indexGUI user manual

A whoosh based educational document indexer

indexGUI user manualRev 1.0

Xilinx like template for LaTeX.
(C) 2013 Cedric DEBARGE <debarge.cedric@gmail.com>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Contents

Chapter 1

Introduction

Introduction

Specifications

The indexGUI project aims to provide a solution to index educational documents. This project is born to help my girlfriend to digitalize her (very) huge manual stack.

With indexGUI you can :

- index documents from various sources: pictures, texts and pdf
- retrieve a list of documents which satisfy given search criteria
- export in PDF a subset of the query result
- use it with its embedded gui or directly in a terminal (without any graphical dependency)

It is written in python and uses many other projects

- whoosh (http://pypi.python.org/pypi/Whoosh/)
- pyPdf (http://pybrary.net/pyPdf/)
- pytesser (http://code.google.com/p/pytesser/)
- pyText2Pdf (http://djangosnippets.org/snippets/1778/)
- P.I.L. (http://www.pythonware.com/products/pil/)
- slate (http://pypi.python.org/pypi/slate)
- wxPython (http://www.wxpython.org/)
- tesseract (http://code.google.com/p/tesseract-ocr/)

This project is distributed under the terms of a free licence (GPLV2). This means that you can freely reproduce, copy or modify all the parts of the project (according to the given licence). See the COPYING file for more details about this.

Acknowledgement

This project is dedicated to Mimo, my beloved beta tester.

Chapter 2

Installation

Installation

Downloading the project

indexGUI can be downloaded from the indexGUI web site (http://thewired.doesntexist.org). This package contains the indexGUI python source code and some already packaged python projects.

Mettre ici le lien exact quand il sera dispo

Python requirements

This project has been tested with python 2.7.2 with 32bits and 64bits systems. It should work with any python 2.6+ version.

Python libraries

indexGUI uses many other libraries. You have to install them before trying to run the program :

- whoosh
- pyPdf
- pytesser

Most of them are already packaged in several Linux distributions but *easy_install* will also work fine.

Non pythonic dependancies

indexGUI uses tesseract to extract text from pictures. You can download it from the tesseract-ocr web site (http://code.google.com/p/pytesser/) or use a pre-packaged archive.

GUI

indexGUI comes with an embedded GUI coded using wxPython. If you want to use it, you will have to install wxPyhton (tested with version 2.8.12) on your system.
Nevertheless, indexGUI can also be used in command line only. In that case, no wxPython installation is required. See the next chapter to learn how to choose between GUI and CLI mode.

indexGUI installation

Just unzip the archive somewhere on your system. That's all.

Operating system portability

Thanks to the python portable nature, indexGUI supports several OS. It has been tested on Linux, MAC OSX Lion. I think it will also work on *BSD systems.

Chapter 3

Using indexGUI

GUI mode

Start the program

Just go to the indexGUI installation directory and launch indexGUI.py with python. According to your operating system and your desktop environment indexGUI can be launched by :

- double click the indexGUI.py file
- double click the indexGUI.py file and select "Launch"
- right-click the indexGUI.py file an select "open with python"
- open a terminal in the indexGUI directory and type "python indexGUI.py"
- open a terminal in the indexGUI directory and type "./indexGUI.py"
- drag the indexGUI.py file on the python launcher program

If the magic sequence is entered successfully, you will be welcomed by the indexGUI main (and only) window with the find tab selected:

Add new document(s)

indexGUI will extract the text from the given file(s) and then index them in a database. indexGUI can index several types of documents :

- PNG picture
- JPEG picture
- TIFF picture

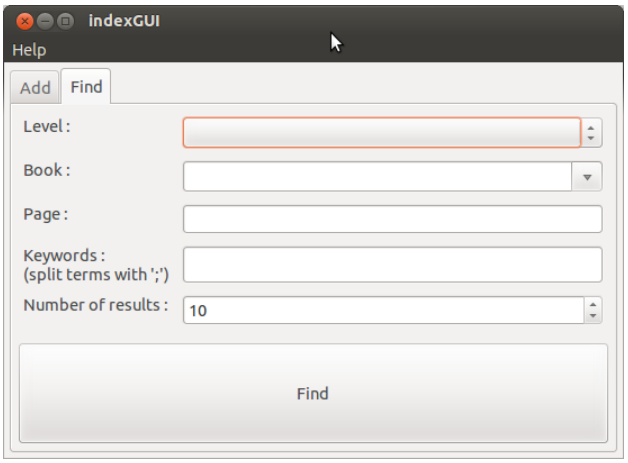

Figure 3.1: indexGUI main window

- BMP picture
- plain text document
- PDF document

In the indexGUI main window, select the *Add* panel. This panel allows you to select your document(s), set some options and finally launch the index process.

Figure 3.2: indexGUI add panel

You can import only one file or all the suitable documents from a given directory. Just select the type of import using 1 and then click 2 to find it on your system. Once the target is selected, the path information below will indicate the given location.

A level may be specified using 3 (either a new one or previously defined one). A blank field is also well accepted.

indexGUI creates a book structure (see next point for more details) for each new document or directory. By default, the book name is the directory name (in case of directory import) or the file name without the extension (for a single file import). If the 4 checkbox is activated, the

default book name will be overwritten by the name given in **5**.

notice : indexGUI also uses a concept of page. If you import a whole directory, a page will be created for each file. The page name is defined as the file name without the extension. It is possible, for example, to scan each page of a book with an incrementing number as file name and then import it into indexGUI. This way, the find feature will tell you which page contains the data you need.

Once you have filled all the elements, you can click the *Add* button **6**. indexGUI will then proceed to the indexation and you will be notified at the end of the operation. According to the number and the type of documents, this process may take some time. See below for more details about the way indexGUI handles your files.

Directory structure

indexGUI automatically creates 2 subdirectories in its main directory. The first one, *database*, contains the whoosh database by itself. The second one, *documents*, contains one directory by book which itself contains a copy of the the indexed documents for the given book.

If *database* or *documents* is missing at indexGUI start-up, the whole database is flushed and new *database* and *documents* are created.

Unless you know exactly what you are doing, it is recommended to avoid any change in one of these two directories.

Find documents

You can launch a request to locate the most suitable documents according to given criteria. This is done by selecting the *Find* panel, filling some options and then clicking the find button to let the magic of whoosh operate.

A level may be chosen in **1**. A blank level will disable level checking. New levels are automatically added when a new document is imported. See the *Add* document section above for more details.

The same behaviour can be applied with book name in **2**. And yes this is an editable combo box because I plan to support regexp search in book name in a future release.

A page can be chosen in **3**. Like the 2 items above, a void area will disable page filtering.

4 is the most important part of this form (although it can be left blank). Here you can enter words or phrases to search. Multiple search terms

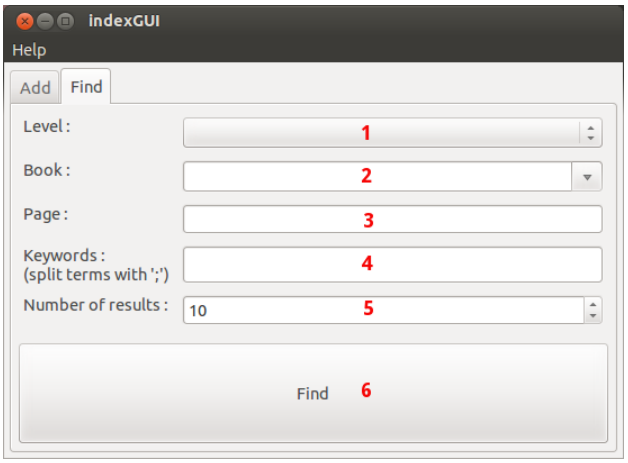

Figure 3.3: indexGUI find panel

must be separated by ';'. If more than one term is entered, a logical *AND* on all given terms is performed.

The maximum number of result is tuned by using **5**.

Once you have filled all the elements, you can click the *Find* button **6**. indexGUI will then parse all the documents in the database to extract the most interesting ones. The results are displayed in the third panel : *Find Results*.

Browse results

The results of the find request are displayed in the *Find Results*.

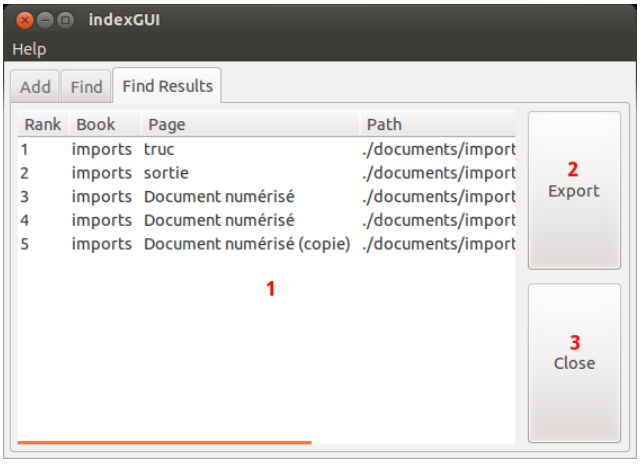

Figure 3.4: indexGUI find results panel

The ranked results are listed in **1**. Double click one line to open a copy of the selected result. This file is located in your OS temp dir (/tmp for Linux) and will be deleted at indexGUI close (if not open anymore). So be sure to save it somewhere else before closing.

You can also select more than one result using the "control" key (at least under Linux) and export all the selected files in pdf by clicking the *Export* button (2).

The *Close* button (3) closes the *Find Results* panel.

CLI mode

indexGUI comes with command line capabilities allowing to quickly *add* and **find** documents the same way you would with the GUI.

Add command

The *Add* command can be invoked using the *cmd* option.

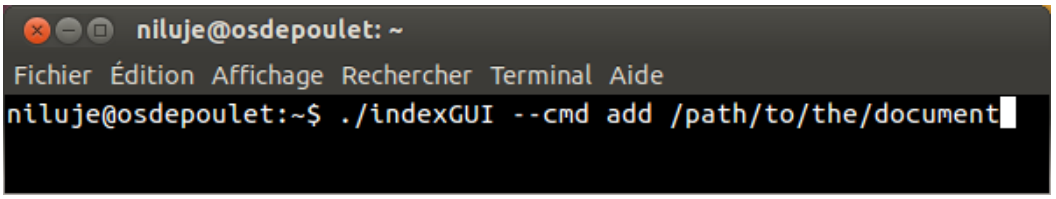

Figure 3.5: Add command

The path to the document might point to either a single file or a whole directory. You can also specify *level* and *book name* using respectively the -l and -b option. Long form options are also available (see the on-line help for more details).

Find command

The *Find* command also reflects the GUI capabilities. Use the following options to specify request parameters :

short option	long option	description
l	level	specifies the level
b	book	specifies the book name
c	content	specifies the key words
p	page	specifies the page name
f	out_format	specifies the output format
r	results_len	specifies the maximum number of results

notice : If you choose pdf or html format for the out_format option, the results will be placed in a results folder.

On-line help

indexGUI provides svn like on-line help for each command. See example below for more details :

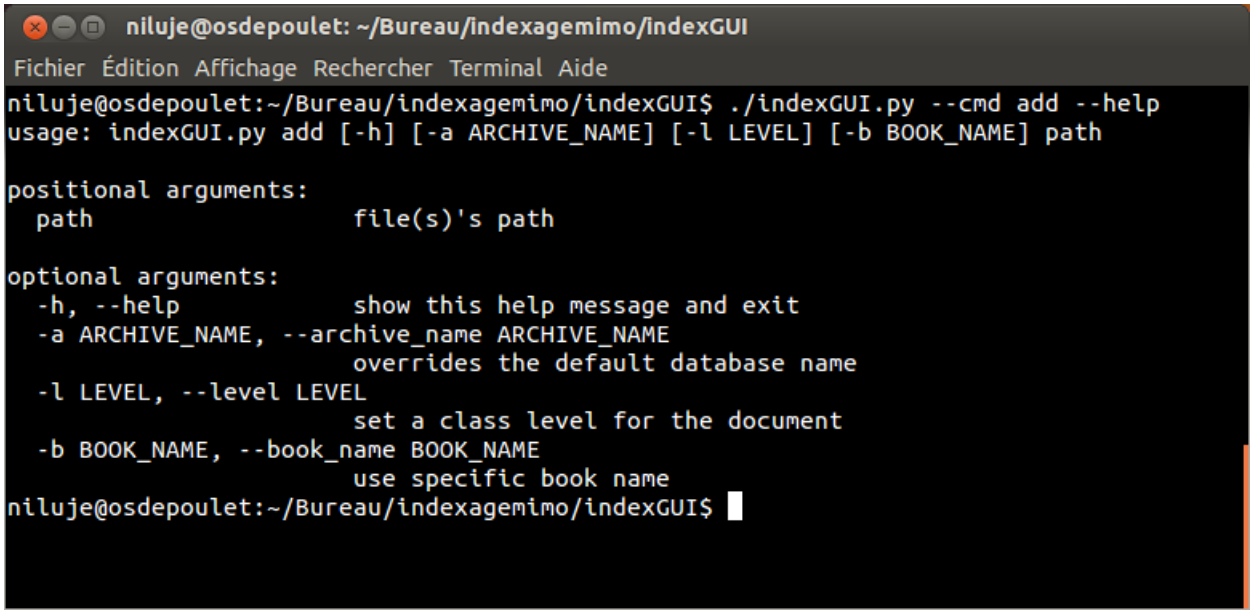

Figure 3.6: indexGUI add command on-line help

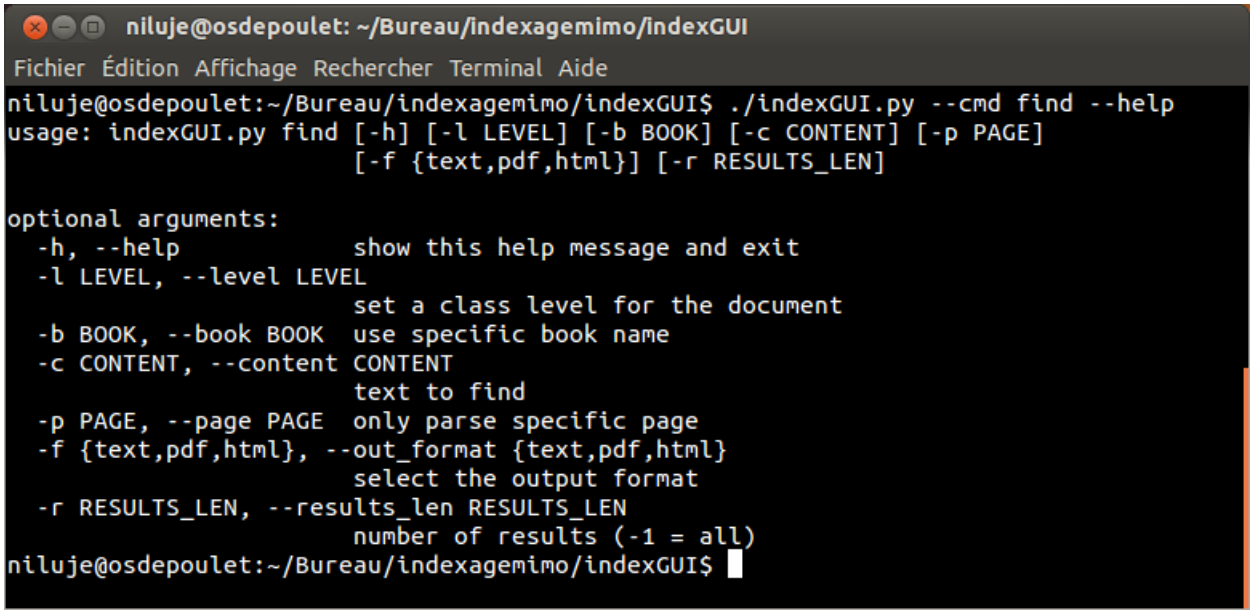

Figure 3.7: indexGUI find command on-line help