

Gnomogram STOKING THE FLAMES

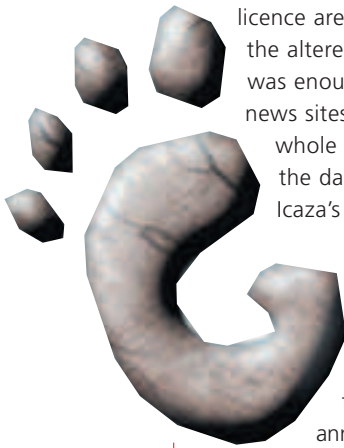
This month

Gnomogram takes a look at GARNOME, Guikachu, Genigma and GNOME's supposed links with .NET

GNOME and .NET

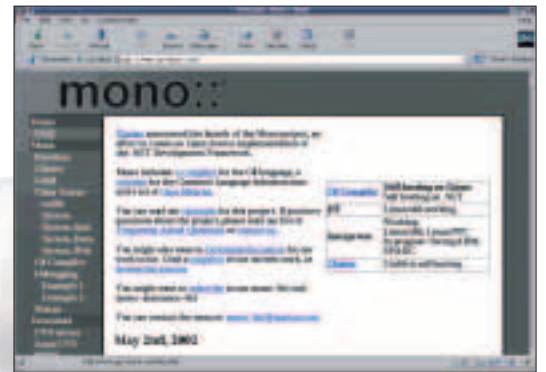
More than just about any other project in the domain of Free software, GNOME finds itself again and again in the middle of unpleasant flame wars. The latest episode in this sorry tale is an interview between *The Register* and Miguel de Icaza with the somewhat misleading title "Gnome to be based on .NET". In this interview Icaza speaks out in positive terms about Microsoft's .NET framework, which he is porting to Linux under the name of Mono. This was in addition to the fact that a few days before, the licence for the Mono class library was changed from the GPL to an X11-type licence, which enables companies such as Intel to contribute code. Even if this licence is free in the sense of the FSF, programs under the X11 licence are distributed in binary form, without the altered source code being released. This was enough to convince the readers of many news sites that Icaza – and thus too the whole GNOME team – had succumbed to the dark side. Accordingly, demands for Icaza's resignation began to mount.

Even if Icaza does play an important role in the GNOME project, this would not put him in any position to take such a fundamental design decision. Within the GNOME Foundation there is an annually re-elected board, with ten people in addition to him who have to reach agreement on such decisions. It is also doubtful whether the GNOME community would go along with such a fundamental change from C to C#. So Mono remains the early version of a C# implementation with plans for GNOME language bindings, like those which already exist for Python or C++. The reports from *The*



Libraries required

Guikachu: gnomemm, libxml1, libglade, gdk-pixbuf, xsltproc



dotNET for Linux

Register that Richard Stallman had spoken out negatively on Icaza's plans also turned out to be a hoax, as Stallman himself put right.

GARNOME

GNOME is known for its many dependencies, and for newbies especially it can be nerve-wracking to hunt down and compile all the necessary packages. Although, with the vicious build scripts there is an option for loading a current GNOME automatically from the CVS – anyone who would rather work with the archives from the official releases, has until now been on their own. With GARNOME it is now possible, in a similar way to BSD's port system, to load all archives automatically with their dependencies and to compile them in the correct sequence. Once GARNOME is unpacked, one simply needs to change to the directory *gnome/meta-gnome-desktop* and enter *make* – the rest is done by GAR. The system is configured via the file *gar.conf.mk*, in which, under "BUILD_PREFIX", one can also specify the installation directory. GNOME 2 should in any case be installed separately from GNOME 1.x, since otherwise there can be conflicts – this is why all compiled programs are installed by default to *~/lgarnome*. (The tilde stands for the home directory.)

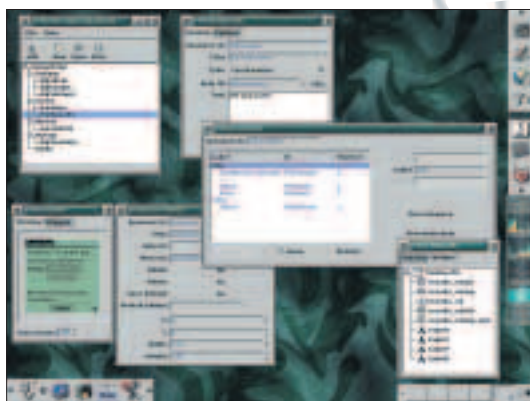
Guikachu

With the aid of Guikachu, just as with Glade, so-called resource files can be created for Palm OS. As

such, it is possible to position various widgets for a graphical user interface with ease, to create menus and dialogs and to save the whole thing as an XML file. Guikachu is not, however, able to compile the created file itself – instead the interface has to be exported, using *FileExport RPC* or with the program *guikachu2rcp* into a format readable for Pirc. *Guikachu2rcp* is in fact only a simple bash script, which converts XML files into a different format with the aid of XSLT – the actual transformation is left to Xsltproc. To actually integrate the interface into a program, however, yet more programs are needed: in the Prc-Tools there are a range of programs which help to produce and to debug code for the Dragonball processor used in Palm. You will also need a Palm OS SDK, which comes with the necessary Includes and can be found at <http://www.palmos.com>. In the SDK archive offered by Palm, in addition to the documentation there is also an rpm package, which can easily be converted under Debian with *alien*. Debian users also have to create a symbolic link, with:

```
ln -s /opt/palmdev/sdk-4/ /usr/share/prc-2
tools/sdk-4
```

Other distributions can search at */usr/local/palmdev/* for the SDK – the link then has to be placed in this directory.



A Palm GUI is quickly created with Guikachu

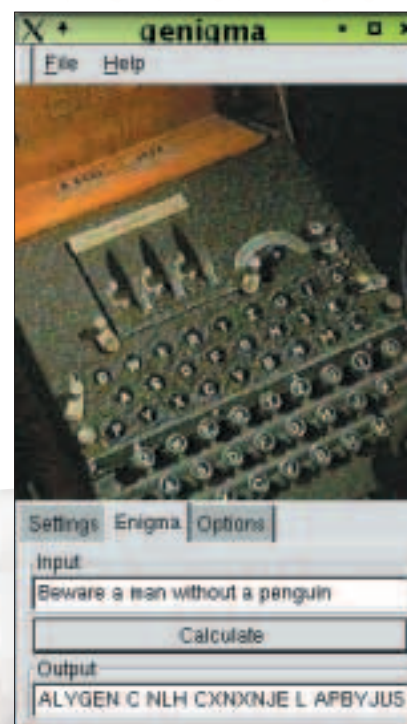
Genigma

Genigma is an emulator of the German Enigma machine – probably the most high-profile encryption machine of all time – which has been used and abused in numerous books and films. Even though the fundamental crypto-analysis of the Enigma was performed by a Polish team, it was only in Bletchley Park Station X that the so-called “Bombe” was developed, which by checking standard phrases (so-called cribs) was able to seek possible keys very quickly.

All types of Enigma share the same basic principle of the “rotors”, the number of which varies depending on the model. Genigma emulates model

M3, in which three of five possible rotors were used. This model was used by the Luftwaffe and the Wehrmacht – the models used in ships and U-boats were more complex still. Each rotor has 26 inputs or outputs, which were wired up in a specific fashion. So a letter typed on the typewriter-like keyboard was replaced several times by a different letter. Crypto-analysis is also made more difficult by a plugboard, with which certain letters are exchanged. The rotors were also turned after each keystroke, which changed the encryption for each letter.

To actually encrypt a text in Genigma, one thus has to enter under roller 1-3 respectively a number between one and five, which stands for a rotor – each rotor is of course included only once. The ring setting can also be changed. On the ring, similar to the contacts on the roller, there were also the letters from A to Z. So setting B equates to a displacement of the ring by one position. In exactly the same way, the roller can be turned even before starting the encryption – hence the setting “Start”. Letter pairs can be specified in the plug field, in order to veil the result even further. Naturally here, too, each letter can only be used once. In addition to the GNOME interface Genigma can also be used from the command line – more precise information on the options can be found in the corresponding manpage.



Genigma encrypting a historic phrase

Info

The Register interview with Miguel de Icaza
<http://www.theregister.co.uk/content/4/23919.html>
 Mono homepage <http://www.go-mono.com>
 The Register RMS clarification
<http://www.theregister.co.uk/content/4/23978.html>
 GNOME's vicious build scripts
<http://developer.gnome.org/dotplan/notes/vicious-build-scripts.html>
 GARNOME homepage
<http://www.gnome.org/~jdub/garnome/>
 Guikachu homepage
<http://cactus.rulez.org/projects/guikachu/>
 Palm OS games
<http://www.ardiri.com/index.cfm?redir=palm=pilrc>
 Prc-tools homepage
<http://prc-tools.sourceforge.net>
 Palm OS developers tools
<http://www.palmos.com/dev/tools/>
 Genigma homepage
<http://home.pages.at/kingleo/development/gnome/gnome-en.html>
 Enigma the movie <http://www.enigma-themovie.com>