

ATLAS (AuTomedated Lidar Analysis Software) manual

Author(s): Nikolaos Siomos

Last Update: 11/23/22

Version: 0.1

Table of Contents

1. Installation.....3

2. Dependencies.....3

3. Execution.....4

1. Installation

ATLAS can be cloned through git. In order to do so:

- Git must be installed in the computer
- ATLAS can be cloned from the stable repository: <https://github.com/nikolaos-siomos/ATLAS>
- Open a terminal and cd to the folder you want to place ATLAS to
- run: `git clone https://github.com/nikolaos-siomos/ATLAS.git`

Alternatively you can just download ATLAS from the link above.

2. Dependencies

The current version of ATLAS has been developed and tested in python 3.9. The following python packages should be installed:

- numpy – version 1.24.0
- pandas – version 1.5.2
- xarray – version 2022.12.0
- netcdf4 – version 1.6.2
- scipy – version 1.9.3
- matplotlib – version 3.6.2
- h5netcdf – version 1.1.0

Note that other package versions (or higher versions) might also coincidentally work.

The recommended way to install the dependencies and run ATLAS is currently through an Anaconda environment (tested with Miniconda). The dependencies for the current version were collected with conda – version 4.12.0

An installation example follows for Linux systems (it should also work for Windows and MAC systems – if problems arise the users are encouraged to provide feedback to the EARLINET forum):

- Download and install the latest [Miniconda](#) for python 3.9
- Activate the **base** environment (usually it is auto-activated by default)
- Create a new environment to install the dependencies of ATLAS (let's call it **atlas_box** here):
 - `conda install -n atlas_box python=3.9`
- It is highly recommended to install **mamba** at this stage that greatly speed ups conda.

- `conda install -n base mamba`
- If mamba is not installed, please replace the word “mamba” with the word “conda in all commands below. Tests were run with mamba – version 0.15.3
- Install all dependencies using mamba in the **atlas_box** environment
 - `mamba install -n atlas_box numpy=1.24 pandas=1.5.2 xarray=2022.12.0 netcdf4=1.6.2 scipy=1.9.3 matplotlib=3.6.2 h5netcdf=1.1.0`
 - If one or more of the packages are not available in the specific version then try installing the version the package manager recommends e.g.:
 - `mamba install -n atlas_box numpy pandas xarray netcdf4 scipy matplotlib h5netcdf`
- Before running ATLAS make sure that the **atlas_box** environment is activated. It is best to close the terminal after installation of the dependencies and open a new one to execute ATLAS

3. Execution

Execution of ATLAS is quite straightforward given that the necessary input files (settings file, configuration file) are properly set.

Let's assume an exemplary folder structure to make the definition of paths a bit more clear:

- `/mydrive`
 - `/ATLAS`
 - `/program`
 - `__master__.py`
 - `/configuration`
 - `processing_options.ini`
 - `config_file.ini`
 - `lidar_data`
 - `drk` – dark measurement (optional for PollyXT systems)
 - `nrm` – normal measurement (must include a pol. Calibration for PollyXT systems)
 - `tlc` – telecover measurement
 - `north`
 - `east`
 - `south`

- west
- pcb (ignored for PollyXT systems – nrm folder is used instead)
 - +45
 - -45
- radiosondes

For command-line execution:

- Activate the **atlas_box** environment using:
 - `conda activate atlas_box`
- Get the absolute path to the `__master__.py` file, eg. `/mydrive/ATLAS/__master__.py`
- We will need at least the following arguments (a full list of available arguments can be found below):
 - `--parent_folder /mydrive/lidar_data/`
 - `--config_file /mydrive/configuration/config_file.ini`
 - `--settings_file /mydrive/configuration/processing_options.ini`
 - `--radiosonde_folder /mydrive/radiosondes`
- Run the following command:
 - `python /mydrive/ATLAS/__master__.py --parent_folder /mydrive/lidar_data/ --config_file /mydrive/configuration/config_file.ini --settings_file /mydrive/configuration/processing_options.ini --radiosonde_folder /mydrive/radiosondes`

It is possible to run ATLAS also from the Spyder IDE. For execution using Spyder:

- Press **Ctrl + F6** OR go to **Run --> Configuration per File**
- Activate **Run file with custom configuration**
- Tick the **Command line options**
- In the box add all arguments:
 - `--parent_folder /mydrive/lidar_data/ --config_file /mydrive/configuration/config_file.ini --settings_file /mydrive/configuration/processing_options.ini --radiosonde_folder /mydrive/radiosondes`
- Process using **F5** OR **Run File** like a normal python script

List of ATLAS arguments:

`-f` or `--parent_folder`

The path to the parent folder that contains the normal folder and all other optional input folders (dark, atmosphere, overlap). If no results folder is provided, it will be created here by default

`--dark_folder`

The path to the dark folder. Defaults to a drk folder inside the parent folder

`--rayleigh_folder`

The path to the rayleigh fit measurement folder. Defaults to a nrm folder inside the parent folder

`--telecover_sectors_folder`

The path to the telecover folder that contains the sector files. Defaults to a tlc folder inside the parent folder

`--telecover_rings_folder`

The path to the telecover folder that contains the ring (inner/outer) files – if any. Defaults to a tlc_rin folder inside the parent folder')

`--pcb_cal_p45_folder`

The path to the polarization calibration +45 folder. Defaults to a pcb/+45 folder inside the parent folder

`--pcb_cal_m45_folder`

The path to the polarization calibration -45 folder. Defaults to a pcb/-45 folder inside the parent folder

`--pcb_cal_stc_folder`

The path to the polarization calibration folder for a calibration with a single calibrator position. Defaults to a pcb/stc folder inside the parent folder

`--radiosonde_folder`

The path to the radiosonde folder. The radiosonde file that is closest to the measurement within 12h will be selected if more than 1 files are provided inside

`--converter_out`

The path to the folder where the converted netcdf files will be placed. This optional argument can be used if the folder must be placed out of the parent_folder. Defaults to parent_folder/netcdf

`--preprocessor_out`

The path to the folder where the preprocessed netcdf files will be placed. This optional argument can be used if the folder must be placed out of the parent_folder. Defaults to parent_folder/netcdf

--visualizer_out

The path to the folders where the plots and ascii folders will be created. This optional argument can be used if these two folders must be placed out of the parent_folder (default)

-c or --config_file

The path to the configuration file that contains the necessary metadata. This optional argument can be used if the configuration file must be placed out of the parent_folder (default)

-s or --settings_file

The path to the settings file that contains the processing options to run ATLAS. This optional argument can be used if the configuration file must be placed out of the parent_folder (default)