
ArchMap Documentation

Release

James Fawcus-Robinson, Johannes L othberg

Dec 16, 2017

Contents

1	About	3
1.1	ArchMap	3
1.2	Installation	3
1.3	Synopsis	4
1.4	Use	4
1.5	License	4
1.6	External links	5
2	Install	7
2.1	System Requirements	7
2.2	How-to	7
2.3	Support	8
2.4	Release Notes	8
3	Use	9
3.1	Examples	9
3.2	Use the code	10
4	Contribute	13
4.1	Roadmap	13
4.2	Contributing	13
4.3	Development	14
5	Indices and tables	17

The ArchMap project creates a map of Arch Linux users all over the world.

In this section:

- *ArchMap*
- *Installation*
- *Synopsis*
- *Use*
- *License*
- *External links*

1.1 ArchMap

archmap generates *GeoJSON* and *KML* files which can be used to display a map of Arch Linux users, it does this by parsing data from the [ArchWiki](#).

Have a look at the [ArchMap](#) page on the ArchWiki for more information about this project.

The documentation is hosted by [readthedocs.org](#).

1.2 Installation

You can install `archmap` from [PyPi](#) by running `pip3 install archmap` or from the [AUR](#) by installing the `archmap-git` package.

1.3 Synopsis

By default, running `archmap` will output four files to `/tmp`; `archmap.txt`, `archmap.geojson`, `archmap.kml` and `archmap.csv`. This can be overridden by either using the config file or by the command line switches.

The config file should be placed in `/etc/archmap.conf`, this can be overridden by using `--config <path-to-config-file>`

1.4 Use

Running `archmap --help` will display this help message:

```
usage:
archmap [-h] [-v] [-q] [--config FILE] [--url URL] [--file FILE] [--pretty] [--text_
↪FILE] [--geojson FILE] [--kml FILE] [--csv FILE]

optional arguments:
-h, --help            show this help message and exit
-v, --verbose         Show info messages
-q, --quiet           Disable info messages
--config FILE        Use an alternative configuration file instead of /etc/archmap.conf
--url URL             Use an alternative URL to parse the wiki list from
--file FILE          Use a file to parse the wiki list from
--pretty             Prettify the text user list. Only works if user output is enabled
--text FILE         Output the raw-text to FILE, use 'no' to disable output or '-' to_
↪print to stdout
--geojson FILE      Output the GeoJSON to FILE, use 'no' to disable output or '-' to_
↪print to stdout
--kml FILE          Output the KML to FILE, use 'no' to disable output or '-' to print to_
↪stdout
--csv FILE          Output the CSV to FILE, use 'no' to disable output or '-' to print to_
↪stdout
```

1.5 License

Everything in the [ArchMap repo](#) is unlicensed.

All of the files that this script can generate (`archmap.txt`, `archmap.geojson`, `archmap.kml`, and `archmap.csv`) will contain text from the [ArchWiki](#) which puts them under the [GNU Free Documentation License 1.3](#) or later.

1.5.1 Unlicense

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org/>

1.6 External links

1.6.1 Protect info

- Project page on the ArchWiki - <https://wiki.archlinux.org/index.php/ArchMap>
- Project talk on the Arch forums - <https://bbs.archlinux.org/viewtopic.php?id=22518>
- Arch Women wiki page - <https://archwomen.org/wiki/aw-tech:archmap>

1.6.2 Data

- The list of user data - <https://wiki.archlinux.org/index.php/ArchMap/List>
- Pre-generated *GeoJSON*, *KML* and *CSV* files - <https://archwomen.org/media/archmap/>

1.6.3 Code

- Repository on GitHub - <https://github.com/guyfawcus/ArchMap>
- Stats on Ohloh - <https://www.ohloh.net/p/ArchMap>
- Code search on sourcegraph.com <https://sourcegraph.com/github.com/guyfawcus/ArchMap>

In this section:

- *System Requirements*
- *How-to*
- *Support*
- *Release Notes*

2.1 System Requirements

Python 3.4 - If your running Arch, this shouldn't be a problem!

- `geojson`
- `simplekml`

2.2 How-to

See also: *Packaging*

2.2.1 Manual git install

This will make a directory `ArchMap/` with a link to `archmap` in it. It uses *git* and *pip* to resolve the dependencies.

Download `this` script:

```
echo -e "==> Make and cd to ./ArchMap\n"
mkdir ArchMap && cd ArchMap

echo -e "\n\n==> Download the ArchMap repo from GitHub\n"
git clone https://github.com/guyfawcus/ArchMap.git ArchMap-git

echo -e "\n\n==> Install the required packages\n"
pip3 install -r ArchMap-git/requirements.txt

echo -e "\n\n==> Make an easy link to archmap.py\n"
cd ../ && ln -s ArchMap-git/archmap.py ./archmap

echo -e "\n\n==> Test by printing the help message\n"
./archmap --help
```

2.3 Support

External links

2.4 Release Notes

See also: [Releases on GitHub](#)

If you would just like to get up and running, have a look at:

3.1 Examples

All of these example assume you have installed **archmap** via pacman or pip, if you would like to use the script directly, just use `./archmap.py`.

3.1.1 Help

The **-help** flag will output a help message with all of the available options:

```
archmap --help
```

3.1.2 Basic use

By default, running **archmap** will output three files to `/tmp`, **archmap.txt**, **archmap.geojson** and **archmap.kml**, this can be overridden by either using the config file or by the following command line switches.

Using the **-verbose** flag will print information on what the script is doing:

```
archmap --verbose
```

You can specify the output location for the user list text, GeoJSON, KML and CSV:

```
archmap --text /tmp/archmap.txt --geojson /tmp/archmap.geojson --kml /tmp/archmap.kml  
↪ --csv /tmp/archmap.csv
```

If you would like to parse an alternate copy of the wiki list, simply pass either the **-url** or **-file** flags:

```
archmap --url https://wiki.archlinux.org/index.php?title=ArchMap/List&oldid=131196
```

or

```
archmap --file "$HOME/Downloads/ArchMap_List - ArchWiki.html"
```

3.1.3 Logging

If the script is run on a system that uses systemd, it will log to it using the syslog identifier - "archmap".

You can review all logs generated by **archmap** by using:

```
journalctl SYSLOG_IDENTIFIER=archmap
```

If you would like to use any of this code, have a look at:

3.2 Use the code

3.2.1 Getting and parsing user data

```
archmap.get_users (url='https://wiki.archlinux.org/index.php/ArchMap/List', local="")
```

This function parses the list of users from the ArchWiki and returns it as a string.

Parameters

- **url** (*str*) – Link to a URL that points to a ArchWiki ArchMap list (default)
- **local** (*str*) – Path to a local copy of the ArchWiki ArchMap source

Returns The extracted raw-text list of users or None if not available

Return type *str* or *None*

```
archmap.parse_users (users)
```

This function parses the raw-text list (*users*) that has been extracted from the wiki page and splits it into a list of namedtuples containing the latitude, longitude, name and comment.

Parameters **users** (*str*) – raw-text list from the ArchWiki

Returns A list of namedtuples, each namedtuple has 4 elements: (latitude, longitude, name, comment)

Return type list of `collections.namedtuple (decimal.Decimal, decimal.Decimal, str, str)`

3.2.2 Output generators

```
archmap.make_text (parsed_users, output_file="", pretty=False)
```

This function reads the user data supplied by `parsed_users`, it then generates a raw-text list according to the formatting specifications on the wiki and writes it to `output_file`.

Parameters

- **parsed_users** (list of `collections.namedtuple (decimal.Decimal, decimal.Decimal, str, str)`) – A list of namedtuples, each namedtuple should have 4 elements: (latitude, longitude, name, comment)

- **output_file** (*str*) – Location to save the text output. If left empty, nothing will be output
- **pretty** (*bool*) – If set to True, the output "columns" will be aligned and expanded to match the longest element

Returns The text written to the output file

Return type *str*

`archmap.make_geojson` (*parsed_users*, *output_file=""*)

This function reads the user data supplied by `parsed_users`, it then generates GeoJSON output and writes it to `output_file`.

Parameters

- **parsed_users** (*list of collections.namedtuple (decimal.Decimal, decimal.Decimal, str, str)*) – A list of namedtuples, each namedtuple should have 4 elements: (*latitude*, *longitude*, *name*, *comment*)
- **output_file** (*str*) – Location to save the GeoJSON output. If left empty, nothing will be output

Returns The text written to the output file

Return type *str*

`archmap.make_kml` (*parsed_users*, *output_file=""*)

This function reads the user data supplied by `parsed_users`, it then generates KML output and writes it to `output_file`.

Parameters

- **parsed_users** (*list of collections.namedtuple (decimal.Decimal, decimal.Decimal, str, str)*) – A list of namedtuples, each namedtuple should have 4 elements: (*latitude*, *longitude*, *name*, *comment*)
- **output_file** (*str*) – Location to save the KML output. If left empty, nothing will be output

Returns The text written to the output file

Return type *str*

`archmap.make_csv` (*parsed_users*, *output_file=""*)

This function reads the user data supplied by `parsed_users`, it then generates CSV output and writes it to `output_file`.

Parameters

- **parsed_users** (*list of collections.namedtuple (decimal.Decimal, decimal.Decimal, str, str)*) – A list of namedtuples, each namedtuple should have 4 elements: (*latitude*, *longitude*, *name*, *comment*)
- **output_file** (*str*) – Location to save the CSV output. If left empty, nothing will be output

Returns The text written to the output file

Return type *str*

In this section:

- *Roadmap*
- *Contributing*
- *Development*
 - *System Requirements*
 - *Documentation*
 - *Testing*
 - *Packaging*

4.1 Roadmap

- Add more tests
- Work on packaging
- Use GitHub pages to build a homepage
 - Use [Leaflet](#) to get and display coords on a ...
 - [MapBox](#) map

4.2 Contributing

Contributions are always welcome! Here are a few ways you could contribute:

- Bug fixes
- New tests
- New features
- Testing on different platforms
- Documentation

Support: [External links](#)

4.3 Development

All of the following commands assume you are starting in the root ArchMap directory.

4.3.1 System Requirements

In addition to the *System Requirements* for the install, the following packages are required:

- To generate these docs:
 - sphinx
- For packaging:
 - setuptools
 - wheel (optional) - for building *wheels*

4.3.2 Documentation

Sphinx can be used to build a variety of formats.

First, make sure you're in the docs directory:

```
cd docs/
```

Make the preferred output:

```
make html
```

Open the the index page in your browser:

```
firefox _build/html/index.html
```

4.3.3 Testing

unittest is used for testing:

```
python setup.py test
```

This will search the `tests` directory for tests.

To check your commits before submitting, it is advisable to set up [pre-commit](#) first. Install it with:

```
pip3 install pre-commit
```

Then install the hooks so that they automatically run before each commit:

```
pre-commit install
```

Make sure the hooks are up to date:

```
pre-commit autoupdate
```

To run the hooks before a commit use:

```
pre-commit run --all-files
```

For further information, have a look at the [pre-commit advanced features](#) page or the `.pre-commit-config.yaml` config file to see what is run.

See also:

- [unittest - Python docs](#)

4.3.4 Packaging

ArchMap is currently packaged in two forms.

Arch Linux package

Packages are built using the `PKGBUILD` and `archmap.install` for settings.

To build package using the `PKGBUILD`:

```
cd pkgbuild
makepkg PKGBUILD
```

See also:

- [Creating packages](#)
- [Python Package Guidelines](#)

Python package

Packages are built using `setup.py` and `setup.cfg` for settings.

To build a [source distribution](#):

```
python3 setup.py sdist
```

To build a [wheel](#):

```
python3 setup.py bdist_wheel
```

See also:

- [Installation & Packaging Tutorial](#)

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

G

`get_users()` (in module `archmap`), 10

M

`make_csv()` (in module `archmap`), 11

`make_geojson()` (in module `archmap`), 11

`make_kml()` (in module `archmap`), 11

`make_text()` (in module `archmap`), 10

P

`parse_users()` (in module `archmap`), 10