

Prisma Tool Installation Guide

FOREWORD	1
GEAR	2
INSTALLING PROCEDURE	2
1. PYTHON INSTALLATION:	2
2. THONNY INSTALLATION:	2
3. PACKAGES INSTALLATION WITH PIP:	2
4. USEFUL TO KNOW	4
5. CHECK PRISMA TOOL	4
PROCEDURE NEEDED TO WRITE GEOTIFF	4
6. INSTALL C++ LANGUAGE COMPILER	4
7. GDAL INSTALLATION	4
CONCLUSIONS:	5
CONTACT DETAILS FOR BUG REPORTS:	6

Foreword

This guide has been realized to help with the installation process of different software and components needed to run “*Prisma Tool*” (PT). PT is a python script useful to read the HDF5 files of PRISMA products.

“PRISMA” is an Italian hyperspectral mission for Earth observation (EO). PRISMA satellite, has been launched by the Italian Space Agency (ASI) and is in orbit since March the 22nd 2019. It can provide free hyperspectral and panchromatic images. It has a high spectral resolution spectrometer integrated with a medium resolution panchromatic camera. It records the radiation reflected from the Earth surface in VNIR (400 – 1010 nm) and SWIR (920 – 2505 nm) with 66 and 174 spectral bands, plus PAN (400 – 700 nm).

PRISMA product are free available at <https://ioccg.cmail20.com/t/t-l-pjylldt-nlkjllkh-n/>

PRISMA images are provided at different levels depending on the processing of the product:

- **L1** product are Top of atmosphere (TOA) Radiance images, radiometrically corrected and calibrated in physical units.
- **L2b** product is obtained by L1 product, images are geolocated at Bottom-of-Atmosphere Radiance.
- **L2c** product is geolocated at Bottom-of-Atmosphere Reflectance. Angstrom correction is applied.
- **L2d** product is a geocoded version of L2c, images are orthorectified by ASI using a DEM.

N.B.: in case of geolocated product it is possible to verify with PT the coordinates provided by ASI. PT generate a .txt file called “informazioni” with the ASI coordinates and generate the GeoTiff images with those coordinates. If ASI coordinates are not correct it is possible, running PT, to insert manually our own coordinates. PT, after the selection of HDF5 file, asks if you want to go ahead with the ASI coordinates or with your coordinates. In this case you can insert them pressing “1” and then enter on keyboard.

(Please Note that in case of L2d products, if ASI coordinates are wrong, you have to presume that also the corrections implemented using DEM level are not valid because of the uncorrected overlapping. In this case you can use for example Nasa DEM to orthorectify by yourself the L2d product.)

PRISMA images are provided with a hierarchical format called **HDF5**. It is a tree format in which data and metadata are organized in a logical way and it is possible to add or read just some branches that contains data and textual information. PT gives the possibility to export the HDF5 file in a format, like **GeoTiff**, that simplify the uploading and visualization in the GIS environment. Maybe the best solution would have been to write the script in C, but we have preferred python 3.

Gear

Prisma Tool is a python script so, first of all, you have to install **Python 3** and a software (e.g., **Thonny**) that helps to run and edit the code. This is not essential but very useful. In addition, some **libraries** are required to read data, save the metadata in a .txt file, read and save the tree of the HDF5 file in a .txt file. For example, you need: **h5py** to read the HDF5 file (need to install); **os** to create directories and files (already available in Python); **tkinter** to open windows (already available in Python); **numpy** to activate mathematical operations (need to install); and others. Libraries can be installed with **PIP** module (this is cleared in the next pages) writing in CMD line: `C:\>python -m pip install library name`

However, those libraries don't give the possibility to generate GeoTiff files, this goal is reached by installing **GDAL** library. To run GDAL, managed by rasterio in Python, is necessary to install Windows Visual Studio.

Installing procedure

1. Python installation:

- 1.1. From python.org/downloads download and install python interpreter for windows 64bit.
- 1.2. If you want, inside the "Extra components to install" folder there is the .exe file previously downloaded "python-3.11.0-amd64" which is useful to install python 3.11.0 version.

2. Thonny installation:

- 2.1. From thonny.org download and install the editing software *Thonny*. You have to download the "windows" version by clicking on "Installer with 64-bit Python 3.10" shown in the next image.
- 2.2. If you want, inside the "Extra components to install" folder there is the .exe file previously downloaded "thonny-4.0.1".
- 2.3. When installation has completed, you have to open *Thonny* and select the correct Python¹ interpreter to use by clicking on the drop menu located down on the right side. The correct Python is the one installed in step 1. Usually the path is:

`C:\Users\UserName\AppData\Local\Programs\Python\Python311\python.exe`

3. Packages installation with pip:

- 3.1. Click search icon in the Windows taskbar and type "CMD" in order to open the command prompt.



¹ **Computer has "different Python"**: for example, Thonny has its own Python, Qgis has its in own python, etc. We have to choose the Python installed in step 1, because on it you're going to install all the packages needed.

3.2. On the CMD line write the follow and then Enter: `python -m pip --upgrade pip`

- If you can't find python on the CMD line, try one of the next three solutions:
 - Type `py` instead `python` in the CMD script
 - Open windows settings -> search manage app execution -> turn "on" both python.exe and python3.exe
 - Add python path in the environment variables by using the procedure described in step 8.2.

3.3. When upgrading is terminated you can start with the **installation of packages**. Libraries must be installed one at a time, so you have to wait until the installation is complete before starting with the next package.

3.3.1. On the CMD line write the follow and then Enter: `python -m pip install numpy`

3.3.2. On the CMD line write the follow and then Enter: `python -m pip install matplotlib`

3.3.3. On the CMD line write the follow and then Enter: `python -m pip install pyproj`

3.3.4. On the CMD line write the follow and then Enter: `python -m pip install rasterio`

3.3.5. On the CMD line write the follow and then Enter: `python -m pip install h5py`

If the installation of one of those libraries doesn't work, for example rasterio, you can try this way:

3.4. From <https://www.lfd.uci.edu/~gohlke/pythonlibs> you can download the .whl file of library you want to install, in this case rasterio, suitable for your computer machine. If you want, inside the "Extra components to install" folder there is the .whl file previously downloaded for python 3.11 on windows amd 64 machine, called "rasterio-1.2.10-cp311-cp311-win_amd64.whl".

Carry out the following steps in order to complete the installation:

- right click + shift inside the folder that contains the .whl file -> click on "open PowerShell window here" in the pop up menu
- On the CMD line write the follow and then Enter `pip install filename.whl`
- If doesn't work, try with `python -m pip install filename.whl`
or `py -m pip install filename.whl`

3.5. If the installation won't work not even that way, you can follow the procedure described at <https://rasterio.readthedocs.io/en/latest/installation.html>, but you have to take in count that the instructions refers to an old version.

3.6. Another useful guide, for example referred to h5py installation, is provided by Bobby Hadz at <https://bobbyhadz.com/blog/python-failed-building-wheel-for-h5py> search "download and install a .whl file if you are on Windows"

N.B.: *AMD64* usually is an expression referred to 64bit processor, both the ones made by AMD and those build by Intel. Sometimes this kind of processor is called *x86-64* instead *AMD64*. But meaning is not the same if we are talking about programs created for AMD or Intel processor.

3.7. Installation is now completed, however is important to check if the libraries list includes also `os` and `tkinter` packages (You will be able to do that in the next step). If they are not included, you can follow the same procedure described above:

3.7.1. On the CMD line write the follow and then Enter: `python -m pip install os`

3.7.2. On the CMD line write the follow and then Enter: `python -m pip install tkinter`

4. Useful to know

- 4.1. To check which are the modules installed on your python, on the CMD line write the follow and then Enter: `python -m pip list`
- 4.2. For the full list of `pip` options, you can run `pip --help` in the terminal and the command will return the usage information.
- 4.3. In order to close the CMD window you can write `exit()` and press Enter.

5. Check Prisma Tool

- 5.1. Copy the file called "**Prisma_Tool_xx**" in a folder (we recommend on your desktop or somewhere on the C unit, because the execution of the script will be faster that way).
- 5.2. Try to run the script following the next steps: open *Thonny* -> File -> Open -> select "Prisma_Tool_xx" -> run -> run current script.
- 5.3. After that a popup window will open that allows you to select the HDF5 file. Select the file and press Open.
- 5.4. Now you could have two different situations:
 - 5.4.1. In case of **L1 product** of PRISMA, images must be georeferenced, so you need to go ahead with this guide and install GDAL. Running the script with this kind of product will return some errors.
 - 5.4.2. In case of **L2d product** of PRISMA, images are already geocoded, so you won't need GDAL library. The PT will create a new folder, located in the same path of HDF5 file, named with the specification of the product `GeoTIFF_L2D_yyyy_mm_dd`. The new folder contains both and GeoTiff files (one for each band) and some .txt file (tree file, metadata file, information file). During the running, the script will open a popup window like the one shown at the end of this guide. You have just to close it to continue the execution.
- 5.5. Anyway, we recommend you installing all the components, also the ones described in the next steps. In order to have a complete version of the Prisma Tool.

Procedure needed to write GeoTiff

6. Install C++ language compiler

- 6.1. From visualstudio.microsoft.com/it/downloads select and download *Build Tools for Visual Studio 2022 (Community)*. If you want inside the "Extra components to install" folder there is the .exe file previously downloaded "VisualStudioSetup".
- 6.2. Upon launching the installation you have to choose which tools you want to install and you have to select them. For example, we have selected:
 - Web & Cloud: python development
 - Desktop: development of applications with C ++
 - Other tool sets: data processing and storage
 - Other tool sets: analytical applications and scientific data analysis

7. GDAL installation

- 7.1. Dal sito <http://www.gisinternals.com/query.html?content=filelist&file=release-1911-x64-gdal-3-0-0-mapserver-7-4-0.zip> scaricare i file adatti alla macchina e alla versione di Python utilizzata. Es.:
 - gdal-300-1911-x64-core.msi

Generic installer for the GDAL core components

- GDAL-3.0.0.win-amd64-py3.7.msi
Installer for the GDAL python bindings (requires installing the GDAL core)

Alternatively, the .msi files already downloaded are in the 'Extra components to install' folder.

- Select the file gdal-300-1911-x64-core.msi, double-click + run.
 - Select GDAL-3.0.win-amd64-py3.7.msi, double-click + run. (The Python path must be specified during the installation process).
-

Note: At this point, certain paths need to be copied to the Windows 'environment variables'. This allows the system to find the installed components. This can be done by following the procedure below, checking that the paths specified match those created when GDAL was installed.

7.2. How to add paths to the Windows environment variable. A good guide can be found in Step 3: *How to add path variables* at: <https://sandbox.idre.ucla.edu/sandbox/tutorials/installing-gdal-for-windows>

- Procedure: Open Windows Explorer > right click > properties > advanced system settings > environment variables
- In the 'System variables' box, double click on 'path' and in the window that opens, add C:\Program Files\GDAL at the end.
- In the 'System variables' box, click on 'new' and type
NAME: GDAL_DATA
VALUE: C:\Program Files\GDAL\gdal_data
- In the 'System variables' box, click on 'new' and write
NAME: GDAL_DRIVER_PATH
VALUE: C:\Program Files\GDAL\gdalplugins
- Press OK and finish the operation.

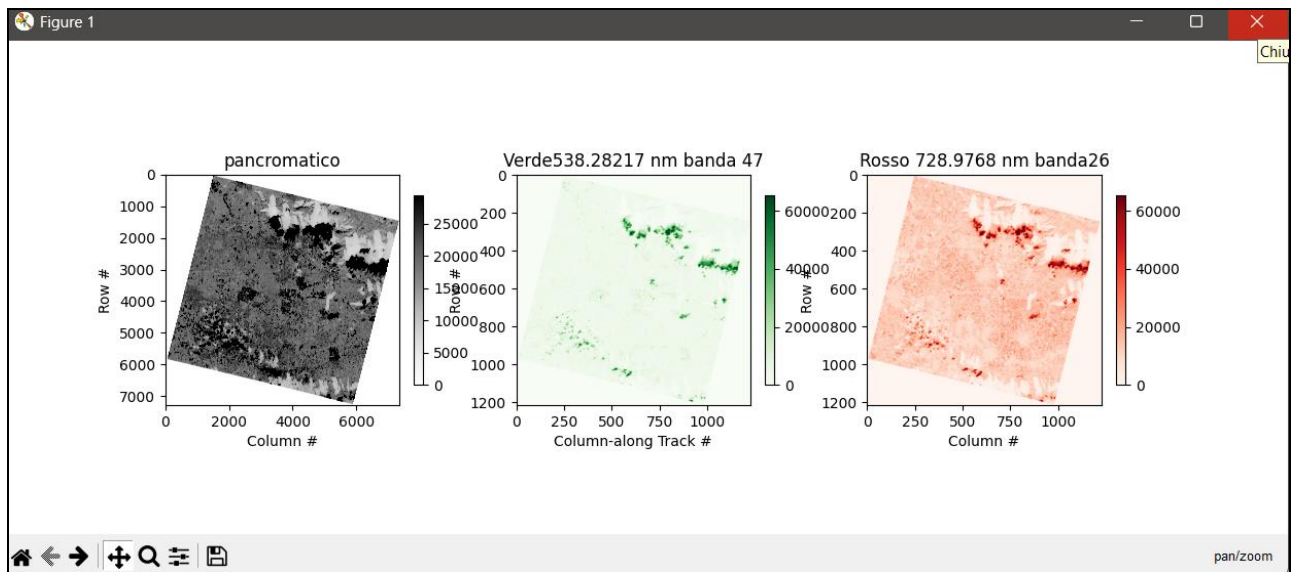
7.3. Check in the Python list if GDAL is present. If it is not there:

- From <https://www.lfd.uci.edu/~gohlke/pythonlibs/#gdal> you can download the .whl file of the gdal library suitable for the machine you are using. If useful, use the file already downloaded and located in the "Extra components to install" folder: *GDAL-3.4.3-cp311-win_amd64.whl*
- Right-click in the directory where you downloaded the file and select 'open in terminal'.
- In the terminal, type `pip install h5py-3.7.0-cp311-cp311-win_amd64.whl`

Conclusions:

Once you have completed the above steps, you can check that all the required libraries are present. Do this as described in [step 5](#).

If they are all present, you will be able to run the program called *Prisma_Tool* as described in [step 6](#). After selecting the HDF5 file, the following window will open. Just close it to continue the execution.



Once the script has finished running, if all went well, we would find the GeoTIFF images extracted from the HDF5 file in the `GeoTIFF_level_data` folder.

Contact details for bug reports:

lorenzo.boccia@unina.it

delogu.gabriele@unigus.it