
elevation Documentation

Release 1.0.1

B-Open Solutions srl

2017-03-28

Table of Contents

1	Quickstart	3
1.1	Installation	3
1.2	Command line usage	4
1.3	Command line reference	4
1.4	Python API	5
1.5	Project resources	5
1.6	Contributing	5
1.7	License	6
2	Design	7
2.1	Mission and vision	7
2.2	Software architecture	7
2.3	Version goals	8

Version 1.0.1

Date 2017-03-28

Easy access to global terrain digital elevation models, SRTM 30m DEM and SRTM 90m DEM.

If you have any feedback or you want to help out head over our main repository: <https://github.com/bopen/elevation>

Global geographic elevation data made easy. Elevation provides easy download, cache and access of the global datasets [SRTM 30m Global 1 arc second V003](#) elaborated by NASA and NGA hosted on [Amazon S3](#) and [SRTM 90m Digital Elevation Database v4.1](#) elaborated by CGIAR-CSI.

Note that any download policies of the respective providers apply.

Installation

Install the [latest version of Elevation](#) from the Python Package Index:

```
$ pip install elevation
```

The following dependencies need to be installed and working:

- [GNU make](#)
- [curl](#)
- [unzip](#)
- [gunzip](#)
- [GDAL command line tools](#)

The following command runs some basic checks and reports common issues:

```
$ eio selfcheck  
Your system is ready.
```

GNU make, curl and unzip come pre-installed with most operating systems. The best way to install GDAL command line tools varies across operating systems and distributions, please refer to the [GDAL install documentation](#).

Command line usage

Identify the geographic bounds of the area of interest and fetch the DEM with the `eio` command. For example to clip the SRTM 30m DEM of Rome, around 41.9N 12.5W, to the `Rome-30m-DEM.tif` file:

```
$ eio clip -o Rome-30m-DEM.tif --bounds 12.35 41.8 12.65 42
```

For the SRTM 90m DEM use:

```
$ eio --product SRTM3 clip -o Rome-90m-DEM.tif --bounds 12.35 41.8 12.65 42
```

The `--bounds` option accepts latitude and longitude coordinates (more precisely in geodetic coordinates in the WGS84 reference system EPSG:4326 for those who care) given as left bottom right top similarly to the `rio` command form `rasterio`.

If you have installed the packages `rasterio` and `fiona` you can clip a DEM on the same extent of any other geospatial data source supported by GDAL and OGR, for example if you have a georeference image `MyImage.tif` you can clip the corresponding DEM with:

```
$ eio clip -o MyImage-DEM.tif --reference MyImage.tif # enable with: $ pip install_
↳rasterio
```

The `--reference` option can take also vector data as input:

```
$ eio clip -o MyShapefile-DEM.tif --reference MyShapefile.shp # enable with: $ pip_
↳install fiona
```

The first time an area is accessed Elevation downloads the data tiles from the USGS or CGIAR-CSI servers and caches them in GeoTiff compressed formats, subsequent accesses to the same and nearby areas are much faster.

The `clip` sub-command doesn't allow automatic download of a large amount of DEM tiles, please refer to the upstream providers' websites to learn the preferred procedures for bulk download.

To clean up stale temporary files and fix the cache in the event of a server error use:

```
$ eio clean
```

Command line reference

The `eio` command as the following sub-commands and options:

```
$ Usage: eio [OPTIONS] COMMAND [ARGS]...

Options:
  --version            Show the version and exit.
  --product [SRTM1|SRTM3] DEM product choice. [default: SRTM1]
  --cache_dir DIRECTORY Root of the DEM cache folder. [default:
                        /Users/amici/Library/Caches/elevation]
  --help              Show this message and exit.

Commands:
  clean      Clean up the product cache from temporary files.
  clip      Clip the DEM to given bounds.
  distclean Remove the product cache entirely.
  info      Show info about the product cache.
```



```
seed      Seed the DEM to given bounds.
selfcheck Audit the system for common issues.
```

The `clip` sub-command:

```
$ eio clip --help
Usage: eio clip [OPTIONS]

Options:
  -o, --output PATH      Path to output file. Existing files will be
                          overwritten. [default: out.tif]
  --bounds FLOAT...      Output bounds in 'left bottom right top' order.
  -m, --margin TEXT      Decimal degree margin added to the bounds. Use '%' for
                          percent margin. [default: 0]
  -r, --reference TEXT   Use the extent of a reference GDAL/OGR data source as
                          output bounds.
  --help                 Show this message and exit.
```

Defaults can be defined by setting environment variables prefixed with `EIO`, e.g. `EIO_PRODUCT=SRTM3` and `EIO_CLIP_MARGIN=10%`.

Python API

Every command has a corresponding API function in the `elevation` module:

```
import elevation

# clip the SRTM1 30m DEM of Rome and save it to Rome-DEM.tif
elevation.clip(bounds=(12.35, 41.8, 12.65, 42), output='Rome-DEM.tif')

# clean up stale temporary files and fix the cache in the event of a server error
elevation.clean()
```

Project resources

Documentation	http://elevation.bopen.eu
Support	https://stackoverflow.com/search?q=python+elevation
Development	https://github.com/bopen/elevation
Download	https://pypi.python.org/pypi/elevation
Code quality	

Contributing

Contributions are very welcome. Please see the [CONTRIBUTING](#) document for the best way to help. If you encounter any problems, please file an issue along with a detailed description.

Authors:

- B-Open Solutions srl - @bopen - <http://bopen.eu>
- Alessandro Amici - @alexamici

License

Elevation is free and open source software distributed under the terms of the [Apache License, Version 2.0](#).

This chapter documents the high-level design of the product and it is intended for developers contributing to the project.

Note: Users of the product need not bother with the following. Unless they are curious :)

Mission and vision

The project mission is to enable easy management of global digital elevation data.

Target use cases:

1. access DEM data on-demand from well-known repositories
2. download and store efficiently elevation data on large areas

Project goals:

1. data download from well-known repositories
2. compact storage of local data

Software architecture

Logical components:

- the datasource Makefile
- the Python API
- the `eio` CLI

Version goals

This project strives to adhere to [semantic versioning](#).

1.1.0 (upcoming release)

To be defined.

1.0.0

Minimal set of features to be operationally useful. No completeness and no performance guarantees.

- Cache management:
 - new SRTM1 and old SRTM3 global high resolution digital elevation model
 - GNU Makefile for cache management (parallelism and dependency tracking)
 - GDAL VirtualRaster as main entry point