This document provides a detailed step by step approach for packaging the python based DA4RDM visualization application. It also documents the necessary files required and the folder structure necessary to build the package. Kindly follow the below steps to proceed accordingly:

Creating files and local folder structure

1. Make sure the latest version of pip is installed by executing the below pip command

python -m pip install --upgrade pip

1. Create a local folder structure as below

Packaging/

└── src/

└── da4rdm\_vis/

├── \_\_init\_\_.py

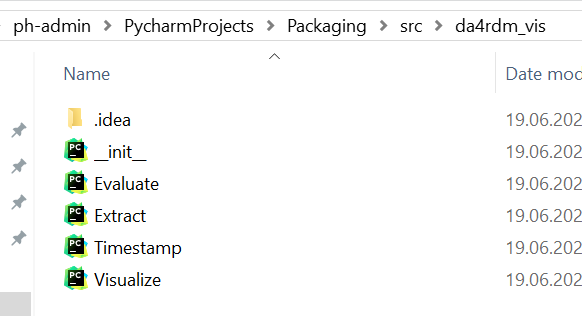
└── Evaluate.py

└── Extract.py

└──Timestamp.py

└──Visualize.py

In the above folder structure, the directory ***da4rdm\_vis*** contains all the executable code files while the package related files (details provided below) will be added to the repository ***Packaging*** later.



1. Once the folder structure is created, add the files that will be used for project distribution. The primary files needed to be added are license, pyproject.toml and README.md. Please refer <https://packaging.python.org/en/latest/tutorials/packaging-projects/> to create the files and understand its content.

Packaging/

├── LICENSE

├── pyproject.toml

├── README.md

└── src/

── da4rdm\_vis/

── \_\_init\_\_.py

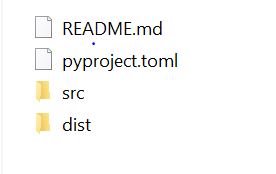
└── Evaluate.py

└── Extract.py

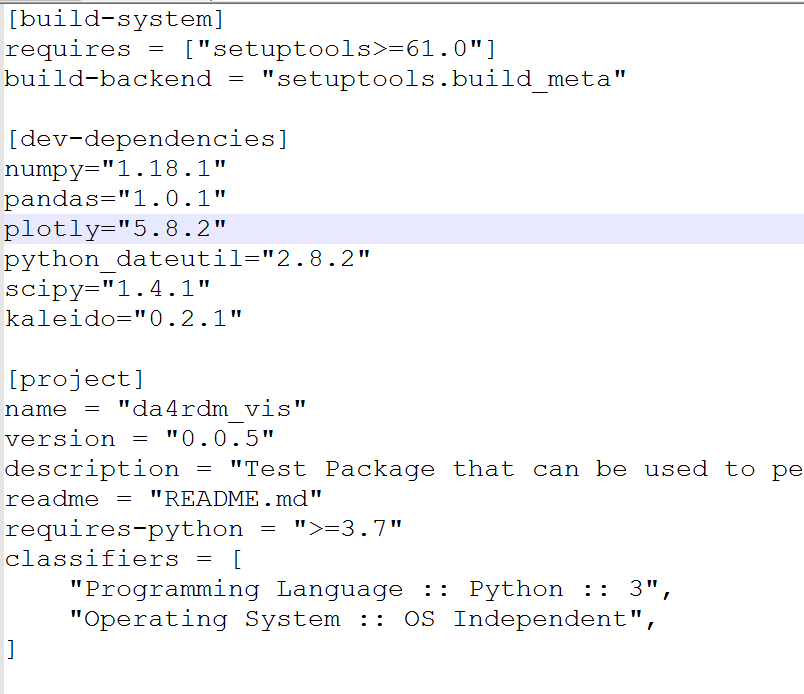
└──Timestamp.py

└──Visualize.py

1. Once completed the files should be available along with src in the Packaging directory.



1. pyproject.toml is used to configure the metadata information such as name and version of the package.



Generating distribution archives

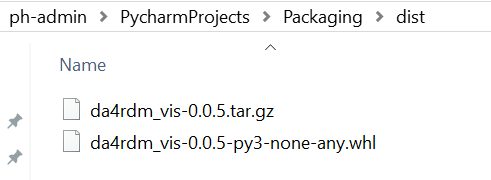
1. Once the files are ready and the folder structure is ready the distribution packages can be generated for the package. Make sure the latest version of build is installed by running the below command

python -m pip install --upgrade build

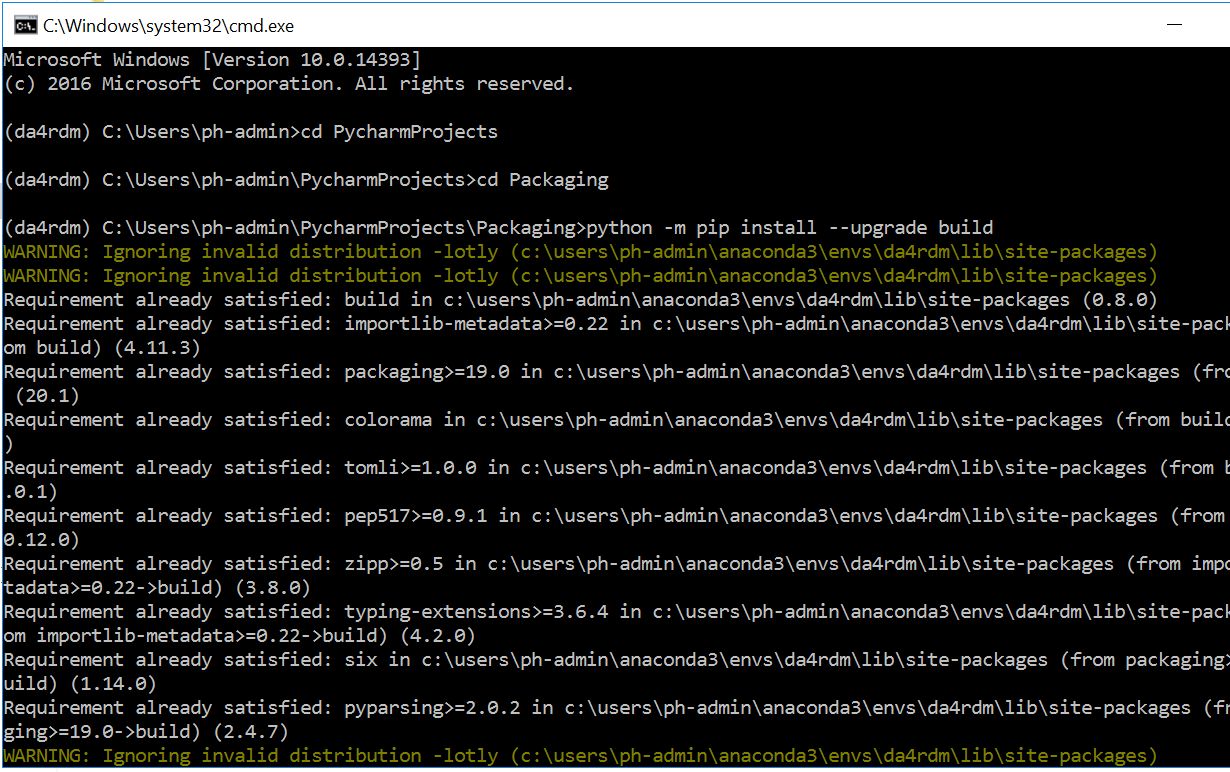
1. Now run the below command from the same directory where pyproject.toml is located :

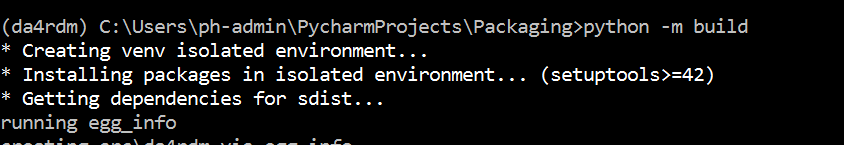
python -m build

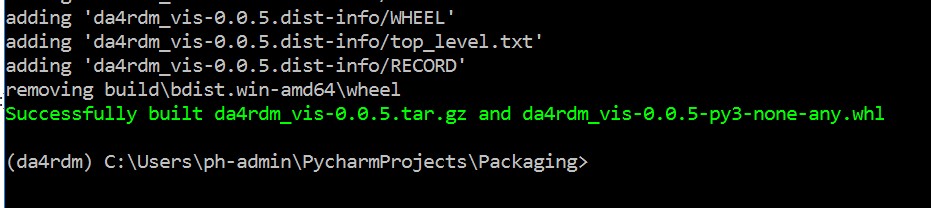
Once the above commands are executed, a new directory named ***dist/*** is created with two files in it.



Below are the snippet from the command line execution of the above few steps for your reference:







## Uploading the distribution archives

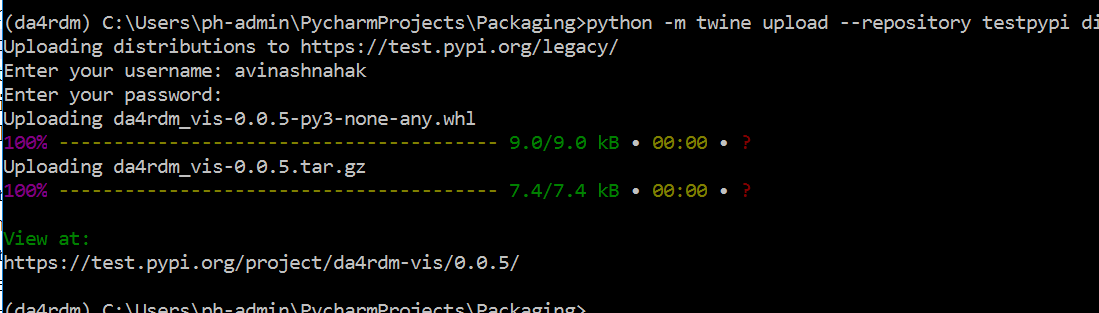
1. To securely upload your project, you’ll need a PyPI  [API token](https://test.pypi.org/help/#apitoken). Create one at <https://test.pypi.org/manage/account/#api-tokens>, setting the “Scope” to “Entire account”. **Don’t close the page until you have copied and saved the token — you won’t see that token again.**
2. Now that you are registered, you can use [twine](https://packaging.python.org/en/latest/key_projects/#twine) to upload the distribution packages. Install twine using the command below

python -m pip install --upgrade twine

1. Once installed, run Twine to upload all of the archives under dist using the ommand below

python -m twine upload --repository testpypi dist/\*

1. You will be prompted for a username and password. For the username, use \_\_token\_\_. For the password, use the token value, including the pypi- prefix.



1. Once uploaded your package can be viewed on TestPyPI, for example, <https://test.pypi.org/project/da4rdm-vis> as shown below

