which is accessed via any browser through HTML, and the OIO library. The server is a flexible, web-based data management system, which manages users, patients and

information about them; although it would of course also be possible to use it for invoices, deliveries or accounts.

The OIO-library is a metadata repository, which allows exchanging metadata like plug-and-play web forms or project descriptions between server and client.

An OIO user can create or modify forms through a web browser, which is then immediately available to be used for data collection over the web.

Later forms can be exported as XML-data to be transferred into a metadata repository like the OIO library or uploaded to another OIO server.

Of course it is also possible to assemble data from different forms into a single dataset that can then be searched/queried over the web with help of logical operations.

Although OIO has been used for some time, development is not complete. Among the planned features for future releases is support for wireless PDAs. Plug-and-play protocols will also be supported. Most helpful at the moment would be more users, more feedback and better packaging.

At least with the last point Debian-Med should be able to help.

## Res Medicinae

Res Medicinae [10] by Christian Heller is also used by the Debian-Med project. Together with Karsten Hilbert he works on making Res Medicinae an extensive software solution in the medical area.

To achieve maximum portability, Res Medicinae is based on Java (API/Swing, Servlets/JSP, JDBC) with some

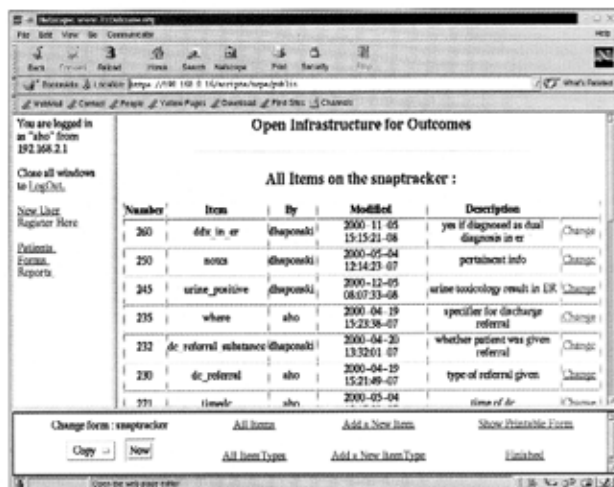


Figure 3: Viewing a complete form

CORBA/IDL and SOAP/XML. This already shows the largest problem of this Free Software project under the GNU General Public License and GNU Free Documentation License, because lacking a full-featured Free Software Java implementation, the freedom of the project is in danger.

But freedom was a major motivational factor for Christian to begin working on Res Medicinae. He wants to overcome the very expensive and proprietary scene of medical information systems and give users in less privileged countries access to a free, stable, secure, platform independent and extensive system.

The project is still rather young. According to the plans, at the end of 2002, the ResMedLib framework should be consolidated and prototypes for two complete modules should be available. In 2003, the administrative module, printing forms and generating reports should work.

Afterwards, an image processing and a management tool as well as a billing and statistical module should be added. A training module as well as a decision support module will then finish the whole project.

So you should probably not try to use the project in your daily life, but those who are interested to bring medical competence, language translations, Java programming or webpage design into the project, will receive a warm welcome in Res Medicinae.

As far as the authors know, Res Medicinae is currently the only Java based GPL project in the medical area and they plan to work together with OpenEMed, a similar Java project under



Figure 3: JavaFish preparing to battle

a BSD license and the already mentioned Gnumed project to achieve full project interoperability.

That should be enough health for today, if there are other projects in this area, I would like to present them in a later issue. An email [1] would be the appropriate way to get this going.

## Romance

Romance [11] is the attempt by Bertrand Lamy and Jean-Baptiste Lamy, to give Free Software a real, Free alternative to Microsoft's .NET.

According to Bertrand, their motivation is that Ximian will not be able to deliver a Free implementation of .NET. That Microsoft has already promised to fight all Free alternatives using software patents does indeed make this a plausible scenario.

Also standards controlled by those companies without any Free reference implementation always have the advantage that the company is several steps ahead, while the Free projects have many prob-

lems following. The situation around Java suffers from this effect.

The answer is clear: We need a Free standard with a Free implementation. This is what Romance seeks to provide.

The first part - and beginning of development - is Rose, the "Romance Object System rosE." Rose provides a protocol, which allows for the sharing of objects between the different programming languages.

The next step of development will be Wise, the "Romance Widget Server." It will be available as a GUI/toolkit library to all Romance applications through the Romance server.

The paradigm employed in Wise is that all widgets remain the property of the Wise process, and not of the different applications. That should allow Romance to make sharing of widgets very fast and simple.

Since Bertrand and Jean-Baptiste believe that 75% of all desktop applications should be written in script languages, they have concentrated on supporting Python, Guile and C first. According to their plans, Rose will also support Perl, Ruby, Lisp, Scheme and other dynamic languages in the future.

There are many examples how Romance can be used.

For large applications, it is often a good idea to define an expansion language. Instead of choosing one

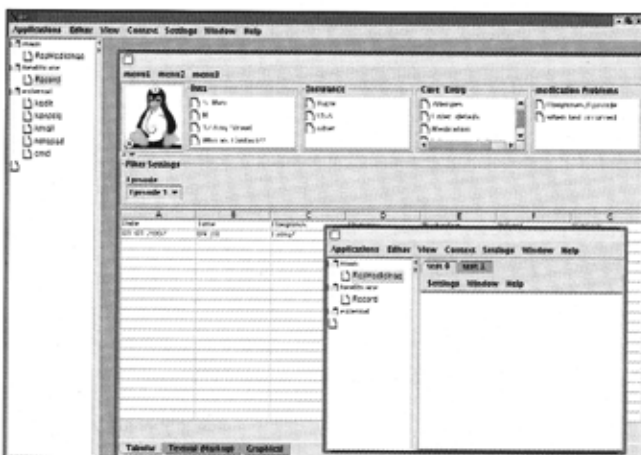


Figure 4: Res Medicinae forms aid open source doctors