

# Manual for [Search Sentinel-2 L1C by tile](#)

## Introduction

This app was developed for research purposes. The user inputs the conditions under which the search through the Copernicus Open Access Hub api. The Sentinel 2 satellite imagery search is tile based. The app gives the opportunity to make single variable search and multiple variables search

## Tile search

The search is conducted to find and download Sentinel-2 L1C imagery using MGRS tile . On the Copernicus Open Access Hub, it seems to be available for most L1C products (product type S2MSI1C) from recent years, but this differs by region, too .

The Military Grid Reference System (MGRS) is the geocoordinate standard used by NATO militaries for locating points on Earth. The MGRS is derived from the Universal Transverse Mercator (UTM) grid system and the Universal Polar Stereographic (UPS) grid system, but uses a different labeling convention. The MGRS is used as geocode for the entire Earth

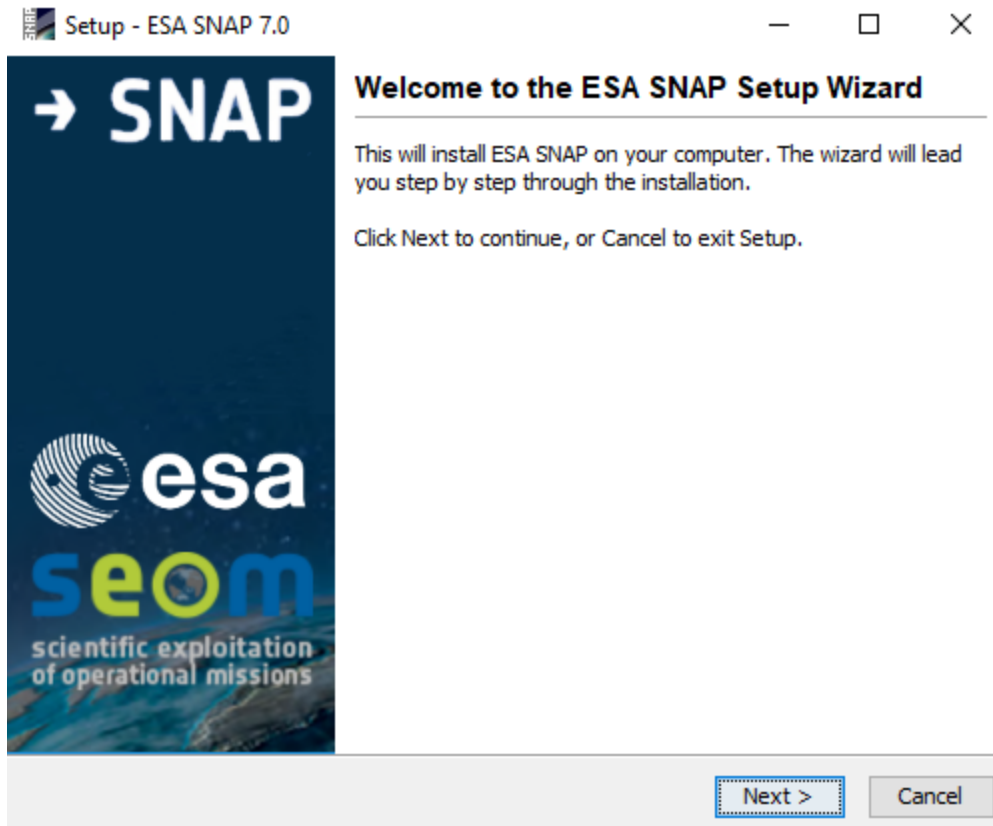
## Environment creation

1. install [Anaconda](#)
2. Install python 3.6
3. Download latest [SNAP](#) release on Windows,
4. Create python virtual environment

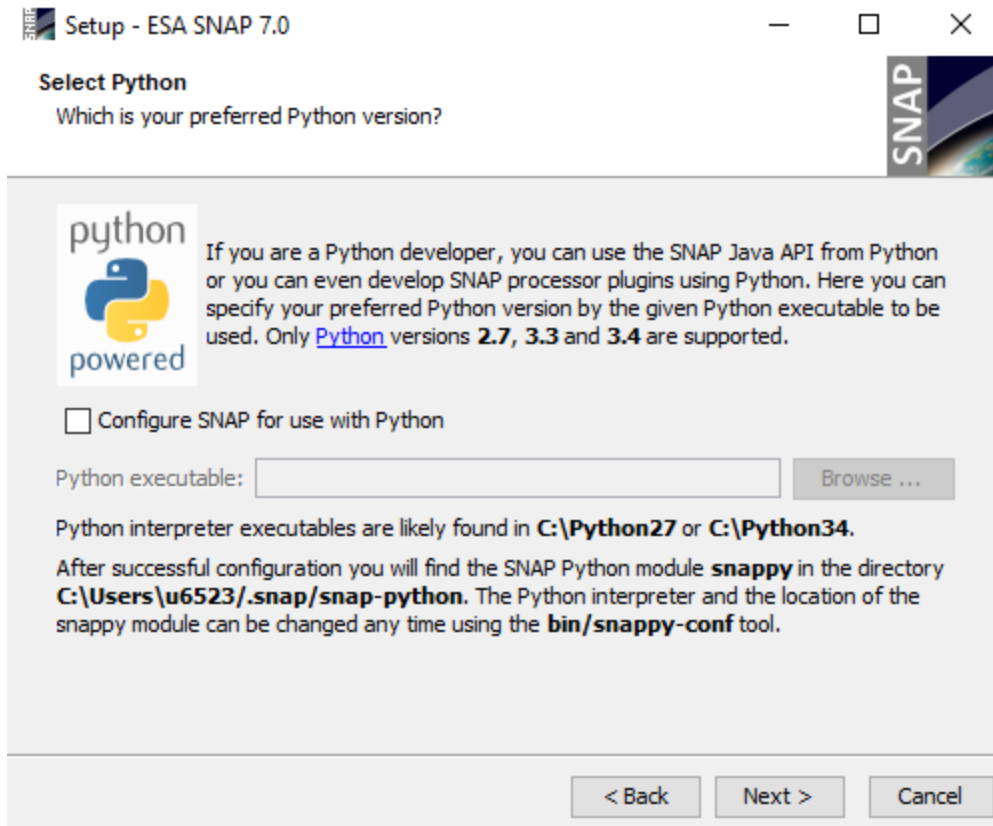
```
conda create -n envname python=3.6
```

5. Install Snap

Next, we can start installing the SNAP software using the downloaded executable file.



You can configure your python installation to use snappy during the SNAP installation itself by checking the tickbox and providing the path to your python directory (as shown below). However, in this tutorial, we will **NOT** check the tickbox and configure snappy after SNAP has finished installing.



The rest of the installation process is pretty straight forward. The first time you start up SNAP it will do some updates. Once they've finished you are all set to use the SNAP software.

#### 6. Configure Snappy

Once SNAP has finished installing, we want to get the directory location of where our virtual environment "snap" is located. From there, we want the following two paths:

- A. Python executable path for your virtual environment
- B. A folder called "Lib" in your virtual environment directory

Open up the SNAP Command-Line Tool, which is installed as part of the SNAP software and run

the following command:

```
snappy-conf {path_to_snap_env}\python.exe {path_to_snap_env}\Lib\
```

## 7. ( Optional )Extra virtual environment settings

Go over to the {virtual\_env “snap ”directory } > Lib > snappy.Open the snappy configuration file called “*snappy.ini*”.Edit the value for the parameter called “*java\_max\_mem*” and set it to a value that corresponds to about 70–80% of your system’s RAM.

```
java_max_mem: 26G
```

In the same directory, lies a file called “*jpyconfig.py*”. Change the parameter “*jvm\_maxmem*” like so:

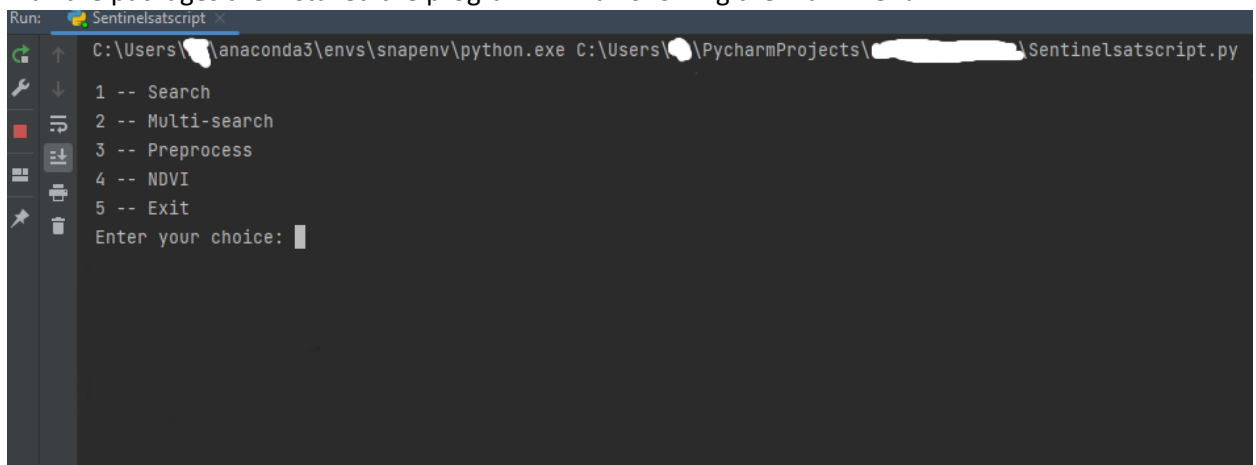
```
jvm_maxmem = '26G'
```

Next, navigate to where your SNAP software is installed. By default it is set to C:\Program Files\snap. Inside it, you will find the folder called *etc* containing the SNAP properties file called “*snap.properties*”. Edit the file and change the parameter called *snap.jai.tileCacheSize*. Set this to a value equal to 70–80% of the *java\_max\_mem* value. The number you enter should be in megabytes. You may have to enable administrator privileges to save the changes to this file.

8. (Later stage reminder) There might be a lot of packages needed to be installed for the program to run

## Users guide

1. If the virtual env ,the Snap etc is installed correctly the the main python file to run is Sentinelatscript.py
2. If all the packages are installed the program will run showing the main menu



6. From this point the user has many different options of usage of the app

## Scenario 1: Simple search in a specific geographic tile for certain period of time and cloud coverage .

1. The user has to do is to log in the [Copernicus Open Access Hub](#) through the program.
2. Then the user will be asked to insert the number of tiles for the area of interest (the tile codes could be found [here](#))
3. The user now has to to insert the tile codes eg 33VUC, 33UUB
4. Now the next stage is to insert the cloud coverage percentage (preferable integers) and the date(YYYYMMDD) that the search for satellite images is to be conducted .
5. When the search is completed the program will show the number of results based on the as above inserted conditions and if there are results it will show them and their online/offline status by ESA ( European Space Agency ) .
6. Then the user will have the choice to download all or just some of the products found . If the user downloads any product it will be saved in <<SentinelSnapIMG\_DATA>> a file that the program will create
7. If the user wants neither to download all the products or some then the results will be redirected to the main menu

## Scenario 2: Multi search for geographic tiles in multiple periods of time or cloud coverage percentages .

1. The user has to do is to log in the [Copernicus Open Access Hub](#) through the program.
2. Then the user will be asked to insert the number of tiles for the area of interest (the tile codes could be found [here](#))
3. Next the user will decide which of the two variables will be a constant for the specific search(date or cloud percentage )

```
1 -- Certain tile/s specific period but different cloud coverage percentage
2 -- Specific tile/s in different time periods but certain cloud coverage percentage
3 -- Back to main menu
Enter your choice:|
```

4. The user now has to to insert the tile codes eg 33VUC, 33UUB
5. Now the next stage is to insert the cloud coverage percentage/s (preferable integers) and the date/s(YYYYMMDD) that the search for satellite images is to be conducted .
6. When the search is completed the program will show the number of results based on the as above inserted conditions and if there are results it will show them and their online/offline status by ESA ( European Space Agency ) .
8. Then the user will have the choice select which of the products found the user wants to download . . If the user downloads any product it will be saved in <<SentinelSnapIMG\_DATA>> a file that the program will create
7. If the user wants neither to download all the products or some then the results will be shown and the user will be redirected to the main menu

### Scenario 3: Preprocessing already downloaded satellite images.

1. When the user choose this option the user will be asked to paste the name of the zip file from the a satellite image product as downloaded from [Copernicus Open Access Hub](#)
2. The program here will unzip the product and start preprocessing it (resampling and doing subset )  
\*at the moment it is sub setting the images based in the test area of the project
3. The now fully auto-preprocessed image will be saved with an appropriate name and save it the file << IMG\_DATA\_preprocessed>> .
4. After that user has the chance to see the preprocessed image in grayscale .

### Scenario 4: Getting an image with NDVI (Normalized Difference Vegetation Index)

1. The user will have to input the name of the .tif preprocessed image that the user wants to apply NDVI .
2. If the file exist it will analyze it pixel by pixel and it will recreate the picture and will assign to each pixel a number indicating the high ,low existence of vegetation .
3. When the initial picture is fully analyzed it will be saved with an appropriate name in the file << IMG\_DATA\_NDVI>>

$$NDVI = \frac{\rho_{NIR} - \rho_{Red}}{\rho_{NIR} + \rho_{Red}}$$