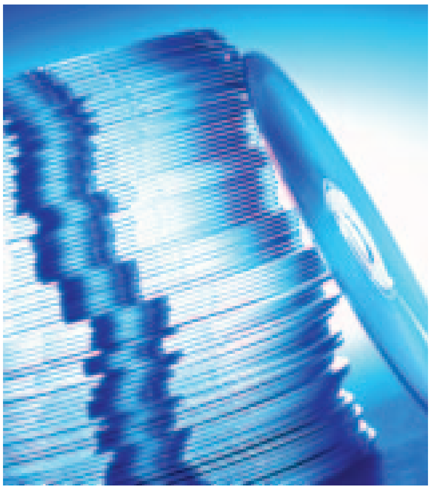
GWhere

A Break for the Disk-Jockey



The GWhere CD Indexer is just what the doctor ordered for those of you suffering from the "can't quite remember what Linux Magazine subscription CD the xyz tool was on" syndrome. BY PATRICIA JUNG

Twenty subscription CDs, a heap of MP3 disks, and a backup of your second computer – that's quite an impressive collection of CDs, but if you do not remember what disk the file xyz is stored on, you have a problem: Insert the CD, launch *find*, remove the CD if you draw a blank, and start again… You would not want to have to repeat all those steps, unless you are looking for a really important file.

"Where there's a will, there's a way", *Sébastien Lecacheur* thought and so he wrote *GWhere*, (*http://www.gwhere.org/*), a little GUI tool that indexes data CDs, floppies, or Zip disks. Allowing you to search the file tree without needing to sign up for a degree in diskjockeying. If you know what medium the required file is stored on, you merely need to locate that particular medium.

The current version of GWhere 0.0.25 is available as a binary for RPM based distributions, but you might prefer to compile the source code stored in *GWhere-0.0.25.tar.gz*. Version 1.2.0 or better of GTK, and the matching *gtk-dev(el)* package must be pre-installed to do so:

```
tar -xzvf GWhere-0.0.25.tar.gz
cd GWhere-0.0.25
./configure
make
checkinstall
```

If *checkinstall* [1] is not available, you can use *make install* instead. If you want to define the root directory for the installation yourself, you can call *configure* with the *--prefix=directory* flag.

First Things First...

Type *GWhere &* (paying attention to the case) in an X terminal session, and then wonder what to do about the more or less empty window that you are confronted with at this point. Neither the *Help* nor the *File* menu give you any clues as to what to do with the heap of CDs that you wanted to index.

Users who require menus in any language apart from English will need to set the *Locale* correspondingly:

```
export LANG=fr_CA; GWhere &
```

will use French Canadian, for example. If you want to define French Canadian as the target language for *GWhere*, but not for the shell, you can put this command in **parentheses**:

```
(export LANG=fr_CA; GWhere &)
```

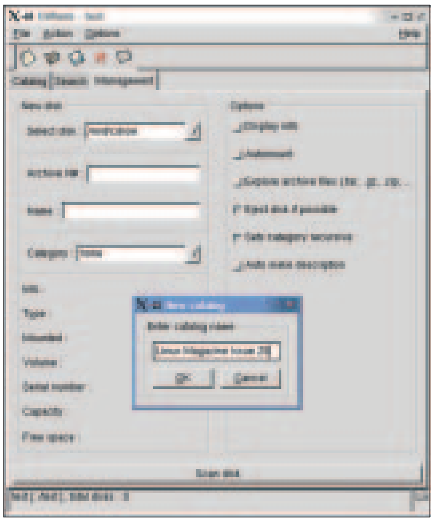
The main indexing function turns out to be accessible via a tab labelled *Management* (Figure 1). You can use the drop-down menu *Choose Volume* to locate the mount point for the CD, insert the CD, and then click on the *Browse Volume* button at the bottom of the window. If the medium is not mounted at this point, *GWhere* will return an error message. To avoid this, you can check the *Automount* checkbox under *Options* before clicking on the browse button.

GWhere prompts you for a *Catalog Name* at first. This name has nothing to do with the CD and is simply a general heading for all the CD indices stored in a single file.

As *GWhere* can only display and manipulate a single index file in the current version, this heading is not really important – just enter *mycds*, or *cdindex* or something similar.

However, you will want to put some thought into answering the next prompt, which refers to the *Volume Name* (Figure 1). This should ideally provoke a reaction such as "Oh yes, that's the CD with…" from the user. All that fancy search functionality is useless if you are unable to physically locate the CD that the *GWhere* search results refer to.

GWhere zooms through the indexing process in next to no time, and if you have additionally selected *Eject Volume if*

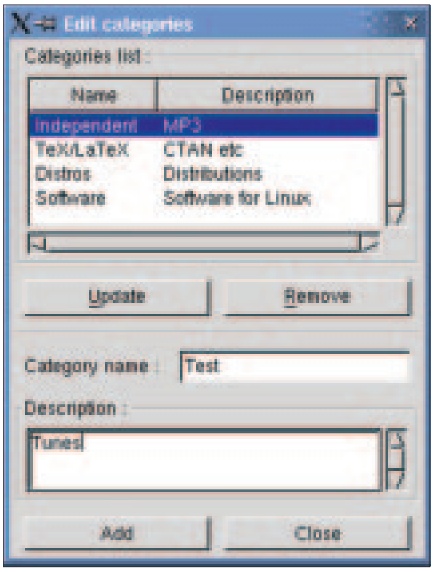


Linux Magazine Issue 28

Figure 1: The Volume Name is important to the success or failure of a search operation

OUT OF THE BOX

There are thousands of tools and utilities for Linux. "Out of the box" takes a pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.



Categories list :

Name	Description
Independent	MP3
TeX/LaTeX	CTAN etc
Distros	Distributions
Software	Software for Linux

Category name : Test

Description : Tunes

Figure 2: The dialog for adding keyword categories is not exactly intuitive

GLOSSARY

; (Semicolon): *A semicolon between two commands in the command line has the same effect as pressing [Enter]: After processing the first command, the second is processed. This allows you to enter a series of short commands in quick succession.*

Parentheses: *Commands in parentheses are not processed by the current shell, but call a new subshell (which is automatically closed after processing). This allows you to use environment variables to provide a command with its own environment. In our example, the value for LANG only affects the subshell in which GWhere was launched, but not the current working shell. So, in this case, only GWhere would be expected to speak French; other programs called in the current shell (and the shell itself) will continue to use the default setting for LANG.*

possible, will automagically open your CD-ROM tray after finishing the index.

Categorized and Described

If you have had more than your fair share of CD diskjockeying, you will probably want to save the index using *File/Save* in the main menu. While you are waiting, you might like to define some keywords for your CD collection under *Action / Edit Categories* (Figure 2). Type the keyword in *Category Name*, and add a few explantory comments in *Description*, before clicking on *Add* to add the entry to the *Category List*. Editing categories you entered previously can be more challenging.

To do so, you first select the list entry and then click on the *Update* button. This toggles the *Add* button in Figure 2 to *Update*. Click on this button when you have finished editing the keyword file.

Clicking on the *Catalog* tab not only reveals the current collection of indexed CDs, allowing you to navigate them, but you can also add metadata. Right click with the mouse to open a menu, allowing you to add a keyword and a description for every directory and file via the *Properties* dialog box (Figure 3).

The Joy (and Pain) of Search Ops

Don't forget the program's current state of development before you get too enthusiastic about entering metadata. Although the *Search* tab (Figure 4) theoretically allows you to search by description only, this feature does not seem to be available at present – neither are the functions for searching by media name (*Disk*), or by keyword (*Category*).

If you prefer not to search by full name, you can activate the *Regular Expression* feature. This allows *GWhere* to find any files containing the *png* string, such as *libpng.so.2* or *top-bg.png*.

If you are only interested in PNG images with the *png* or *PNG* suffix, you can use a dollar sign to indicate the end of the string and search for *png$*. If you also select *Upper/Lower Case*, *GWhere* will respect the case of your search string: In this case a capital letter will find only those files that have a capital letter at the appropriate position in their names.

The program has an annoying habit if a search op fails to find a result: it has no progress indicator, no status message and no way of knowing whether the search op is still in progress or has failed.

This kind of inconsistency in the user interface certainly keeps you on your toes, but in our opinion *GWhere* is indispensable for anyone just starting to lose track of their CD collection. It is a pity that *GWhere* can only maintain a single index file, as you really do need two catalog files to avoid mixing up your MP3 collection and your Linux Magazine CDs. But on the upside, you can always run multiple parallel *GWhere* processes if you require. ■

INFO

[1] Christian Perle: "Say Hello Wave Goodbye", Linux Magazine Issue 22, p78.

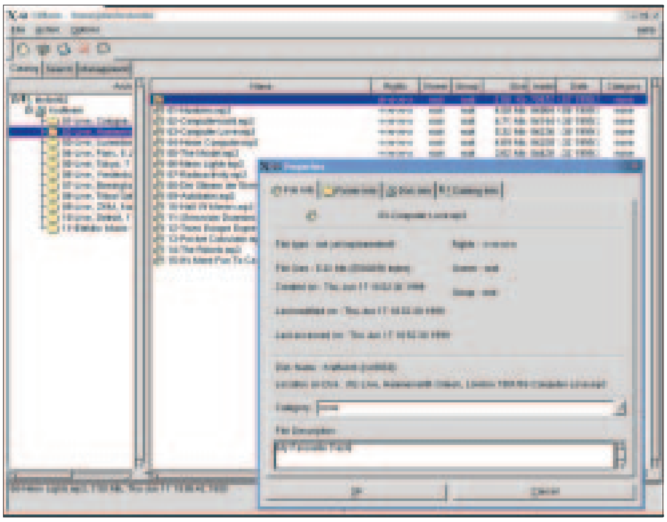


Figure 3: If required, you can add a description and a keyword to every file

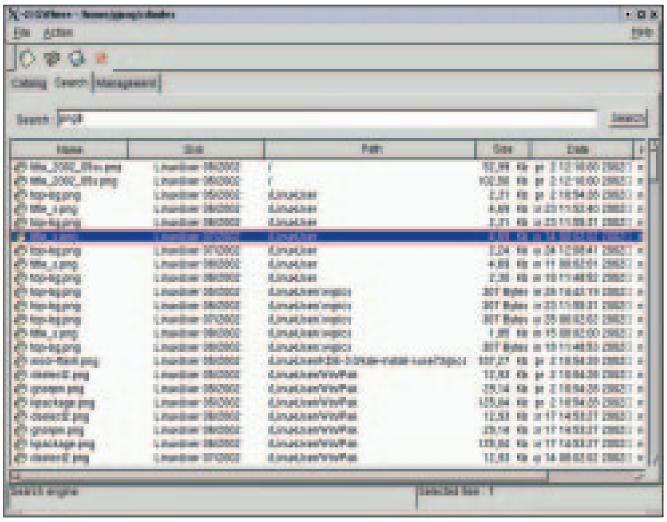


Figure 4: Searching for PNG files