cahier Documentation

Release 0.1.1

Louis Paternault

March 13, 2016

One-directory-a-day calendar management

Note: I am no longer using this software, so it will not be improved. Feel free to ask questions and to submit bugs anyway, but this will not be my priority.

– Louis

CHAPTER 1

Rationale

Note that althougt I wrote this program for this particular purpose, I had extensibility in mind. Please read on the usage and configuration sections before deciding that this program does not fit your needs.

As a part of my job as a teacher, I have to log my work on a daily basis (*cahier de textes* in French, hence the name). To do so, I use IkiWiki. Each entry is stored in a markdown file named *YYYYMMDD.mdwn*, and each attached document is located in the *YYYYMMDD* directory. So, after a few days, an `ls` gives:

```
$ ls
20130906       20130927.mdwn  20131018b      20131122       20131217.mdwn
20130906.mdwn  20130930.mdwn  20131018b.mdwn 20131122.mdwn  20131218.mdwn
20130909.mdwn  20131001.mdwn  20131104.mdwn  20131125.mdwn  20131220
20130910.mdwn  20131002       20131105       20131126.mdwn  20131220.mdwn
20130911.mdwn  20131002.mdwn  20131105.mdwn  20131127.mdwn  20140107.mdwn
20130913.mdwn  20131004.mdwn  20131106.mdwn  20131129a.mdwn 20140108.mdwn
20130916.mdwn  20131007.mdwn  20131108       20131129b.mdwn 20140110
20130917       20131008       20131108.mdwn  20131202.mdwn  20140110.mdwn
20130917.mdwn  20131008.mdwn  20131112       20131203       20140113.mdwn
20130918       20131009.mdwn  20131112.mdwn  20131203.mdwn  20140114.mdwn
20130918.mdwn  20131011       20131113a.mdwn 20131204.mdwn  20140115
20130920       20131011.mdwn  20131113b.mdwn 20131206.mdwn  20140115.mdwn
20130920.mdwn  20131014.mdwn  20131115       20131209.mdwn  20140117
20130923.mdwn  20131015.mdwn  20131115.mdwn  20131210.mdwn  20140117.mdwn
20130924       20131016       20131118.mdwn  20131211       20140120.mdwn
20130924.mdwn  20131016.mdwn  20131119.mdwn  20131211.mdwn  20140121.mdwn
20130925.mdwn  20131018a      20131120       20131213.mdwn  20140122
20130927       20131018a.mdwn 20131120.mdwn  20131216.mdwn  20140122.mdwn
```

There are several problems.

- Command line completion is broken: file names are so close that it is almost useless:
  - typing in the file name to edit it is frustrating;
  - typing in the directory name to create it is frustrating;
  - typing in the directory name to move a file to it is frustrating.
- When editing the file, I have to fill the date and time of the calendar entry, although this information is contained in the file name, and could be guessed (my timetable does not change from one week to another).
- I have different classes, with different calendar, and I want the right class to be automatically taken into account.

I wrote this program to automate this task.

CHAPTER 2

Download and install

See the main project page for instructions, and changelog.

CHAPTER 3

Usage and configuration

3.1 Base commands

The base commands operate on the *right* calendar, with the *right* date.

- The *right* calendar is determined using profiles (more information in section *Profiles*). In this configuration, you can associate several directories to one calendar. For instance, if I associate directory `~/courses/cs` with calendar `~/calendars/cs101`, a `cahier` command run in `~/courses/cs` (or any of its subdirectories) will consider the associated calendar. This is useful if you have several calendars.
- The *right* date is also configured in the profiles. You can register a simplified version of your time table, so that when editing a new entry, its date is calculated using this time table, and the appropriate date and time is filled in the template.

The base commands are (a full list of commands is available in section *Full command line options*).

- `cahier new` Create a new entry, at the right place, with the right date, and start editing it.
- `cahier edit` Edit the last created entry.
- `cahier attach` Attach a file to the last entry, that is:
  - create the directory if necessary;
  - proprocess files if relevant: for instance, I want command `cahier attach foo.tex` to compile `foo.tex` as a pdf, and attach the resulting pdf;
  - copy the files in this directory.
- `cahier cd` Start a shell in the calendar directory
- `cahier wiki` Arguments to this command are passed to `ikiwiki`, in the calendar directory. More options are available as well, like `cahier refresh` which compile the wiki (whatever the working directory is).
- `cahier git` Arguments to this command are passed to `git`, called in the calendar directory.

3.2 Configuration

3.2.1 String formatting

Strings of configuration files are formatted in two ways.

- They are formatted according to the rules of the configparser module.

- They are formatted, using the `{key}` format, with the following values:
  - `basename`: basename of the edited file (without directory nor extension);
  - `configdir`: `cahier` configuration directory;
  - `dirname`: directory of edited file;
  - `filename`: like basename, with the extension.

3.2.2 General configuration

General configuration is set in file `.cahier/cahier.cfg`. Example:

```
[options]
casesensitive: no

[bin]
editor: screen -t "$EDITOR" sh -c "(cd {dirname} && $EDITOR {filename})"
shell: screen

[wiki]
options: --verbose --no-rcs
fileformat: %%Y%%m%%d
fileformat-length: 8
template: {configdir}/templates/template.mdwn
```

- `options`:
  - `casesensitive`: Set whether profile names are case sensitive or not.
- `bin`:
  - `editor`: Command to call to edit files.
  - `shell`: Shell to invoke with `cahier cd`.
- `wiki`:
  - `options`: Options used when calling `ikiwiki`.
  - `fileformat-length`: Length of date in the file names (e.g. if your file names are *YYYYMMDD-foo*, `fileformat-length` will be 8; if your file names are *MMDD-foo*, `fileformat-length` will be 4).
  - `fileformat`: Date format of files, as recognized by the datetime.strptime() function.
  - `template`: Template to use for newly created files.

3.2.3 Profiles

Profile configuration is set in `.cahier/profiles/NAME.cfg`. Example:

```
[DEFAULT]
ikiwiki: ~/prof/1S3/cahier

[options]
workdays: monday:08 tuesday:09 wednesday:08 friday:15:30

[config]
setup: %(ikiwiki)s/wiki.setup
```

```
[directories]
calendar: %(ikiwiki)s/seances
sources: ~/prof/1S-math ~/prof/1S3
```

- `DEFAULT`: Default values for all sections.
  - `ikiwiki`: This is an example of a trick taking advantage of *string formatting* to factorize configuration: the `%(ikiwiki)s` part in following options are replaced by value of this string.
- `options`:
  - `workdays`: Timetable, with times. This is a space separated list of *DAY:HOUR*. If this option is set, when editing a new file, the following date corresponding to one of those work days is used as the date. Otherwise, the current date and time is used.
- `config`:
  - `setup`: Path to the IkiWiki setup file.
- `directories`:
  - `calendar`: Path to the directory containing the calendar files.
  - `sources`: Paths associated to this profile. When calling `cahier` in one of those directory, the corresponding profile is used.

3.2.4 File plugins

File plugins are configured in files `.cahier/ftplugins/EXTENSION.cfg` (where *EXTENSION* is the extension of files impacted by this particular configuration file).

```
[preprocess]
    cmd: pdflatex {basename}
    name: {basename}.pdf
```

- `preprocess`: Commands to preprocess files before attaching them. For instance, with this example, LaTeX files are compiled, and their compiled version is attached to the current date.
  - `cmd*`: Values of keys starting with `cmd` are executed before attaching files.
  - `name`: Name of the file to attach, if different from the base file name.

3.2.5 File templates

Files `.cahier/templates/template.foo` are used as templates when editing a new file of type `foo`. Type is the extension of the file.

Template content is formatted as a Python string, with only one variable:

- `date`: the date and time of the log of the created file.

3.3 Full command line options

Here are the command line options for *cahier*.

Manage ikiwiki calendar items.

```
usage: cahier [-h] [-a] [-p PROFILE] [--version]
              {cd,git,new,edit,show,attach,rm,wiki,refresh,fresh,rebuild} ...
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
--version	show program's version number and exit

Sub-commands:

cd Run a shell in calendar directory.

```
usage: cahier cd [-h] [-a] [-p PROFILE]
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile

git Run git.

```
usage: cahier git [-h] ...
```

Positional arguments:

args	Arguments to pass to git

new Add new date item.

```
usage: cahier new [-h] [-a] [-p PROFILE] [-n] [-s NUMBER]
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
-n=False, --dry-run=False	Does nothing, but print entry that would be done.
-s=0, --skip=0	Number of entries to skip.

edit Edit last entry.

```
usage: cahier edit [-h] [-a] [-p PROFILE] [-n]
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
-n=False, --dry-run=False	Does nothing, but print entry that would be done.

show Show last entry.

```
usage: cahier show [-h] [-a] [-p PROFILE] [-n] [-s NUMBER]
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
-n=False, --dry-run=False	Does nothing, but print entry that would be done.

-s=1	Number of entries to show.

attach Attach file to current entry.

```
usage: cahier attach [-h] [-a] [-p PROFILE] [-n] files [files ...]
```

Positional arguments:

files	Files to attach to latest date.

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
-n=False, --dry-run=False	Does nothing, but print entry that would be done.

rm Remove the last entry

```
usage: cahier rm [-h] [-a] [-p PROFILE] [-n] [-f]
```

Options:

-a=False, --ask=False	Force asking profile and filename (if relevant).
-p, --profile	Force profile
-n=False, --dry-run=False	Does nothing, but print entry that would be done.
-f=False	Force deletion.

wiki Run IkiWiki.

```
usage: cahier wiki [-h] ...
```

Positional arguments:

args	Arguments to pass to IkiWiki

refresh (fresh) Alias for “wiki –refresh”

```
usage: cahier refresh [-h] ...
```

Positional arguments:

args	Arguments to pass to IkiWiki

rebuild Alias for “wiki –rebuild”

```
usage: cahier rebuild [-h] ...
```

Positional arguments:

args	Arguments to pass to IkiWiki

CHAPTER 4

Indices and tables

- genindex
- modindex
- search