`mbib` Documentation

As of February 22, 2018, this is a work in progress

`mbib` is a literature manager, with capabilities similar to `JabRef`, but intended to better cope with large databases (thousands of references). Key features and limitations are:

- Import from and export to BibTeX
- Import records from DOI and PubMed identifiers
- Push citations to Texmaker/TexStudio
- Push citations to OpenOffice/LibreOffice (collectively referred to as "`OOo`" below). Formatting of bibliographies in `OOo` piggybacks on `JabRef`. Thus, you need to have `JabRef` installed in order to fully use `mbib` with `OOo`.
- Written in Python3.
- Console-based GUI, based on the `urwid` and `urwidtrees` Python libraries.
- Data are stored in a SQLite database.
- Developed and tested only on Linux, and will likely not work out of the box on other platforms. (Volunteers for porting it to other platforms are welcome.)

1 Preliminary notes

1.1 Motivation

I started writing `mbib` after my previous literature manager (`bibus`) broke down because of growing incompatibilities with LibreOffice, wxPython etc. I tried using `JabRef` for a while, and while it's really pretty good in many ways, it really bogs down once you have several thousand references in your database. (The same goes for Zotero and Mendeley.)

1.2 Status

At present, `mbib` is alpha software. Incompatible changes might still happen to the code and the database structure. However, in the latter case, I will provide a script for migrating the database to the new format (since I will have to migrate my own data anyway).

The program was written with my own needs in mind; I work in biochemistry and use PubMed as my main online literature database and therefore have implemented import of references via PubMed identifiers. People in other fields might miss tighter integration with other databases. I'm open to adding support for those, but unless prodded, it won't be a priority. Similarly, some other bits of functionality are tailored to my own personal preferences and may seem a little idiosyncratic to others.

The code violates all manner of software engineering gospel. There are no unit tests, and I don't plan to add them; the doc strings are a bit spotty and not formatted for automatic conversion into API docs. That said, I will give an overview of the program structure below, which hopefully will help you find your way through the code.

2 Installation

In the following, I am going to describe how things work on *my* system. I am running Debian with a KDE desktop. I don't suppose there will be any major differences with other Linux distros or window managers, but I am not going to verify this by experiment. If you manage to get it to work on other systems and have some specific tricks to share, please let me know, and I will include them here.

2.1 Prerequisites

In order to run `mbib`, you first need to install these programs and libraries:

- Bash
- Python3
- SQLite
- The `urwid` and `urwidtrees` libraries for Python3
- If you intend to use `mbib` with `OOo`, you will also need `JabRef`
- If you want to copy items to the X clipboard, you will need `xclip`
- For viewing or emailing PDF files, `mbib` relies on `xdg-open` and `xdg-email`

On Debian, all of these prerequisites can be installed through the system's package manager. A copy of SQLite already comes as part of the standard library when you install Python3, but you may also want to install the `sqlite3` package, which provides the command line client that lets you run SQL statements on your database.

The `xdg-open` and `xdg-email` utilities are probably installed by default on any graphical Linux desktop; in Debian, they reside in the `xdg-utils` package.

2.2 Installing `mbib`

Just clone or unzip the repository and add the main directory (mbib) to your bash $PATH.

2.3 Configuration

The first program start will generate a configuration `mbib.ini` file in your home directory. The settings in this file are explained in comments.

3 Tutorial

Here we give an overview of the general work flow. A more complete description of all features will follow later.

3.1 Starting the program

Assuming you have installed all prerequisites and added the `mbib` directory to your shell's `$PATH`, you should be able to open a console window and run

```
mpalmer@rehakles:~$ mbib.sh
```

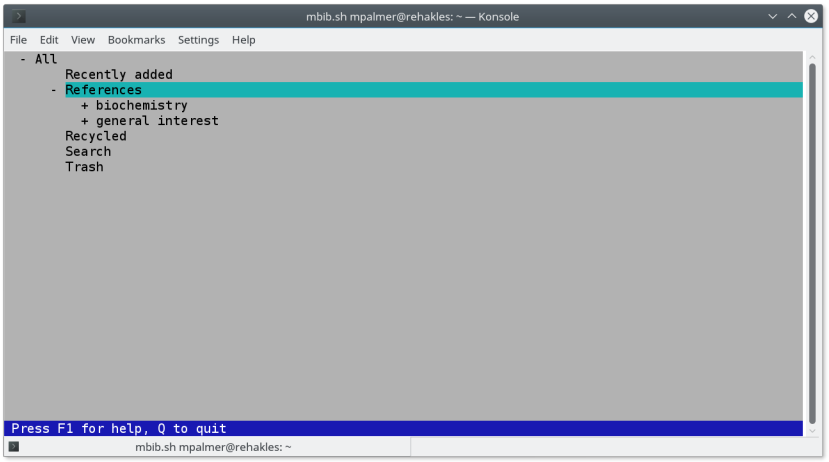
After the first start, the expected output should be

```
Config file /home/mpalmer/.mbib.ini not found!
Create default configuration file and proceed (1) or exit (2)?
1) proceed
2) exit
#?
```

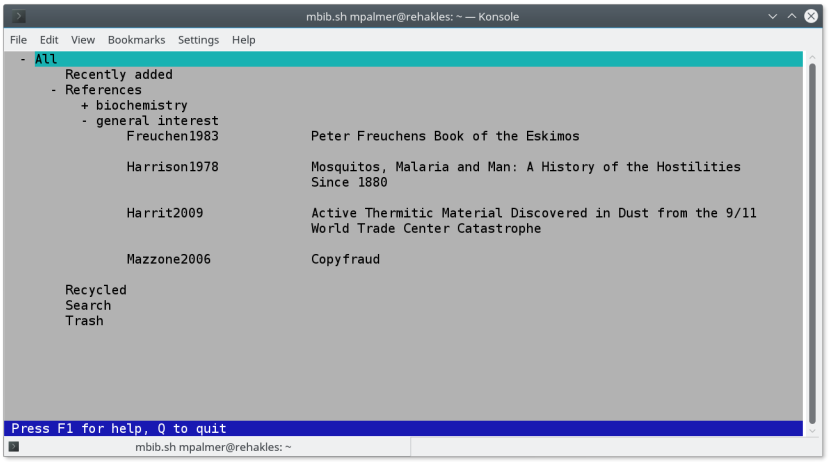
Press “1” and enter, and you should see

```
#? 1
OK
Database file /home/mpalmer/mbib.sqlite not found. Create? (y/n)
```

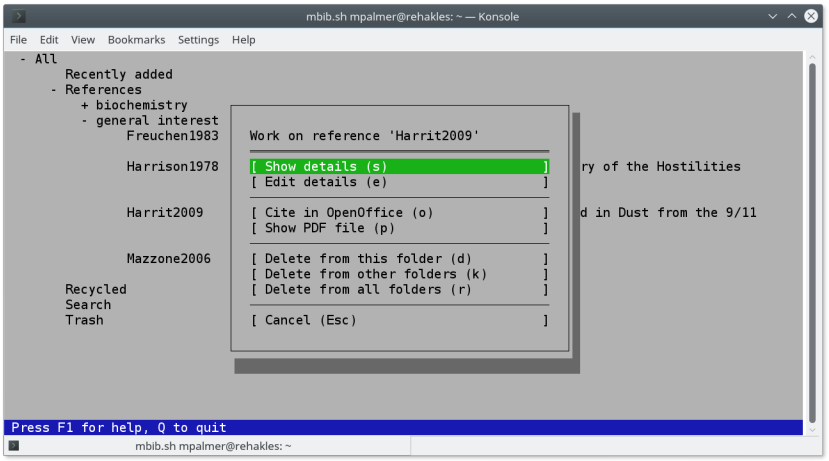
Press “y” and the program should start. The interface should look like this:



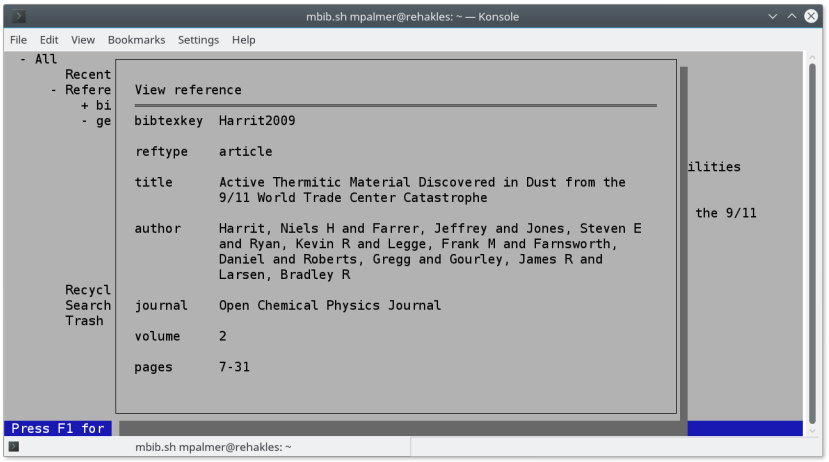
`mbib` displays all references in a tree structure. The main `References` folder at this time has two sub-folders. Use the down-arrow key (or the mouse) to highlight the “general interest” folder and press `F2`. You should now see the references contained in this folder:



Use the arrow keys or the mouse again to highlight the record labelled `Harrit2009`, press `enter`, and you should see a context menu:



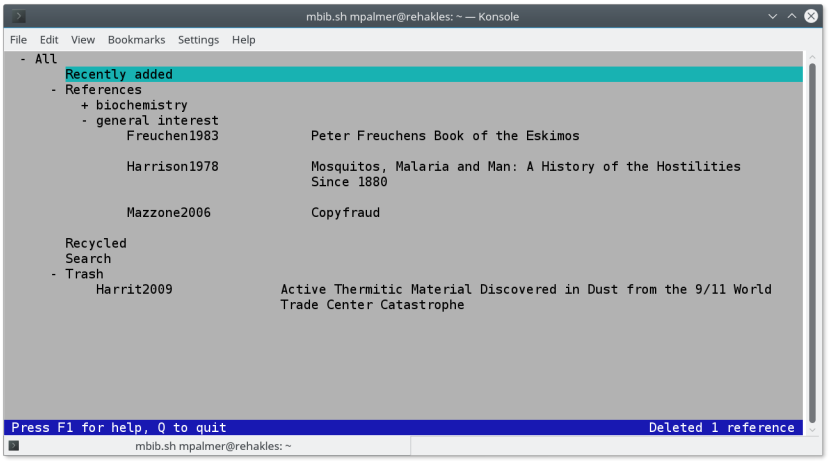
Use the arrow keys and `enter`, the indicated shortcut keys, or the mouse to select the desired menu item. If we select `Show details`, the `View reference` dialog comes up:



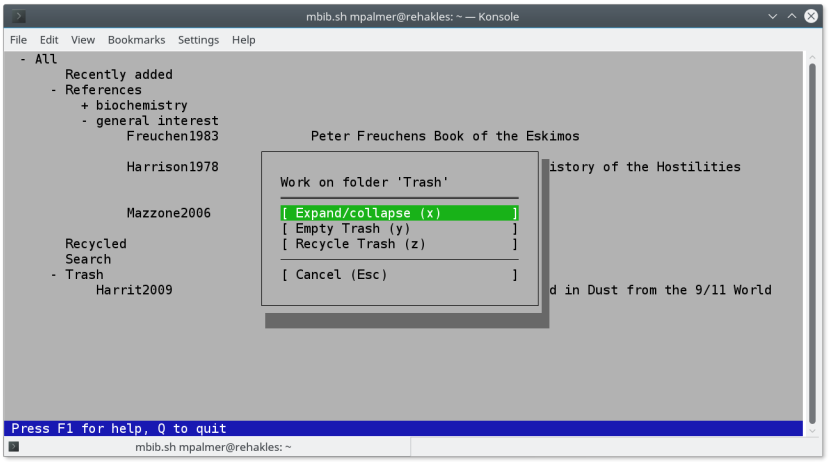
Use the arrow keys again to scroll through this dialog. The abstract is at the bottom. Above it, there are three fields (let's call them buttons) that are active. The one in focus (and which would respond to pressing `Enter`) is highlighted in blue; or you can again use the mouse. Triggering a button will open a browser window and take you to the URL.

Press `Esc` to close the dialog. Press `Enter` again to reopen the menu. Maybe you prefer to disregard the scientific facts about 9/11[1] and therefore want to delete this reference. Press `d` or `r`—in this case, they are equivalent—to delete the reference. After confirmation, the reference now shows up in the Trash folder:

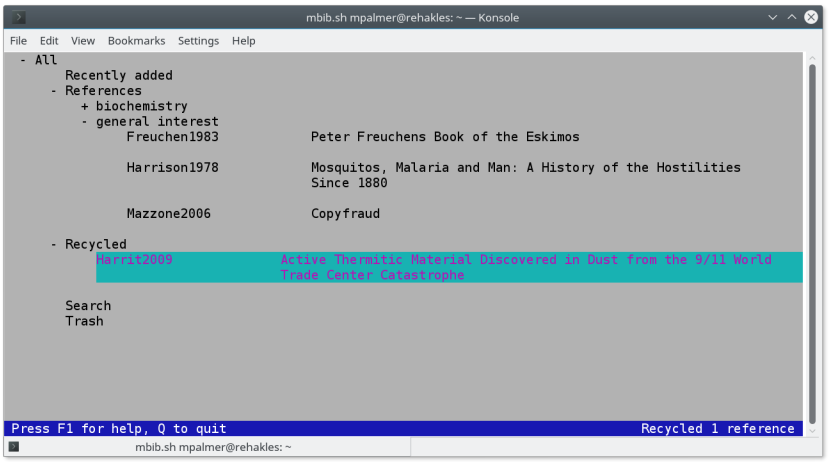
[1] After all, hasn't "truther", and by affiliation truth itself, become a dirty word?



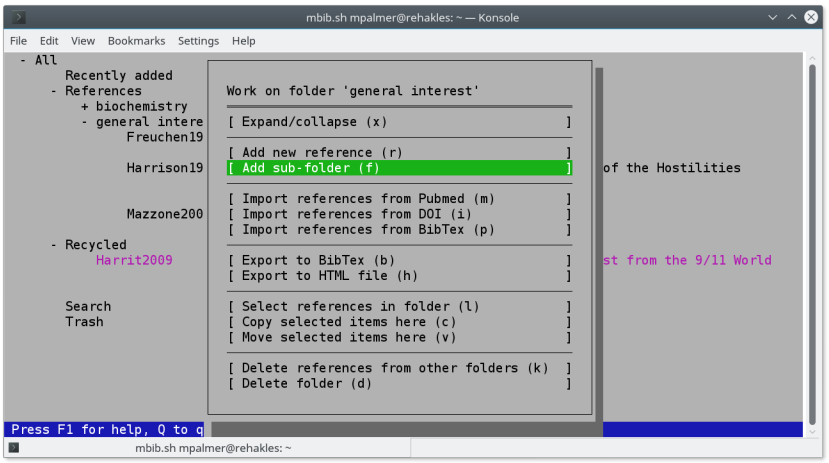
Navigate down to the Trash folder and press `Enter` or double-click on it to bring up this folder's context menu:



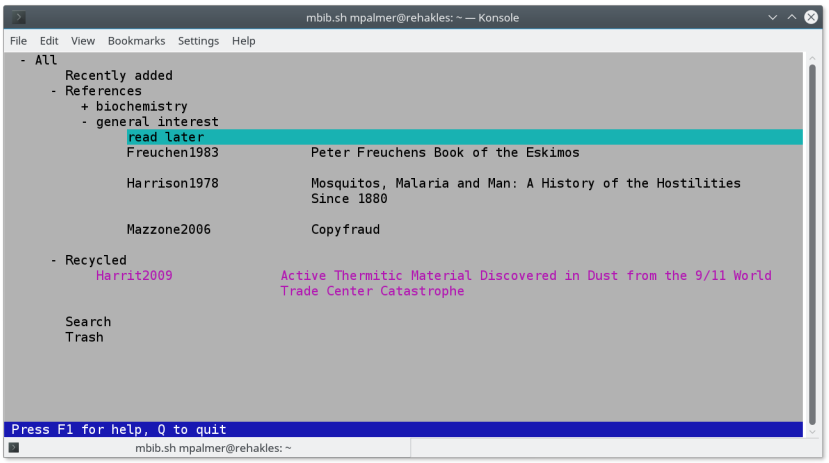
Select the "Empty Trash" option to get rid of the deleted reference entirely and irreversibly. Alternatively, use the "Recycle" option to move the deleted reference to the Recycled folder. Navigate to it and press the Space key to select it:



Let's assume that you intend to look into this matter some other time, and you want to collect such references in a separate folder. Navigate to the "general interest" folder and activate its menu (Enter or double-click), then activate the option "Add sub-folder":[2]

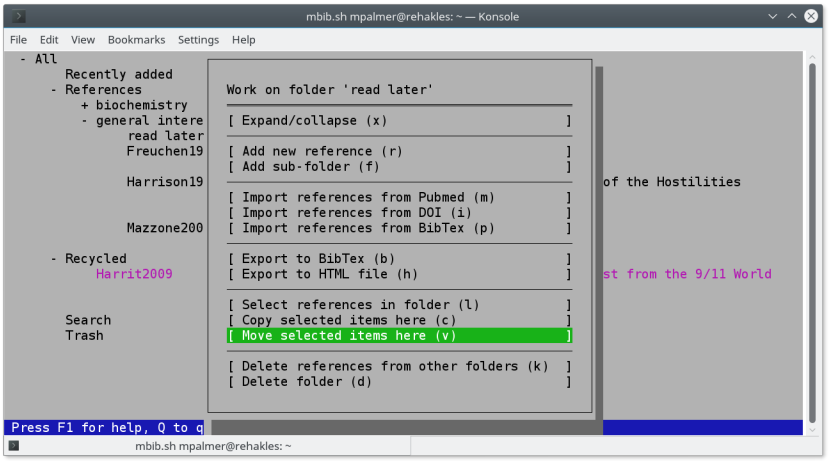


This will bring up a dialog that lets you enter the name for the new sub-folder. After completing it, it should look similar to this:



The new folder (here named "read later") has no + or − before it because it is empty. We can move the previously recycled reference here. Navigate to the new folder and activate its context menu, then select the "Move selected items here" option:

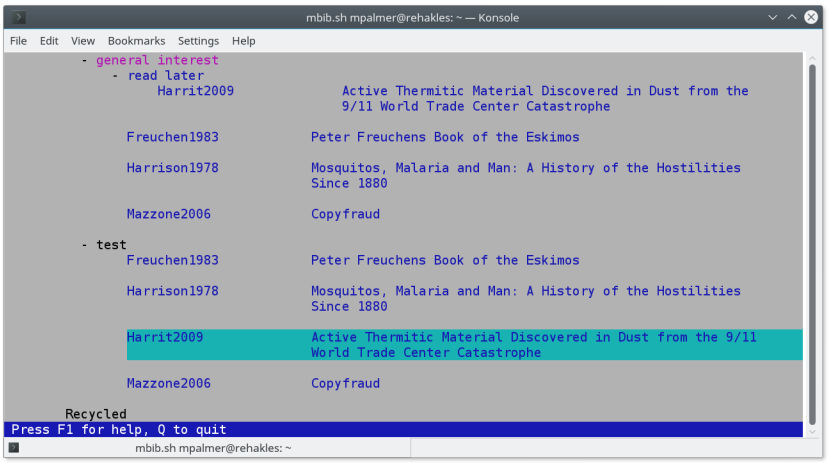
[2] The folder context menu is quite long. In these screenshots, which use a small window, some items at the bottom are missing; use the down arrow key to scroll them into view.



After completing this, you should now have the recycled reference in the new folder. (Use the F2 to expand the folder view.)

In addition to one or more individual references, you can also select and then move or copy folders. As a practical exercise, create a folder “test” directly below “References”. Navigate back to “general interest” and select it with the space bar. Go back to “test”, open its menu and select “Move selected items here”.

You will now have cloned the entire folder with sub-folders and references. You can flatten that nested folder structure by opening the menu for “test” again and selecting “Flatten folder.” This will give you the following result:



You will notice that the references in both “test” and “general interest” are highlighted in blue; this indicates that they are, by virtue of being contained in a selected folder somewhere in the tree, part of the currently active selection. Deselecting “general interest” (by hitting Space on it again) will also deselect the references.

Let us now turn to importing references. I will discuss this for PubMed. Go to the PubMed website (https://www.ncbi.nlm.nih.gov/pubmed/) and type the following into the search bar:

```
blue-native[ti] anal-biochem[so]
```

This will give you (at the time of this writing) 16 results. Choose the “PMID list” option from the display format control to see the identifiers for these papers. Use your mouse to select them.

In `mbib`, create a new sub-folder in biochemistry (e.g. called “blue-native”). Open its menu and choose “Import references from PubMed”. Hold down the Shift key and click you middle mouse button to paste the selected PMIDs into the dialog. Click OK to start the import.

The procedure for importing references from DOIs is similar. When importing BibTex, you can paste directly, too; be warned, however, that pasting large chunks of text may be quite slow. It is better in this case to first save the BibTex to file, and then enter the file name into the BibTex import file dialogue.

To export individual folders to BibTex, select the appropriate item from their context menu. The same operation performed on the References folder will export the entire database. The exported file will be free of duplicates; you are free to keep copies of the same reference in as many folders as you wish.

4 Some gotchas

There is more to `mbib`’s functionality, but it will have to wait a couple of days; I must get to work on some urgent stuff right now. Meanwhile, here are some things to keep in mind.

- Moving selected items from search results: the search folder is basically a simple folder in the database. So, if items in this folder are selected and moved, the copies in the permanent folder stay where they are.
- All the special top-level folders are special and protected just inside mbib, because specific operations are available and others excluded; in particular, they cannot be deleted, moved, or renamed. However, no such protective restrictions apply when working with the database using an SQLite shell.

  Generally speaking, it is a good idea to make a safety backup copy of the database file before performing major surgery in SQL.