# poca Documentation

*Release 1.1*

**Mads Mide Michelsen**

**Jun 26, 2021**

# Contents

Poca is a fast, multithreaded and highly customizable command line podcast client, written in Python 3.

**Contents**

# CHAPTER 1

## Features

- **Maximum amount.** Specify how many episodes the subscription should get before deleting old episodes to make room for new ones.
- **Override ID3/Vorbis metadata.** If you want *Savage Love* to have *Dan Savage* in the artist field (rather than *The Stranger*), poca will automatically update the metadata upon download of each new episode. Or set genre to be overwritten by *Podcast* as a default.
- **Filter a feed.** Only want news reports in the morning or on Wednesdays? Use criteria such as filename and title, or the hour, weekday or date of publishing to filter what you want from a feed.
- **Rename files automatically.** Not all feeds have sensibly named media files. Specify a renaming template like date_title to know what you're dealing with or to get alphabetical ordering to match chronology.
- **From the top.** A latecomer to *Serial* or other audiobook style podcasts? Specify `from_the_top` to get oldest episodes first, rather than the latest. To move on to later episodes simply delete old ones and poca will fill up with the next in line.
- **Keeping track.** Poca logs downloads and removals to a local file so you easily see what's changed. Or configure it with an SMTP server and get notified when a feed stops working.
- **Manage your shows** by editing an easy-to-understand xml file. Or use the accompanying tool to add, delete, sort them, or get info about their publishing frequency, average episode length and more.

Poca also: has excellent unicode support for feeds, filenames and tags, gets cover images for feeds, has the ability to spoof user agents, can pause your subscriptions, deals intelligently with interruptions, updates moved feeds (HTTP 301) automatically, and more.

CHAPTER 2

# Interface

All configuration is done in a single XML-format file. For cron job compatibility, Poca has a quiet mode in addition to normal and verbose.

CHAPTER 3

# Installation

You can install poca from pypi using pip. Be mindful that poca is python 3 so use `pip3`:

```
pip3 install poca
```

If you are upgrading from any pre-1.0 release, please see this Upgrade. To remove Poca simply do:

```
pip3 uninstall poca
```

## 3.1 Requirements

- Python 3.6 or later
- Third-party modules: `requests` `feedparser` `lxml` `mutagen`
- Pip will automatically install any one of these found missing
- A unicode capable terminal is recommended but not required
- Due to dependencies of third-party modules (lxml requires libxml2 v. 2.9.2 and libxslt 1.1.27) distros no older than Ubuntu 18.04 are recommended.
- For use on WSL, the "-g wsl" flag is recommended as it will substitute out characters known not to work on WSL (see https://github.com/microsoft/WSL/issues/75)

# CHAPTER 4

# Quickstart

```
[ ~ ] poca
No config file found. Making one at /home/user/.poca/poca.xml.
Please enter the full path for placing media files.
Press Enter to use default (/home/user/poca): /tmp/poca
  Default config succesfully written to /home/user/.poca/poca.xml.
Please edit or run 'poca-subscribe' to add subscriptions.

[ ~ ] poca-subscribe add
Url of subscription: http://crateandcrowbar.com/feed/

Author: The Crate and Crowbar                            PUBLISHED / 5 WEEKS
Title:  The Crate and Crowbar

Last episode: Episode 216: Videocrates Crowdog
Published:    24 Nov 2017

Avg. size of episode:   52 Mb
Avg. length of episode: 1h 52m                           M  T  W  T  F  S  S

Title of subscription: (Enter to use feed title)
Maximum number of files in subscription: (integer/Enter to skip) 5
Get earliest entries first: (yes/no/Enter to skip) no
Category for subscription (Enter to skip): gaming
To add metadata, rename or filters settings, please edit poca.xml

[ ~ ] poca --verbose
THE CRATE AND CROWBAR. 5
  CCEp214.mp3  [56 Mb]
  LGCEp004.mp3  [35 Mb]
  CCEp215.mp3  [61 Mb]
 ...
```

# CHAPTER 5

# Configuration

The poca.xml file contains all poca's general and subscription specific configuration. To get started simply run poca once to get a basic config file placed in ~/.poca.

The configuration file is divided into three main parts:

```
<poca version="1.0">
  <settings>
  ...
  </settings>
  <defaults>
  ...
  </defaults>
  <subscriptions>
  ...
  </subscriptions>
</poca>
```

- **Settings**: Contains general settings
- **Subscriptions**: Contains a list of all the podcasts you want to subscribe to
- **Defaults**: Options in defaults are similar to those in subscriptions, only they apply globally (unless overridden).

## 5.1 Settings

In poca settings are general, non-podcast specific options. For all podcast specific options, see *Subscriptions*.

**Contents**

### 5.1.1 XML

When entering text values into poca.xml settings elements, as in

```
<setting>text_value</setting>
```

you can use any unicode character directly without resorting to escapes or code point values. I.e. this is valid:

```
<setting> and </setting>
```

The only exceptions to this rule are

- < or less than - to insert, use the code `&lt;`
- `&` or ampersand - to insert, use the code `&amp;`

E.g. to use an ampersand instead of *and* in the above example, do

```
<setting> &amp; </setting>
```

### 5.1.2 Structure

The `<settings>` element contains the following required settings:

- `basedir`

and the following optional settings:

- `id3v2version`
- `id3removev1`
- `filenames`
- `useragent`
- `email`

### 5.1.3 Required settings

#### base_dir

The directory that will contain the audio files. Under this directory poca will create a subdirectory for each subscription:

```
base_dir
    |--- my_podcast
    |        |--- ep01.mp3
    |        |--- ep02.mp3
    |        |--- ...
    |--- my_other_podcast
    |         |--- ...
etc.
```

At startup poca tests if `base_dir` exists and the user has write permissions to it. If it does not, poca tries to create it. Failure at this stage causes poca to quit with a failure message. Remember that base_dir must be a legal path on the filesystem used.

### 5.1.4 Optional settings (id3)

Poca allows for modifying the metadata of the downloaded files by setting override values. Within the 'settings' context two options allow you to choose what format poca should use when writing id3 headers. There are no settings needed for writing vorbis comments in ogg, flac, etc.

#### id3v2version

`id3v2version` accepts `3` or `4` as settings, representing id3v2.3 and id3v2.4 respectively. Historically, Windows and some hardware players have preferred v2.3 as they cannot read the UTF-8 encoded characters used in v2.4. Default is `4`.

#### id3removev1

This allows you to remove id3v1 headers (if any exist) from the files. It has the valid values **yes** and **no**. It will only be applied in any given subscription if the subscription settings (or defaults) include id3 overrides.

### 5.1.5 Optional settings (other)

#### useragent

This is a fallback option, if you experience a lot of download failures. Some servers refuse downloads using Python's request library ("If bad guys are using hammers, ban hammers, right?"). Setting `useragent` allows poca to make a second attempt in case of failure due to blocking, only this time with a spoofed user agent. Enter any user agent string you want - search for "what's my useragent" and copy your browser's string. Leave it empty/remove it if you don't want poca to use spoofing. We suggest you leave as it is and only return to it if you experience persistent download failures.

### filenames (new in 1.1)

The filenames setting allows the user to set an upper level of filename character permissiveness. Some filesystems allow for more characters in filenames than others. Poca handles this by trying four different filenaming restriction settings, one after the other, in order of decreasing 'permissiveness'. In other words: The more attempts, the more characters are removed from the filename of the file being written. This system mostly comes into effect when using feed data to generate a filename using the rename scheme. The four levels are:

- `permissive`: Only `/` and (the null character) are removed from the filename. These are the only characters that are outright forbidden on linux file systems, like ext4 and others.
- `ntfs`: Any characters that are not acceptable on NTFS and FAT filesystems (mounted using VFAT, the restrictions are the same for FAT as for NTFS) are removed from the filename. That includes all control characters, slashes (backward and forward), colons, asterisks, question marks etc.
- `restrictive`: Unlike permissive and ntfs, restrictive is defined by the characters included, not by those removed. Accepted characters are alphanumerical, hyphens, and underscores. In regex terms: [**a-zA-Z0-9_**-]. Spaces are converted to underscores, rather than removed.
- `fallback`: This option is defined as the publishing date of the entry in the feed in the format YYYY-MM-DD, followed by 9 random hexadecimal digits.

`permissive` and `ntfs` both retain all (non-excluded) unicode characters. The filenames setting does not definitively determine the scheme, poca will use. It allows the user to set a 'lower' starting point than would otherwise be used. The default starting point is the `permissive` setting.

If the files are to be shared using a protocol, less tolerant of filename characters than the filesystem used, it might be preferable to have poca apply more restrictions from the start rather than having to rename files later. E.g. ext4 filesystem but the files are made accessible via SAMBA/CIFS.

The setting applies regardless of whether a subscription uses the default, original filenames or a rename scheme. It is applied only to the basename, and after a possible `rename` operation.

Note that the `fallback` setting will also be applied regardless of user settings, if poca detects that multiple entries to be downloaded will have the same filename if the configuration is followed to the letter. E.g. if no rename scheme is in effect with a subscription from acast.com (which names every file media.mp3) or if the user has chosen [subscription title].mp3 as the rename scheme.

### email

In order to properly enable email logging (`poca -e`) you will need a working email setup in your settings section. To keep things simple the email tag and its sub-tags will not appear in a standard auto-generated `poca.xml` file. You will need to add them manually.

Email logging works similar to file logging in that poca summarizes changes to a subscription rather than listing each new/deleted episode individually. At the end of a run Poca will send off one email (if there have been sufficient changes) or none if not much has changed. See `threshold` below for details.

The following settings are used:

- **only_errors**: If set to `yes`, only errors will be logged to email. This does not affect file logging. If set to `no` all the notifications you find in the file log will go into the log emails as well (episodes downloaded, removed, user deleted etc.). Default is `no`.
- **threshold**: The number of entries required before an email is sent. At the end of the run the number of logged entries is compared to the `threshold` value and if it is equal to or greater than that number, the logged entries are emailed off and the cache is cleared. Otherwise the entries are saved to the cache and included next time Poca runs. So a value of 1 means that a run that produces any entries (or errors if only_errors is set) will fire off an email. A run that produces 56 entries will likewise fire off a single email. A run that produces no entries will

not result in an email. Increasing this setting is mostly useful in combination with **only_errors**. By setting a threshold of say 20 or 30, you will get notified when your subscriptions consistently produce errors (e.g. a feed is not working anymore) but not (instantly) when a server is merely offline for an hour or a single request gets lost. Default is `1`.

- **fromaddr**: The sender address for the log emails. No default.
- **toaddr**: The recipient address for the log emails. No default.
- **host**: The email server's name/address. Default is `localhost`.
- **starttls**: To keep things simple Poca only accepts two kinds of setup: Either you relay without authentication on port 25 (probably only local servers) or you do submission with STARTTLS on port 587. Enter `yes` for the latter, `no` for the former. Choosing `yes` will require a `<password>` entry (see below), choosing `no` will not. Default is `no`.
- **password**: The password for the SMTP server. Only used with STARTTLS set to `yes`.

If you intend to use a Gmail account for this purpose, please be aware that you'll need to allow access for insecure apps in order for password authentication over STARTTLS to work.

### Examples

```
<email>
    <fromaddr>me@localhost</fromaddr>
    <toaddr>me@localhost</toaddr>
</email>
```

The very minimal configuration sends off one email for each run that produces any output. It relies on a local smtp server that accepts emails originating from the machine itself unquestioningly.

```
<email>
    <only_errors>yes</only_errors>
    <threshold>20</threshold>
    <host>smtp.gmail.com</host>
    <starttls>yes</starttls>
    <password>123456</password>
    <fromaddr>mypocainstance@gmail.com</fromaddr>
    <toaddr>me@hotmail.com</toaddr>
</email>
```

The full STARTTLS setup overriding defaults. It will only inform us of errors and when at least 20 have been logged, either in a single run or over multiple runs. It will authenticate over STARTTLS (the standard login/password way of authenticating when using any email client) and send your password over an encrypted channel to gmail.com.

## 5.2 Subscriptions

Subscriptions are podcast feeds you want poca to follow and (usually) download any new entries from. You create a subscription in poca by adding an xml `subscription` element with all the relevant settings to your `poca.xml` configuration file. Using the add command from *poca-subscribe* is a nice shortcut if you don't like editing xml.

**Contents**

### 5.2.1 Structure

Each `<subscription>` is created as an element under the `<subscriptions>` element. The structure of a subscription element is as follows:

```
<subscription category="..." state="...">
    <title>...</title>
    <url>...</url>
    <max_number>...</max_number>
    <from_the_top>...</from_the_top>
    <track_numbering>...</track_numbering>
    <metadata>
        <headerfield1>...</headerfield1>
        <headerfield2>...</headerfield2>
        <headerfield3>...</headerfield3>
    </metadata>
    <rename divider="..." space="...">
        <title/>
        <episode_title/>
        <date/>
        <uid/>
        <org_name/>
```

(continues on next page)

(continued from previous page)

```
    </rename>
    <filters>
        <filename>...</filename>
        <title>...</title>
        <weekday>...</weekday>
        <hour>...</hour>
        <after_date>...</after_date>
    </filters>
</subscription>
```

### 5.2.2 Required elements

`subscription` entries without the required elements are silently skipped during updates.

#### title

Title of show/subscription. Used only to name the subscription subdirectory under the **base_dir**, not to insert into ID3 headers.

Before attempting to download files for a subscription, poca checks to see if the subscription subdirectory exists and the user has permission to write to it. If not, poca attemps creation of the subdirectory. Failure at this stage causes poca to skip the subscription. The title must be a legal directory name on the filesystem used (linux filesystems have almost no restrictions, NTFS and FAT have a few).

#### url

The address of the RSS feed of the podcast.

That's all. If the number of optional settings are a bit overwhelming, you can simply leave it at that. `max_number` would probably be the only truly important non-required setting (see just below).

### 5.2.3 Optional elements

Note that all optional elements in a subscription can also be added to the `<defaults>` section to be applied globally.

#### max_number

The maximum number of episodes (integer) for the subscription to have at any one time. Only the `max_number` most recent episodes will be downloaded. If this is not set poca will - unless otherwise restricted - download all files in the feed. This setting replaces the `max_mb` setting in versions of poca prior to 0.6.

#### from_the_top

This is an alternative to default mode. The latter prioritises later episodes over newer: If `max_number` is 4, poca will download the latest 4 episodes. If from_the_top is set to 'yes', poca will instead start at the beginning, downloading the oldest 4 episodes. To move forward in the feed, you simply delete old files (episodes 1 and 2). Poca will detect this and fill up the `max_number` quota with newer files (episodes 5 and 6). This is useful for audiobook-style podcasts like *Serial* or working your way through old episodes of a newly discovered podcast.

### track_numbering

Track numbers in podcasts are hit or miss. Some include them, some don't. Most players will then simply play the files in the order of filenames. If you want to ensure that the episodes have track numbers set to `yes` which will overwrite the episodes track number with an artificial track number that starts at 1 with the first episode poca downloads (so if you start subscribing at episode 247, this will get tracknumber 1 etc.) Set to `if missing` to only insert track numbers when they are absent.

This can be a useful setting in `<defaults>`. Set to `no` or leave the option out of your subscription to leave the track number as is.

Note that track numbers can also be overwritten using the `tracknumber` element in metadata (see below). That, however only sets track numbers to a static value - or if no value is entered removes the track numbers entirely.

### metadata

With 1.1 tagging now works for m4a files as well as vorbis comments (ogg, opus, flac amongst others) and id3 tags.

Each element under **metadata** is a field in the id3/m4a/vorbis comment header that should be overwritten or added (if there is no such field in the original metadata). If you do not wish to touch the vorbis comment/id3 headers simply leave out the **metadata** element altogether.

- Example: The 'Savage Love' podcast has the publisher 'The Stranger' as the value of the 'artist' field. Creating an `<artist>Dan Savage</artist>` element tags the downloaded files with *Dan Savage* instead.
- Example: If your player arranges files by genre, it might be advantageous to have all podcasts genre labelled 'podcasts' in a uniform fashion by inserting a `<genre>podcast</genre>` element either in each individual subscription or in the `defaults`.

A list of all the field names that poca recognises for id3 and m4a headers can be printed by running `poca-subscribe tags` using either the `--mp3` or the `--mp4` flag.

Relying on mutagen's "easy" modules, poca allows you to use `title` for track title, `artist` for artist, etc. Vorbis comment tags are not restricted in what keys can be used (though all vorbis comment keys must be ascii) but this Xiph.org list can be used as a reference for tag names convention.

Any empty value, i.e.:

```
<album></album>
```

or:

```
<album/>
```

will cause the frame to be removed rather than overwritten. This especially true for the `<chapters/>` element, as that can only be used to remove. poca will disregard any text value associated with it and will only use it to remove CTOC and CHAP frames from id3 tags in the subscription.

### rename

An option to rename the media files downloaded. Not all feeds name their media consistently, helpfully (e.g. all files are named media.mp3) or alphabetically (e.g. just using the episode title rather than conventions like padded-tracknumber_title).

Renaming is done by slotting in the new name components as XML elements in the file name order desired. The available components from the feed and the user settings are as follows.

- `title`: The title of the subscription as indicated by the user

- `episode_title`: The title of the episode as indicated by the publisher in the feed
- `date`: The date at which the episode was published to the feed (aka pubdate). The date is rendered in the format YYYY-MM-DD.
- `uid`: The episode's unique identifier in the feed. This can be a number (`7932`), a random string (`d39gs9db3f6ihhbzx5`) or the url for the episode. All non-alphanumerical characters are discarded for naming purposes.
- `org_name`: The original filename for the episode. In case you just want to preface it with a date or uid.

Each component can be used as many times as desired or not at all. The components can come in any order desired.

Please note, that the rename pattern is resolved before the file is downloaded. Therefore there is no option to make use of media metadata (id3/vorbis comments) when renaming. Any non-available components (e.g. a feed does not have uids for entries) will be replaced with 'missing' or similar.

By default the components are divided by underscores when assembling the full file name. Alternatively you can set the `divider` attribute and give it the desired divider value. Spaces in the filename originating from using a title or similar can be replaced by using the `space` attribute. Spaces will be replaced with the value of the attribute. Note that the resulting filenames are sanitized, see the *filenames* section in **Settings**.

Example:

```
<rename divider="_" space="_">
    <date/>
    <org_name/>
</rename>
```

The above configuration as applied to the Python Bytes feed result in files named like this:

```
2017-10-25_your-technical-skills-are-obsolete-now-what.mp3
2017-11-02_bundling-shipping-and-protecting-python-applications.mp3
```

### filters

The filters element should contains one or more of the following tags that filter the entries in the feed based on various criteria. All filters are positive in the sense that the entry must meet the criterion to be included. Each filter can only be used once per subscription.

### filename

The filename of the entry must match this string in order to be included. Note that the value is interpreted as a regex, so certain characters should be escaped (e.g. a literal point should be written '.') Apart from this it is perfectly possible to use simple strings and ignore the regex aspect. The filename matched is the original filename, not those resulting from using `rename` (see above). Example:

`<filename>^episode</filename>` will only include regular Judge John Hodgman episodes and not the special cheese shows, *shootin' the bries* that have filenames starting with "shootin'".

### title

The same as above, only for the title in the RSS feed (not in the music file's metadata). Example:

`<title>Wires</title>` only gets the 'Within the Wires' episodes from the Welcome to Nightvale feed.

#### hour

The hour (24h-format) at which the entry was published. This is useful for podcasts that put out more episodes a day than you need, e.g. news broadcasts. Example:

`<hour>21</hour>` only gives you the evening edition of BBC's Newshour.

#### weekdays

Excludes all episodes not published on the specified weekdays. Each weekday to be included is written as a single digit where Monday is 0, Tuesday is 1, etc, up to 6 for Sunday. Example

`<weekdays>024</weekdays>` to get Monday, Wednesday, and Friday episodes.

#### after_date

Only includes episodes published on or at a later time than the specified date. Format is YYYY-MM-DD. This is useful is you don't want to restrict the `max_number` of the subscription but still avoid downloading the entire back catalogue. Example:

`<after_date>2016-08-23</after_date>` will ignore all episodes published before August 23rd 2016.

### 5.2.4 Optional attributes

Each subscription tag can have two optional attributes:

#### category

Setting a category will sort the outputted list of poca-subscribe's `list` command into groups, each category being grouped together. Example:

`<subscription category="news">...</subscription>`

#### state

The state attribute has two valid settings: `active` and `inactive`. If a subscription does not have the attribute it is counted as being active. Active subscriptions are updated as normal. Inactive subscriptions are passed over when poca is run. Additionally, setting a subscription's state to inactive using poca-subscribe's `toggle` command will delete old audio files. Example:

`<subscription state="inactive">...</subscription>`

### 5.2.5 Example

Here is an example of a subscription to a news in French podcast:

```
<subscription category="news">
    <title>francais facile</title>
    <url>http://www.rfi.fr/radiofr/podcast/journalFrancaisFacile.xml</url>
    <max_number>3</max_number>
    <metadata>
```

(continues on next page)

(continued from previous page)

```
        <artist>Radio France Internationale</artist>
        <album>Journal en français facile</album>
        <genre>podcast</genre>
    </metadata>
    <rename>
        <title/>
        <date/>
    </rename>
</subscription>
```

# CHAPTER 6

# Running

Poca is run by issuing the poca command. Subscriptions can be managed by using the helper script poca-subscribe.

## 6.1 poca

### 6.1.1 Help

```
usage: poca [-h] [-q | -v] [-l] [-e] [-c CONFIG] [-g GLYPHS] [-t THREADS]

Poca 1.1 : A fast and highly customizable command line podcast client

optional arguments:
  -h, --help            show this help message and exit
  -q, --quiet           No output to stdout (useful for cron jobs)
  -v, --verbose         Output details on files being added and removed.
  -l, --logfile         Output to file in poca config directory
  -e, --email           Output to email (set in config)
  -c CONFIG, --config CONFIG
                        Use alternate config directory
  -g GLYPHS, --glyphs GLYPHS
                        Glyph set to use. Options: default, ascii, wsl, and
                        emoji
  -t THREADS, --threads THREADS
                        Number of concurrent downloads to allow. '--verbose'
                        forces single thread.
```

### 6.1.2 Output

In normal and verbose mode poca summarizes operations for each subscription. The glyphs should be interpreted as follows:

- signifies the number of user-deleted files detected
- signifies the number of files to be removed due to to hitting the `max_number` cap
- signifies the number of files to be downloaded

In verbose mode poca will also generate output for each individual operation. The glyphs should be interpreted as follows:

- indicates deletion of a file
- indicates download of a file
- indicates an error

Using the -g parameter these can be changed to other glyph sets, including WSL-friendly and ASCII-only.

## 6.2 poca-subscribe

### 6.2.1 Help

```
usage: poca-subscribe [-h] [-c CONFIG]
                      {add,list,tags,delete,toggle,stats} ...

poca-subscribe 1.1 : A subscription management tool for poca

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG, --config CONFIG
                        Use alternate config directory

commands:
  'poca-subscribe command --help' for futher information

  {add,list,tags,delete,toggle,stats}
    add                 Add a new subscription interactively
    list                List current subscriptions
    tags                List available id3 tags
    delete              Remove subscription, delete files
    toggle              Set state of current subscriptions
    stats               Get feed stats for current subscriptions
```

Each command also has it's own set of flags/arguments. Run `poca-subscribe [command] --help` to see them.

### 6.2.2 Commands

#### add

Add a subscription. `add` starts up a step-by-step setup assistant that allows you to enter a title, url, max_number setting etc. Once completed, the resulting subscription is added to your poca.xml. Does not at the moment allow for metadata and filters settings to be added.

This is an easy-to-use way to quickly add some subscriptions if you're not entirely comfortable editing xml.

### stats

Get some basic stats on a current subscription. Use -t [title] for matching against the subscription title, -u [url] for matching against the url (e.g. check all subs from the BBC). If no matching is done it will loop through all current subs. Prints:

- publishing frequency over the past five weeks (how many episodes and on what weekdays they publish)
- latest episode title and date
- average size (Mb) and length (hours:minutes) of an episode

`stats` is automatically called when using `add` after entering a url.

Example:

```
PYTHON BYTES
Author: Michael Kennedy                                  PUBLISHED / 5 WEEKS
Title:  Python Bytes

Last episode: #29 Responsive Bar Charts with Bokeh
Published:    08 Jun 2017

Avg. size of episode:   15 Mb
Avg. length of episode: 20m                               M  T  W  T  F  S  S
```

### delete

Delete a subscription. Goes through all subscriptions and asks you to decide on delete/keep. List can be narrowed using the command-specific `-t/--title TITLE` or `-u/--url URL` parameters.

### toggle

Toggle the *state* attribute of a subscription between active and inactive. Goes through all subscriptions and asks you to decide on setting as active or inactive. Inactive subscriptions are not updated. Subscriptions with no state attribute are considered active. List can be narrowed using the command-specific `-t/--title TITLE` or `-u/--url URL` parameters.

### list

Prints out a list of subscriptions. List is sorted by category attribute if any are present.

### tags

Prints out all useable tags (use grep/less) for either mp3 or mp4 (use -3/-4 flags to get the relevant list).

Poca uses Mutagen's EasyID3 editor so it understands sensible tag names, i.e.

Don't:

```
<metadata>
    <TALB>Call of the Wild<TALB>
</metadata>
```

Do:

```
<metadata>
    <album>Call of the Wild<album>
</metadata>
```

Ogg, FLAC et al. use VorbisComment with no restrictions on tag names (though there are conventions).

CHAPTER 7

# Changelog

## 7.1 Version 1.1

### 7.1.1 Features

- Filenames: Auto-detect and fix filename collisions (e.g. if all files are named media.mp3)
- Metadata: Tagging now extends to mp4 files (as well as mp3 and files using vorbis comments for metadata)
- Metadata: Can now use 'comment' field to overwrite comments on ID3 tags (all existing comment frames are deleted)
- Metadata: Empty values in metadata elements (e.g. `<album></album>` or `<album/>`) can now be used remove the frame entirely.
- Metadata: `<chapters></chapters>` is a special case of that. Adding the chapters element can be used to remove the table of contents and chapters frames (CTOC and CHAP). It will disregard any text value inserted.

### 7.1.2 Fixes

- Filenames: Filesystem safe names. poca will automatically adapt the default or chosen filename to create legal filenames on the filesystem used for the podcast files (e.g. remove question marks from filenames on NTFS). It will not by default 'degrade' the filename more than needed for any given file system (see below).
- Filenames: New setting to manually set filename permissiveness: Override the default behaviour by ordering poca to start at a lower level of filename character permissiveness.
- Config: File permissions check is more granular. Checks out as long as user has read permission to poca.xml and write permission to db and media folder. Note: A read-only config file prevents poca from updating the url if status == 301.
- Filters: Filename filtering has been fixed
- Filters: Filtering errors are now reported

- Download: Cover images are downloaded once, and only downloaded again if there is a new one, rather than with each update.
- Improved unicode support (better testing for support and fallbacks)

## 7.2 Version 1.0

- Added file renaming options (#16)
- More consistent output in normal and verbose mode
- audiosearch api removed again due to service closure (#87)
- Bugfixes, quibbles and niggles aplenty

## 7.3 Version 0.9

- Artificial track numbers (#43)
- Support for tagging with other formats than mp3: Ogg, opus, mp4, flac, … (#18)
- Reintroduced support for id3v2.3
- poca-subscribe search: Seach for shows with audiosearch's api
- 'Preview' feed in poca-subscribe's add command (#55)
- Multithreading support with option for concurrent downloads as well as concurrent updates (#45)
- Subscription URLs are automatically updated when feeds move (HTTP status code 301)
- New dependency: `requests` for downloads due to downloading in threads

## 7.4 Version 0.8

- New dependency: `lxml` for greatly simplified configuration parsing
- System defaults in case of missing or bad config and user set global subscription defaults
- New script: `poca-subscribe`, a cli tool to manage subscriptions

## 7.5 Version 0.7

- Log: Terse logging (skip writing about subscription if nothing has changed) (#32)
- Email logging: A sensible email logging system (#26)
- Style: Investigating Syntastic complaints (#39)
- Bug fix: Hangs on feeds without entry size info (#40)
- Bug fix: Crash on entries without enclosures (#41)

## 7.6 Version 0.6

- max_number: Limit by number of files (#14)
- max_mb has been dropped in favour of max_number
- filters: Filter entries by filename, date, … (#29)
- from_the_top: Option to start podcast from the beginning (#28)
- Download cover image from feed (#25)
- Testing for unicode exceptions in feed treatment and mp3 metadata (#17)
- Spoofed useragent introduced as fallback if urllib is denied

## 7.7 Version 0.5

- Completed port to Python 3
- Completely revised and simplified stream and log output
- Mp3 tagging reimplemented
- New download function with proper timeouts

## 7.8 Version 0.4

- Reduced functionality port to Python3

# CHAPTER 8

# Links

- Homepage: https://projects.brokkr.net/projects/poca
- Source code: https://github.com/brokkr/poca
- Python Package Index (pypi): https://pypi.python.org/pypi/poca
- Documentation: https://poca.readthedocs.io/en/latest/