QUANEKO

FIND THE STUFF ON YOUR LOCAL HARDDISC

User Manual

Thomas Jund <info@jund.ch>
Andrew Mustun <andrew@mustun.com>
Laurent Cohn <info@cohn.ch>

13th May 2004

Version 1.0

quaneko

User Manual

Authors: Thomas Jund, Laurent Cohn, Andrew Mustun

Release: May 2004

Contents

Chapter 1

License

Copyright

Document ID: $Date: 2004/05/13 15:39:23 $
Release: April 2004

Trademarks

Intel is a registered trademark and 80286, 80386, 286, 386, 486, Pentium and Pentium Pro are trademarks of Intel Corp.
Linux is a registered trademark of Linus Torvalds.
Microsoft, and MS are registered trademarks and Windows, Windows 95, Windows 98 and Windows NT are trademarks of Microsoft Corp.
Postscript is a registered trademark of Adobe Systems. Inc.
All other brand names, product names, or trademarks belong to their respective holders.

The GNU General Public License

Quaneko is released under the terms of the GNU General Public License (GPL), Version 2.

Chapter 2

Preface

Scope of This Manual

The intention of this manual is to provide a quick introduction from the user perspective. It does not cover any details of how quaneko internally works. If you are interested in those topics, please refer to the "Project Report and Technical Documentation" which is available from the quaneko homepage at: http://quaneko.sf.net.

The main topics of this document are:

- Using the graphical user interface (GUI)
- Using the command line interface (CLI)
- Configuring format filters

Chapter 3

Basics

This chapter is a brief introduction into quaneko. It should give you an overview over what quaneko is and what it can do for you.

What is quaneko?

quaneko is a tool that allows you to quickly search for keywords in the files and directories on the local hard disk. It creates indexes over the words in those files. For you, this means that a typical search query will only take seconds.
quaneko can not only search for words in plain text files but also in various other file formats, depending on its configuration (e.g. doc, pdf, html, xml, ..).

What is quaneko not?

quaneko focuses on searching local harddisks. It is not a web crawler or web search engine.

Is quaneko for me?

There are various reasons why you might consider using quaneko:

- You have a lot of personal and downloaded data on your hard drive and finding the right file has become difficult and time consuming.
- There are folders on your hard drive that contain a lot of files which don't have speaking names. For example files that are numbered like 'rfc1006.txt', 'rfc1234.txt', ..
- You need to be able to search the file contents of non-text documents such as Word doc, PDF or other formats.

Choice of User Interfaces

quaneko comes with two user interfaces: a graphical user interface (GUI) and a command line interface (CLI). Which one you want to use is mainly a matter of taste. However, if you need to automate the indexing or query process, the CLI is usually the better choice.

Chapter 4

Filters

Before you can start using quaneko, you will most likely want to add support for your favorite file formats (e.g. doc, html, pdf, ..). quaneko allows you to install and configure individual filters for file types you want to index. This section describes how to do this. If you want to index plain text files with the extension "txt" only, you can skip this section.

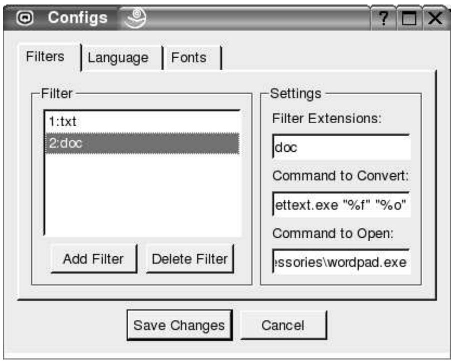
Adding Support For Various File Formats

Adding support for a new file format means configuring a new filter in quaneko. By default, quaneko only supports plain text files with the extension "txt". If you want to index other file types, you need to configure filters for them.
If you want to index a file format that is not listed here, you will also find a generic description about how to add support for any file format at the end of this section.

The example screenshot in figure 1 shows how to configure a utility called "gettext.exe" under Windows for parsing Word documents. The exact settings may vary for your system. In these settings we assume that you've installed "gettext.exe" in "C:\Program files\gettext\" and that the file "wordpad.exe" is located in "C:\Program files\Windows NT\Accessories":

Filter extensions:	doc
Parse Command:	C:\Program files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program files\Windows NT\Accessories\wordpad.exe

Once configured like described here, the Word doc filter should be ready to use.



1:txt
2:doc

Filter Extensions: doc

Command to Convert: ettext.exe "%f" "%o"

Command to Open: ›ssories\wordpad.exe

Figure 1: Configuration of the Word doc filter.

You can configure the other format filters in the same way. The following sections list some possible filter configuration settings for Windows and Linux.

Configuration Under Windows

For Windows there is one filter available which works for many common formats:

doc	Microsoft Word document format
xls	Microsoft Excel spreadsheet format
ppt	Microsoft PowerPoint presentation format
pdf	Adobe portable document format
html, htm	Hypertext Markup Language
txt	Plain text
rtf	Rich Text Format
wpd	Corel WordPerfect® document format
hlp	Microsoft Help format

The utility to convert all these formats into plain text is called "GetText" and is available from:
http://www.kryltech.com/freestf.htm.

Word

Filter extensions:	doc
Parse Command:	C:\Program files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program files\Windows NT\Accessories\wordpad.exe

Excel

Filter extensions:	xls
Parse Command:	C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program Files\Microsoft Office\Office10\excel.exe

PowerPoint

Filter extensions:	ppt
Parse Command:	C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program Files\Microsoft Office\Office10\powerpnt.exe

Adobe Portable Document Format (PDF)

Filter extensions:	pdf
Parse Command:	C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program Files\Adobe\Acrobat 5.0\Reader\AcroRd32.exe

Hypertext Markup Language (HTML)

Filter extensions:	html htm
Parse Command:	C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command:	C:\Program Files\Internet Explorer\iexplore.exe

Plain Text

Filter extensions: txt
Parse Command: C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command: C:\Program files\Windows NT\Accessories\wordpad.exe

Rich Text (RTF)

Filter extensions: rtf
Parse Command: C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command: C:\Program files\Windows NT\Accessories\wordpad.exe

Word Perfect®

Filter extensions: wpd
Parse Command: C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command:

Help Files

Filter extensions: hlp
Parse Command: C:\Program Files\gettext\gettext.exe "%f" "%o"
Open Command: winhlp32

Configuration Under Linux

There are numerous filters available to convert file formats into plain text. Some might already come with your favorite distribution, for others you might have to download the sources and compile them.

Word

Filter extensions:	doc
Parse Command:	antiword "%f"
Open Command:	OOo
Download Filter:	http://www.winfield.demon.nl/

Adobe Portable Document Format (PDF)

Filter extensions:	pdf
Parse Command:	pdftotext "%f" "%o"
Open Command:	acroread
Download Filter:	http://www.foolabs.com/xpdf/

MP3 Description (ID3 Tags)

Filter extensions:	mp3
Parse Command:	id3info "%f"
Open Command:	xmms
Download Filter:	http://id3lib.sourceforge.net/

Hypertext Markup Language (HTML)

Filter extensions:	htm html xml
Parse Command:	html2text "%f"
Open Command:	konqueror
Download Filter:	http://www.linux.org/apps/AppId_7912.html

Adding Support for Other File Formats

If you want to index file types that are not mentioned in the previous sections, you need to configure your own filters for them. The following steps are required to configure a new filter:

1. Download an appropriate converter application. The utility must be able to produce a plain text file from a file in an other format. Further, it should neither show a GUI nor require any user interaction.
2. Install the application on your system.
3. Configure the converter as a filter in quaneko.
4. After this procedure, the filter is ready to be used with quaneko.

What Is A Filter?

A filter configuration for one filter consists of:

- A list of file types this filter supports (e.g. "htm html").
- A string which specifies the application call for converting those types into plain text (e.g. `html2text "%f" "%o"`). We refer to this string as 'Filter Conversion String'.
- Optionally the name of the application which can be used to open that document type (e.g. "mozilla")

Filter Conversion Strings

The filter command to parse a file and convert it into plain text can be configured as a string which contains **%f** for the data file that is handed from quaneko to the converter and **%o** for the output file.
Example: The string

```
pdf2text "%f" "%o"
```

is converted at runtime to:

```
pdf2text "/home/tux/file.pdf" "/home/tux/.qnk_tmp.txt"
```

If **%o** is omitted, quaneko assumes that the filter streams the plain text to standard output (internally it adds ">%o" to the command).

It's usually recommendable to add quotation marks around **%f** and **%o**. Otherwise you will experience problems with spaces in file names.

Manual Configuration of Filters

This section shows how to add and edit filters with a plain text editor. This can be handy if you cannot run the GUI of quaneko for example when working on a server.

Location of the Configuration File

First of all, you need to locate your configuration file and open it with a text editor. The configuration file is called '.quanekorc' and located in the user's home directory.

Example of the file location:

```
/home/username/.quanekorc
```

Most common place (depending on your Windows version, configuration and language):

```
C:\Documents and Settings\Tux\.quanekorc
```

Adding a Filter

The filters in the configuration file are numbered with a unique ID. The first ID has to be 1 and no ID can be left out.

Example configuration for filter 1, which can write directly into an output file (preferred notation):

```
/filters/filter1_app pdftotext "%f" "%o"
/filters/filter1_open acroread
/filters/filter1_type pdf
```

Example configuration for filter 2 which writes to standard output:

```
/filters/filter2_app antiword "%f"
/filters/filter2_open word.exe
/filters/filter2_type doc
```

Example configuration for filter 3 which deals with more than one extension:

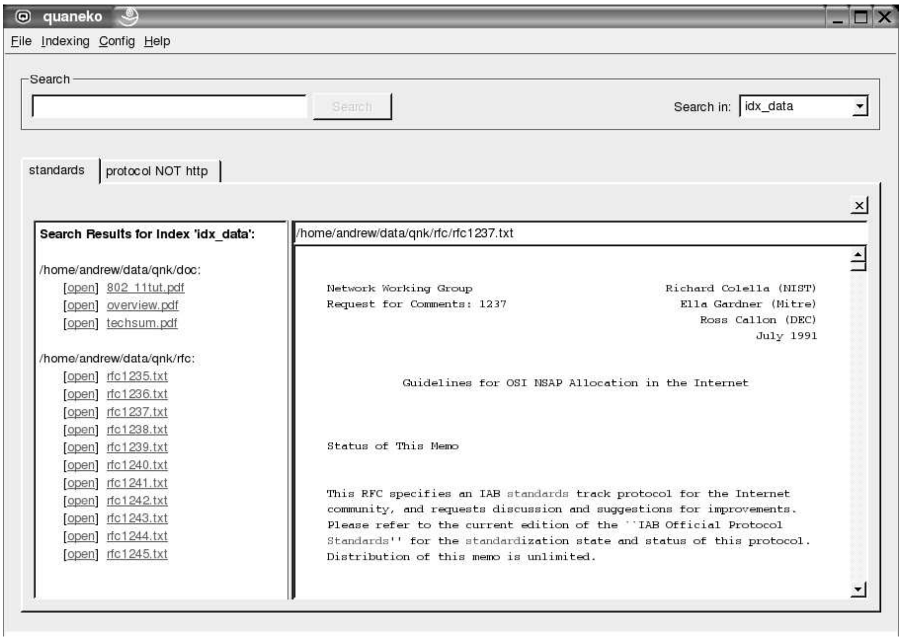
```
/filters/filter3_app htmltotext "%f"
/filters/filter3_open mozilla
/filters/filter3_type htm html xml
```

Chapter 5

Graphical User Interface (GUI)

This section describes the graphical user interface of quaneko. The exact look and feel might be slightly different on your platform than it is shown in the screenshots.

The Main Application Screen



Search Results for Index 'idx_data':

/home/andrew/data/qnk/doc:
[open] 802_11tut.pdf
[open] overview.pdf
[open] techsum.pdf

/home/andrew/data/qnk/rfc:
[open] rfc1235.txt
[open] rfc1236.txt
[open] rfc1237.txt
[open] rfc1238.txt
[open] rfc1239.txt
[open] rfc1240.txt
[open] rfc1241.txt
[open] rfc1242.txt
[open] rfc1243.txt
[open] rfc1244.txt
[open] rfc1245.txt

/home/andrew/data/qnk/rfc/rfc1237.txt

```
Network Working Group                                  Richard Colella (NIST)
Request for Comments: 1237                               Ella Gardner (Mitre)
                                                            Ross Callon (DEC)
                                                                    July 1991


             Guidelines for OSI NSAP Allocation in the Internet


Status of This Memo


This RFC specifies an IAB standards track protocol for the Internet
community, and requests discussion and suggestions for improvements.
Please refer to the current edition of the ``IAB Official Protocol
Standards'' for the standardization state and status of this protocol.
Distribution of this memo is unlimited.
```

Figure 2: The main application screen of quaneko.

quaneko presents itself with a main application window that is split into three parts:

1. The **search control area**. This is the top area where you can type a search query and select a search index.
2. The **result view** at the left shows the result set of a query.
3. The **preview** at the right previews the contents of a file in plain text.

Creating A New Index

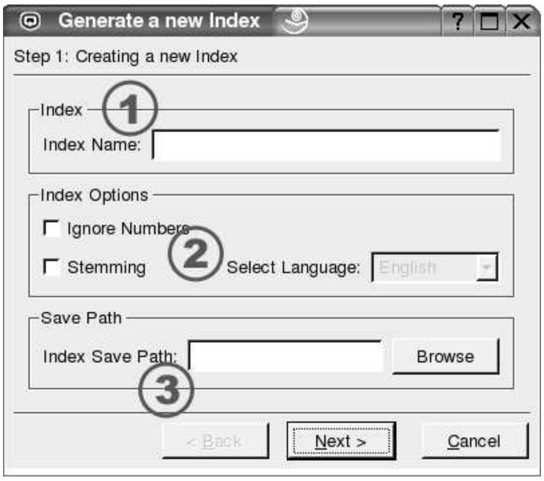


Figure 3: The first step of the wizard to create a new index.

Before you can search your documents, you have to setup and create an index. Choose the menu 'Indexing' - 'Generate Index' to display the wizard for creating a new index (see Figure 3).

In the first screen during the creation process there are 3 areas:

1. **Index Name:** Type a unique name for your new index. The name is case sensitive.
2. **Ignore Numbers:** Tick to exclude numbers from being indexed. If this is checked it means that you won't be able to search for numbers such as years or numeric IDs. Indexes without numbers can be considerably smaller, depending on the files that are indexed.
   Stemming: Tick to enable stemming. Stemming indexes the stem of words rather than the full word (e.g. "cycl" for "cycling", "cycled", "cycles" and "cycle"). It allows you to find all documents that contain any of those words by searching for any of them. Stemming typically reduces the index size by about 15 to 33%. If you activate stemming, you need to chose a language for the stemmer. If your documents are written in different languages, create individual indexes for them or disable stemming.
3. **Index Save Path:** Chose a directory where you want to store the index files. The files will be stored in a sub folder that is named like the unique index name. E.g. if you type "c:\temp" here and the index is called "MyData", the index files will be placed in "c:\temp\MyData\".

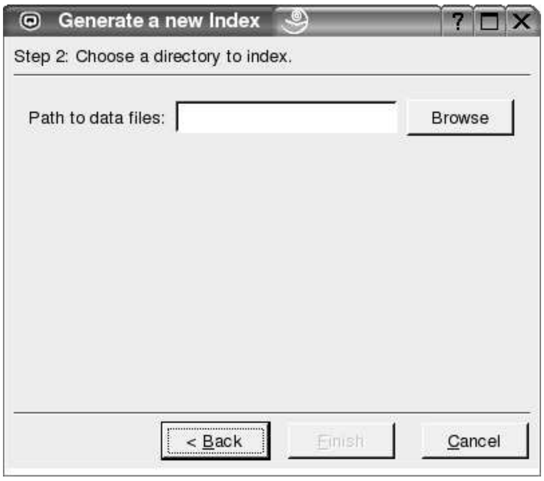
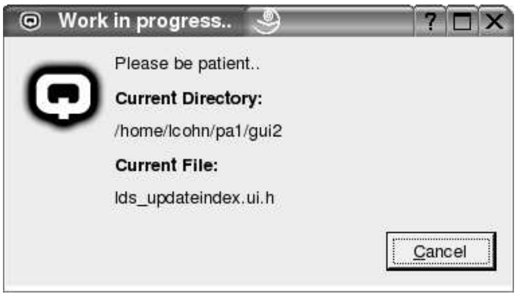


Figure 4: The second step of the wizard to create a new index.

In the second step of the index creation wizard you can parse an initial directory into the new index (see Figure 4). Choose the directory in which the documents are located that you want to index. You can add more folders or individual files to the same index later.

If all the given directories exist and the index name is a unique new index name, quaneko will now parse the files in the given data directory into the index. This can take some minutes or hours depending on the number and size of the data files. During the process you can see which directory and file is currently being processed.


Please be patient..

Current Directory:

/home/lcohn/pa1/gui2

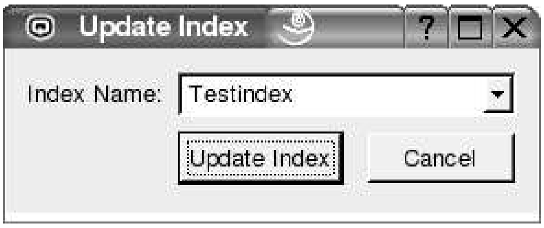
Current File:

lds_updateindex.ui.h

Figure 5: quaneko is parsing, converting and indexing the files.

Updating An Index

Choose the menu 'Indexing' - 'Update Index' to display the update dialog (see Figure 6). Updating an index means to parse all files and directories that were added since the last update to the index. Further, all files that have changed will be re-parsed. Depending on the number of files there are in the index, an update can take a rather long time.

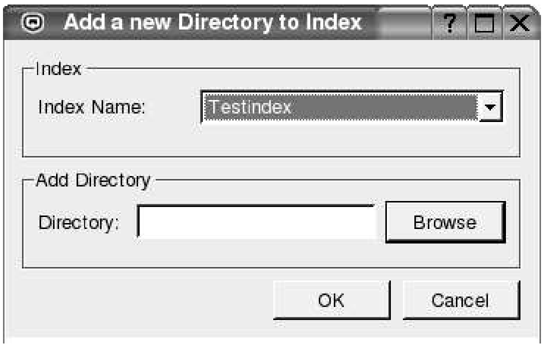


Index Name: Testindex

Figure 6: The update dialog.

Adding A Directory

Choose the menu 'Indexing' - 'Add Directory' to add an individual directory to an existing index (see Figure 7).



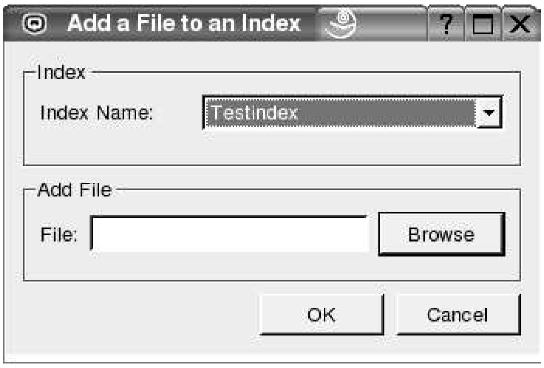
Index Name: Testindex

Directory:

Figure 7: Adding A Directory.

Adding A File

Choose the menu 'Indexing' - 'Add Directory' to add an individual directory to an existing index (see Figure 7).



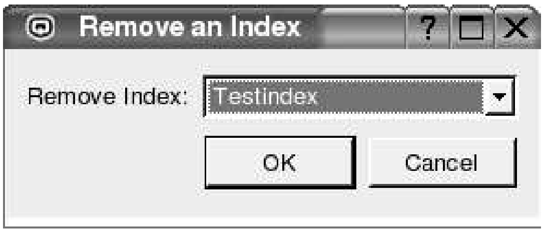
Index Name: Testindex

File:

Figure 8: The Add a File Screen.

Removing An Index

Choose the menu 'Indexing' - 'Remove Index' to remove an existing index (see Figure 9).



Remove Index: Testindex

Figure 9: The Remove Index Screen.

Chapter 6

Command Line Interface (CLI)

The CLI is available in subdirectory 'cli' of the quaneko installation directory. This chapter describes the available commands. Further configurations are in the chapter Filter Configuration - Manual.

The CLI offers a number of command line switches:

```
cli [ --create INDEXNAME PATH \
               [ --ignore-numbers] [ --stemming LANGUAGE]  |
     --parse   INDEXNAME FILE |
            --update INDEXNAME |
     --query   INDEXNAME EXPRESSION |
     --remove INDEXNAME
     --list]
```

`--create`

Creates a new index with the given name, e.g. 'Index01'. The index files are stored in the folder /tmp/Index01/. The created index is empty and ready for files to be parsed in.

```
./cli --create Index01 /tmp
```

Additionally, you might want to pass the following arguments to the `--create` command:

`--ignore-numbers`

The command line switch --**ignore-numbers** configures the index to ignore words that are plain numbers. This can reduce the index size by about 10-20%.

`--stemming LANGUAGE`

The command line switch --**stemming en** enables stemming for the given language. Stemming can reduce the index size significantly but can only be activated for one language for the whole index. Please refer to the Appendix for a list of supported languages.

`--parse`

Parses all given files and directories with subdirectories into the given index, e.g. 'Index01'.

```
./cli --parse Index01 /home/tux/data /home/tux/somefile.t
xt
```

`--update`

Updates all files and directories in the given index, e.g. 'Index01'. Files that are no longer existent are removed from the index. Directories are never removed. New files in previously parsed directories are added. An update always makes sure that all future search queries will return correct results.

```
./cli --update Index01
```

`--query`

Searches the index (e.g. Index01) for the given word (e.g 'fishes') and lists all files which contain this word.

```
./cli --query Index01 fishes
```

Instead of a word you can also pass a logical expression to this command for a more complex specification of the search query. Please refer to Appendix for more details about supported logical expressions.

`--remove`

Removes the given index. Please note that only the index configuration is removed. The index files are still on the disk and need to be removed manually.

`--list`

Lists all available indexes.

Chapter 7

Appendix

Supported Languages for Stemming

English	en
Danish	da
German	de
English	en
Spanish	es
Finnish	fi
French	fr
Italian	it
Dutch	nl
Norwegian	no
Portuguese	pt
Russian	ru
Swedish	sv

Logical Expressions in Search Queries

AND Both words must occur in the file. This is the default operator when more than one word is specified.
Example:
`Tux AND Igloo`

OR Either of the words or both of the words left and right of the operator must occur in the file.
Example:
`Tux OR Tuxedo`

NOT The word after the operator must not occur in the file.
Example:
`cheese NOT holes`

- Short for NOT.
Example:
`cheese -holes`

+ Short for AND.
Example:
`cheese +swiss`

Index