

## 衛星版 BLB 損害評価ツールインストールガイド

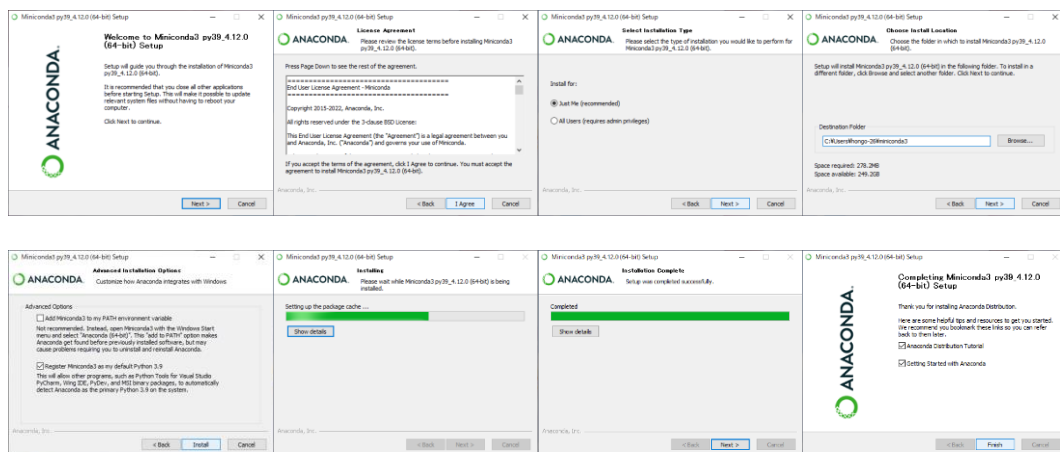
本マニュアルでは衛星版 BLB 損害評価ツール（SatelliteTool）およびその実行に必要なソフトウェアを Windows PC にインストールする方法について説明する。以下にインストールの手順を示す。

1. Miniconda3 のインストール
2. Python モジュールのインストール
3. JAVA のインストール
4. SNAP のインストール
5. SatelliteTool のインストール

### 1. Miniconda3 のインストール

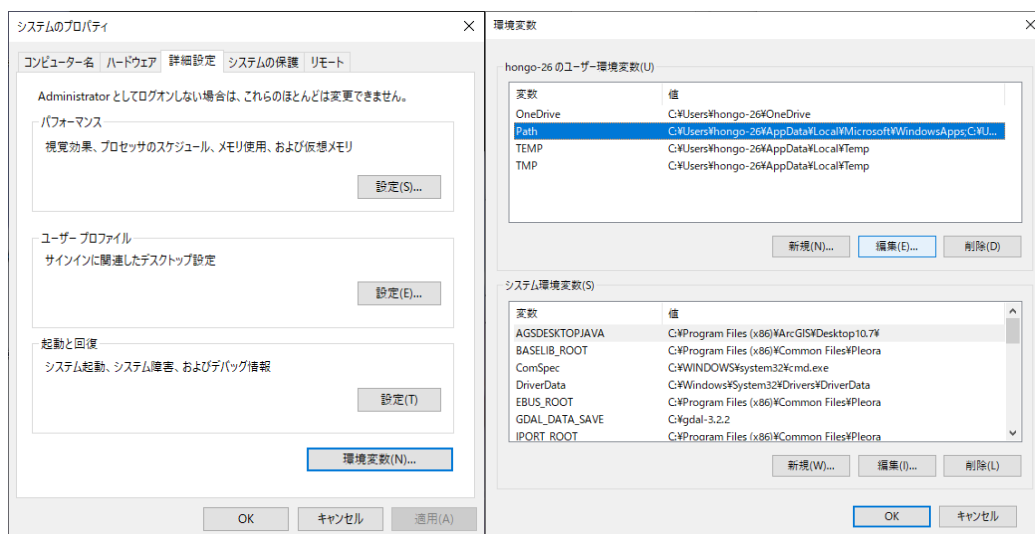
- 以下の URL からダウンロードした Miniconda3 のインストーラ（例：Miniconda3-py39\_4.12.0-Windows-x86\_64.exe）を実行して Miniconda3 をインストールする。インストールの設定は全て初期値（デフォルト）のままで構わない。動作確認された Python のバージョンは 3.9。

<https://docs.conda.io/en/latest/miniconda.html>



- 環境変数 Path の設定

デスクトップ上の Windows アイコンを右クリック -> 設定 -> 詳細情報 -> システムの詳細設定 -> 環境変数(N)... をクリックして環境変数の設定画面を表示させる。

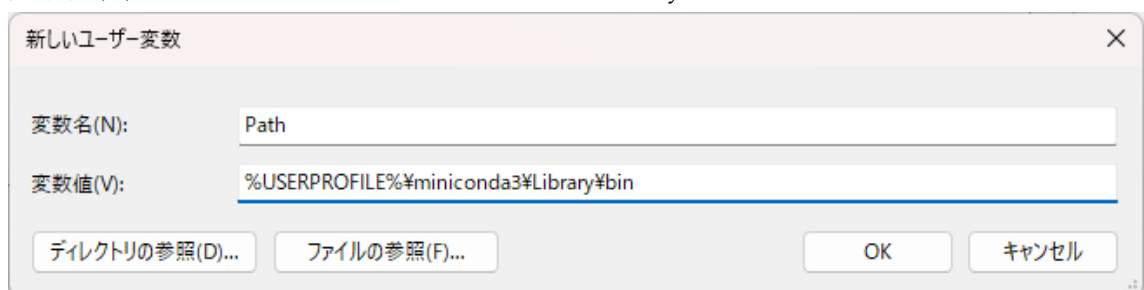


ユーザー環境変数に元から変数 Path があるかどうかによって、(1)または(2)のように Path の設定を行う。

(1) ユーザー環境変数に変数 Path がない場合、以下のようにユーザー変数 Path を新規作成する。

変数名(N) : Path

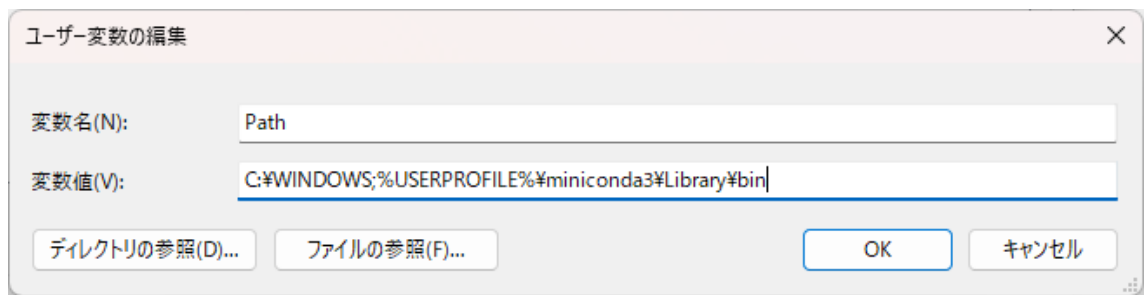
変数値(V) : %USERPROFILE%\¥miniconda3¥Library¥bin



(2) ユーザー環境変数に変数 Path がある場合、(2-1)または(2-2)のようにユーザー変数 Path を編集する。

(2-1) 値にパスが一つしか含まれていない場合 (例 : C:\¥WINDOWS)

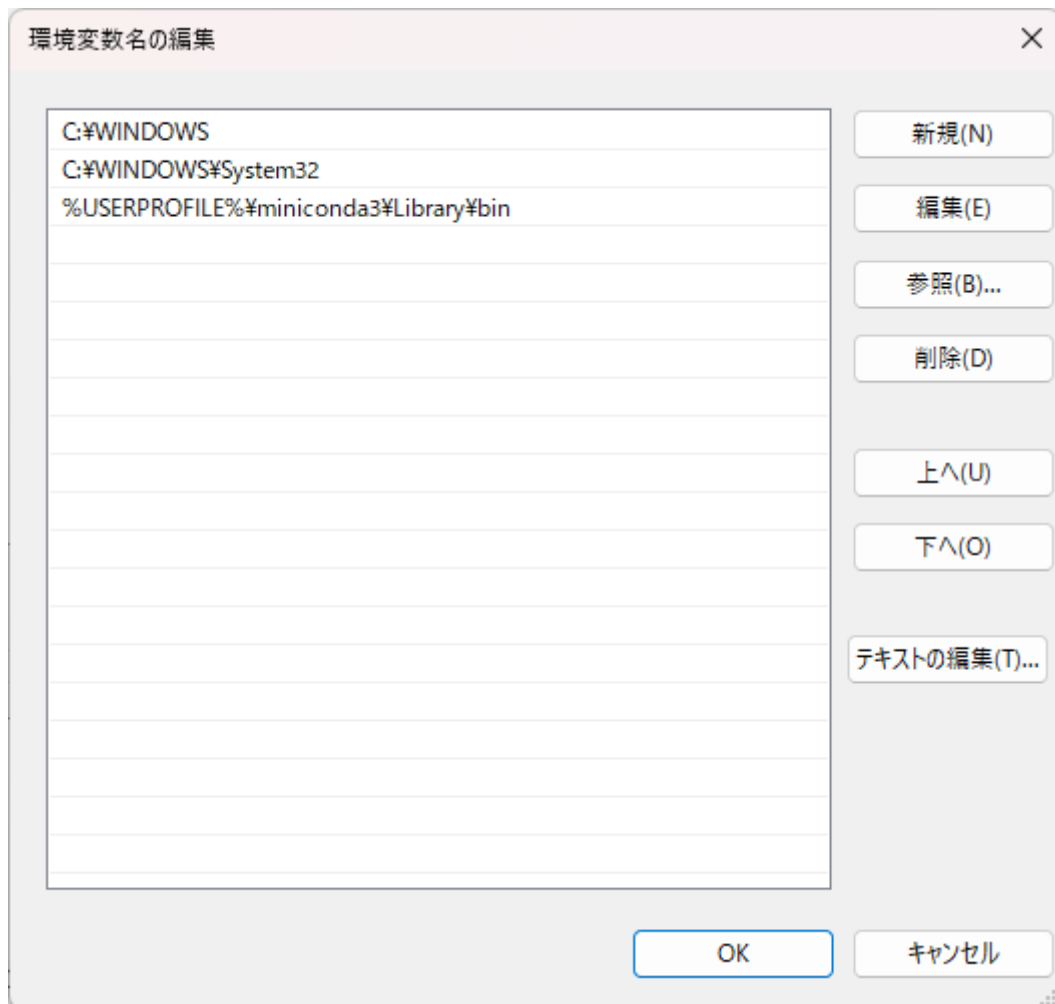
「ユーザー変数の編集」において変数値にセミicolon (;) で区切って %USERPROFILE%\¥miniconda3¥Library¥bin を追加する。



(2-2) 値にセミコロンで区切られたパスが複数含まれている場合（例：

C:¥WINDOWS;C:¥WINDOWS¥System32）

「環境変数名の編集」において新規で%USERPROFILE%¥miniconda3¥Library¥bin を追加する。



Path を設定したら、一度 Anaconda PowerShell Prompt を閉じて再起動する。

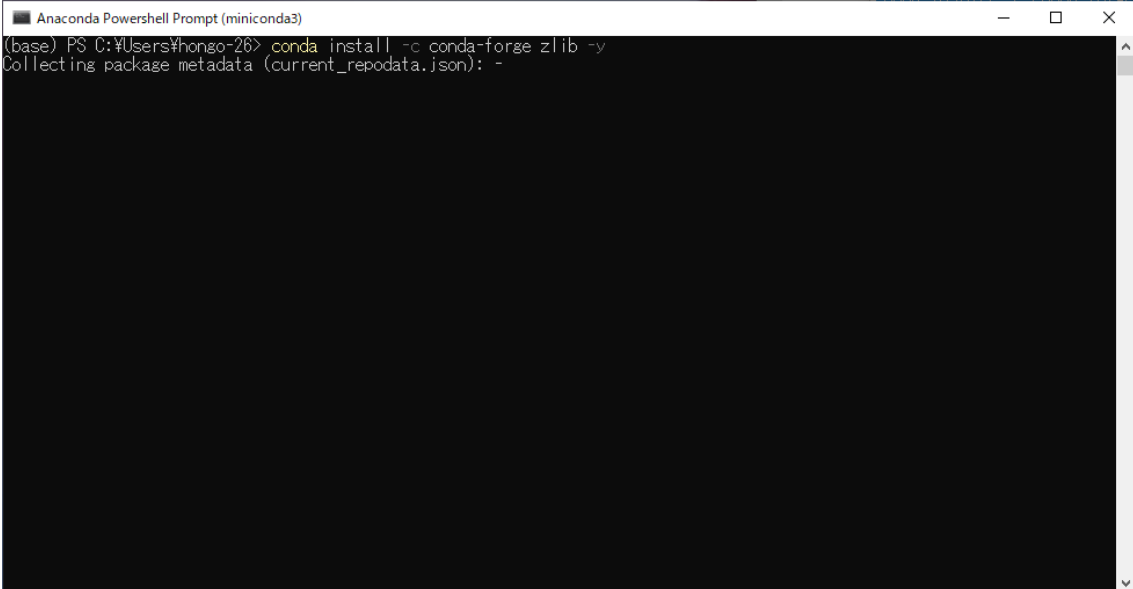
## 2. Python モジュールのインストール

- 手順 1 でインストールされた Anaconda Powershell Prompt で以下のコマンドを実行し、Python モジュールをインストールする。

```
conda install -c conda-forge zlib -y
conda install -c conda-forge numpy -y
conda install -c conda-forge scipy -y
conda install -c conda-forge pandas -y
conda install -c conda-forge xlrd==1.2.0 -y
conda install -c conda-forge openpyxl -y
conda install -c conda-forge matplotlib -y
conda install -c conda-forge ipython -y
conda install -c conda-forge scikit-learn -y
conda install -c conda-forge scikit-image -y
conda install -c conda-forge statsmodels -y
conda install -c conda-forge geopandas -y
conda install -c conda-forge rasterio -y # Optional
conda install -c conda-forge rasterstats -y # Optional
conda install -c conda-forge pyshp -y
conda install -c conda-forge cartopy -y # for Transplanting
conda install -c conda-forge psutil -y
conda install -c conda-forge xmldict -y
conda install -c conda-forge 7zip -y
pip install tkcalendar
pip install tkfilebrowser
pip install PyDrive2 # Optional
pip install -U csaps
pip install sentinelsat
```

※ “for Transplanting”は作付日推定に必要（SatelliteTool には不要）。

※ “Optional”は現状のスク립トには使われていない。



```
Anaconda PowerShell Prompt (miniconda3)
(base) PS C:\Users\hongo-26> conda install -c conda-forge zlib -y
Collecting package metadata (current_repodata.json): -
```

- constants.py の修正

%USERPROFILE%\miniconda3\Lib\site-packages\tkfilebrowser\constants.py (86 行目付近) を以下のように修正する。(後ろに LANG = 'en' を追加)

修正前 :

```
try:
    LANG = locale.getdefaultlocale()[0]
except ValueError:
    LANG = 'en'
```

修正後 :

```
# --- translation
try:
    LANG = locale.getdefaultlocale()[0]
except ValueError:
    LANG = 'en'
LANG = 'en' # <- added
```

```
Ubuntu
72 IM_FOLDER_LINK = os.path.join(PATH, "images", "folder_link.png")
73 IM_NEW = os.path.join(PATH, "images", "new_folder.png")
74 IM_FILE = os.path.join(PATH, "images", "file.png")
75 IM_FILE_LINK = os.path.join(PATH, "images", "file_link.png")
76 IM_LINK_BROKEN = os.path.join(PATH, "images", "link_broken.png")
77 IM_DRIVE = os.path.join(PATH, "images", "drive.png")
78 IM_RECENT = os.path.join(PATH, "images", "recent.png")
79 IM_RECENT_24 = os.path.join(PATH, "images", "recent_24.png")
80
81 # --- translation
82 try:
83     LANG = locale.getdefaultlocale()[0]
84 except ValueError:
85     LANG = 'en'
86 LANG = 'en'
87 |
88 EN = {}
89 FR = {"B": "octets", "MB": "Mo", "kB": "ko", "GB": "Go", "TB": "To",
90       "Name": "Nom", "Folder": "Dossier", "Size": "Taille",
91       "Name": "Nom", "Modified": "Modifié", "Save": "Enregistrer",
92       "Open": "Ouvrir", "Cancel": "Annuler", "Location": "Emplacement",
93       "constants.py" 208 lines --41%--
87,0-1 37%
```

- Windows 環境変数の追加

手順 1 の(1)と同様の方法で、ユーザー変数 PROJ\_LIB を新規作成する。

変数名(N) : PROJ\_LIB

変数値(V) : %USERPROFILE%\¥miniconda3¥Library¥share¥proj

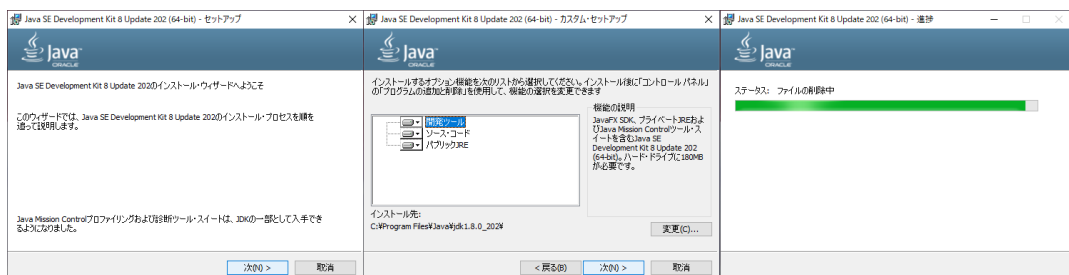
環境変数を追加したら、一度 Anaconda PowerShell Prompt を閉じて再起動する。

### 3. JAVA のインストール

- Oracle JDK のインストール

以下の URL からダウンロードした Java SE Development Kit のインストーラ（例：jdk-8u202-windows-x64.exe）を実行して JAVA をインストールする。動作確認された JAVA のバージョンは 8u202。

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>





## ● Windows 環境変数の追加

手順 1 の(1)と同様の方法で、ユーザー変数 JAVA\_HOME を新規作成する。

変数名(N) : JAVA\_HOME

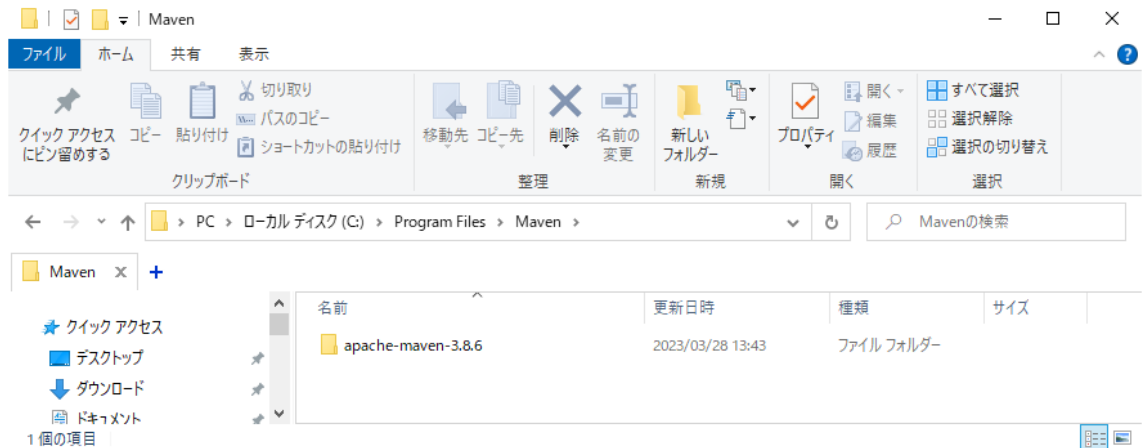
変数値(V) : C:\Program Files\Java\jdk1.8.0\_202

環境変数を追加したら、一度 Anaconda PowerShell Prompt を閉じて再起動する。

## ● Maven のインストール

以下の URL からダウンロードした Apache Maven のバイナリファイル（例：apache-maven-3.6.3-bin.zip）を展開し、中身を C:\Program Files\Maven の下に移動する。動作確認された Maven のバージョンは 3.6.3 および 3.8.6。

<https://maven.apache.org/download.cgi>



### ● 環境変数 Path の設定

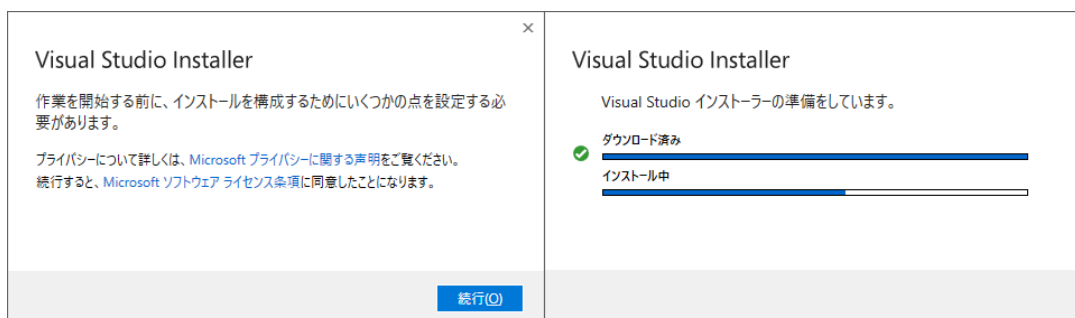
手順 1 の(2)と同様の方法で、ユーザー環境変数 Path に上の手順で作成された Maven の bin フォルダ（例：C:\Program Files\Maven\apache-maven-3.6.3\bin）を追加する。

Path を設定したら、一度 Anaconda PowerShell Prompt を閉じて再起動する。

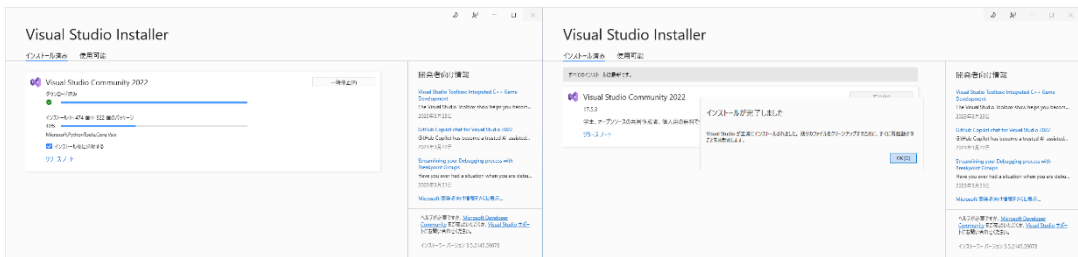
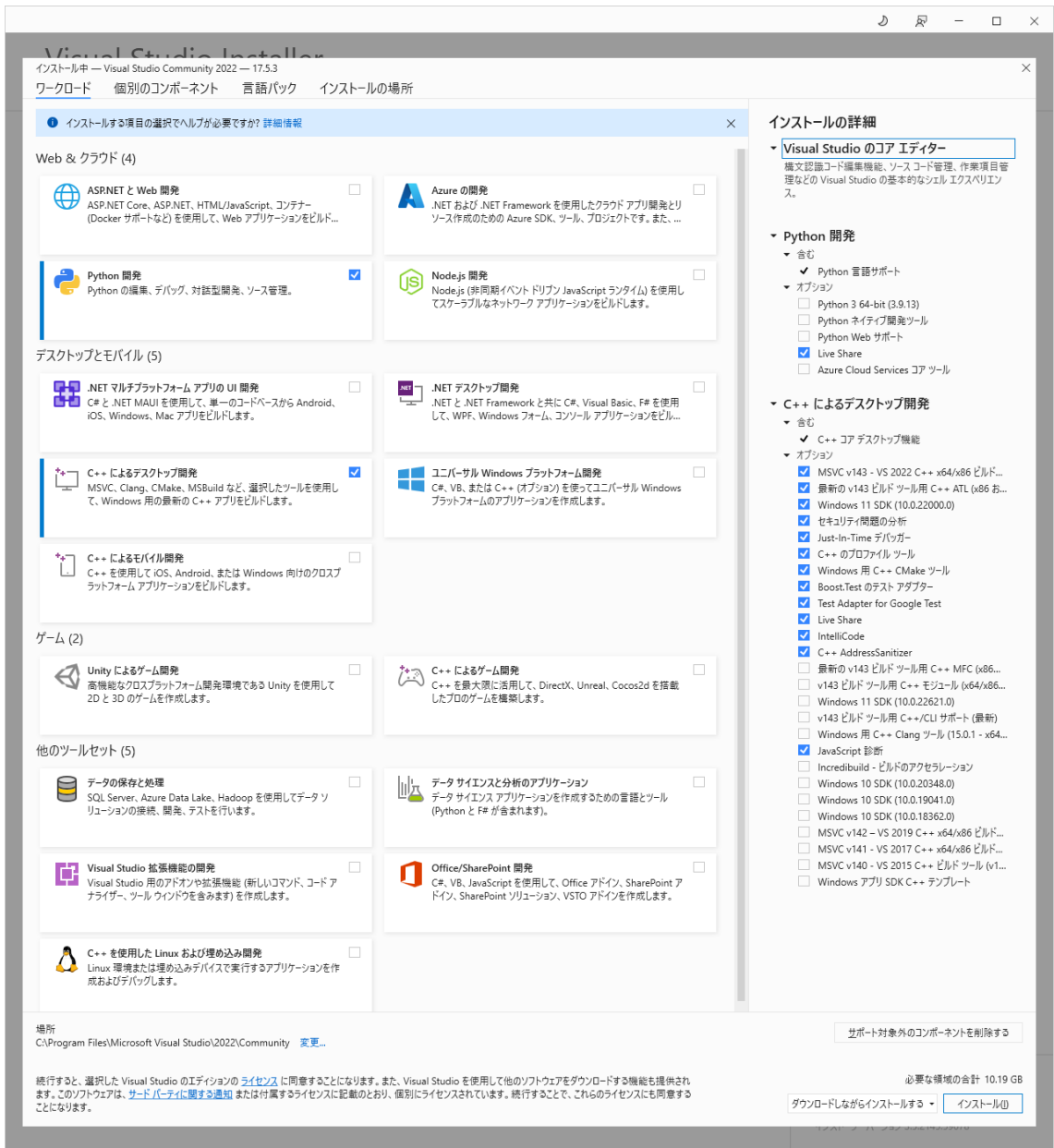
### ● Visual Studio のインストール

以下の URL からダウンロードした Visual Studio のインストーラ（例：VisualStudioSetup.exe）を実行して Visual Studio をインストールする。インストールの際、Python 開発と C++によるデスクトップ環境開発は必ず選択する。動作確認された Visual Studio のバージョンは 2022。

<https://visualstudio.microsoft.com/downloads/>







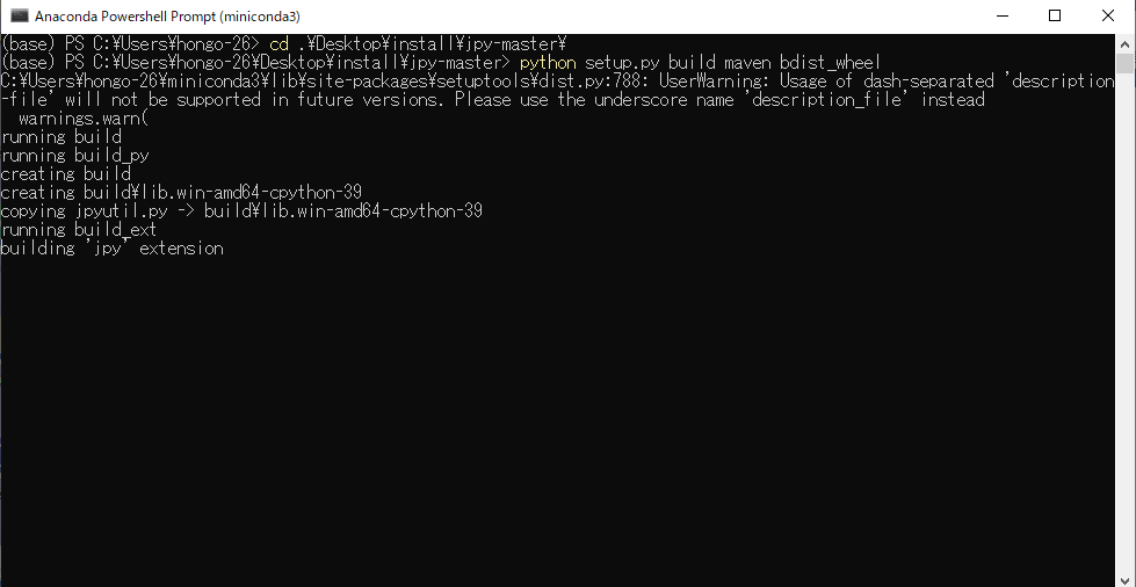
- jpy のインストール

以下の URL からダウンロードした jpy のソースファイル (jpy\_master.zip) を展開する。

<https://github.com/bcdev/jpy>

Anaconda Powershell Prompt で jpy\_master.zip を展開してできたフォルダに移動し、以下のコマンドを実行して jpy のビルドを行う。

```
python setup.py build maven bdist_wheel
```



```

Anaconda Powershell Prompt (miniconda3)
(base) PS C:\Users\hongo-26> cd .\Desktop\install\jpy-master\
(base) PS C:\Users\hongo-26\Desktop\install\jpy-master> python setup.py build maven bdist_wheel
C:\Users\hongo-26\miniconda3\lib\site-packages\setuptools\dist.py:788: UserWarning: Usage of dash-separated 'description-file' will not be supported in future versions. Please use the underscore name 'description_file' instead
  warnings.warn(
running build
running build_py
creating build
creating build\lib.win-amd64-cpython-39
copying jpyutil.py -> build\lib.win-amd64-cpython-39
running build_ext
building 'jpy' extension

```

ビルドが完了したら dist フォルダが作成されるので、Anaconda Powershell Prompt で dist フォルダに移動し、以下のコマンドを実行して jpy をインストールする。

```
pip install .\jpy-0.10.0.dev1-cp39-cp39-win_amd64.whl
```

※ cp39 の部分は Python のバージョンによって異なる可能性がある。

```

Anaconda Powershell Prompt (miniconda3)

-----
2021/05/18      1:00      6756  CHANGES.md
2021/05/18      1:00     11434  ez_setup.py
2021/05/18      1:00     25257  jpyutil.py
2021/05/18      1:00     11357  LICENSE.txt
2021/05/18      1:00      327   MANIFEST.in
2021/05/18      1:00      6910  pom.xml
2021/05/18      1:00     5157  README.md
2021/05/18      1:00      110   setup.cfg
-a-----
2023/03/28     14:25      265   setup.out
-----
2021/05/18      1:00     12919  setup.py
2021/05/18      1:00     2659  winbuild.cmd

(base) PS C:\Users\hongo-26\Desktop\install\jpy-master> cd .\dist\
(base) PS C:\Users\hongo-26\Desktop\install\jpy-master\dist> ls

ディレクトリ: C:\Users\hongo-26\Desktop\install\jpy-master\dist

Mode                LastWriteTime         Length Name
----                -
-a-----      2023/03/28      14:25         67575 jpy-0.10.0.dev1-cp39-cp39-win_amd64.whl

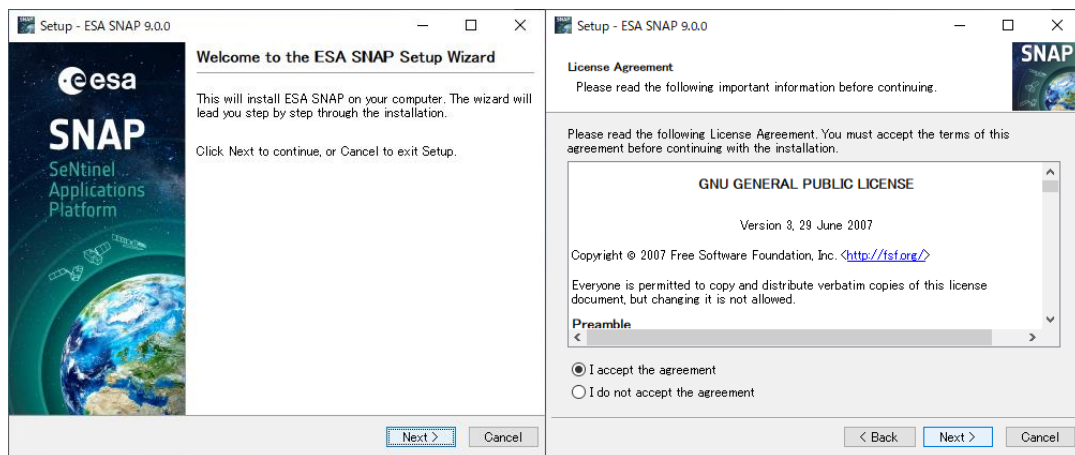
(base) PS C:\Users\hongo-26\Desktop\install\jpy-master\dist> pip install .\jpy-0.10.0.dev1-cp39-cp39-win_amd64.whl
Processing c:\Users\hongo-26\Desktop\install\jpy-master\dist\jpy-0.10.0.dev1-cp39-cp39-win_amd64.whl
Installing collected packages: jpy
Successfully installed jpy-0.10.0.dev1
(base) PS C:\Users\hongo-26\Desktop\install\jpy-master\dist>

```

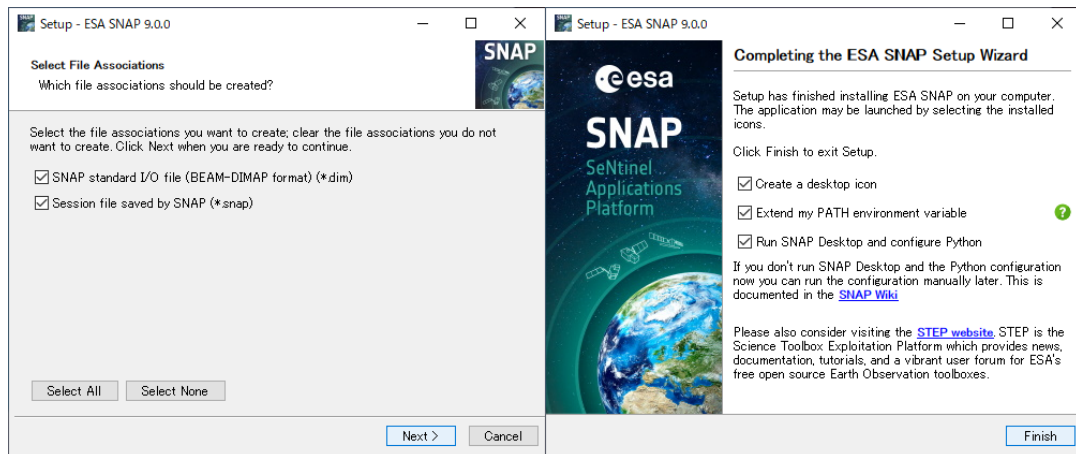
#### 4. SNAP のインストール

- 以下の URL からダウンロードした SNAP のインストーラ（例：esa-snap\_sentinel\_windows-x64\_9\_0\_0.exe）を実行して SNAP をインストールする。  
動作確認された SNAP のバージョンは 9.0.0。

<https://step.esa.int/main/download/snap-download/>







インストール中、Select Python の画面で

□ Configure SNAP for use with Python

というチェックボックスにチェックを入れ、

Python executable に Python 実行ファイルの場所

(%USERPROFILE%\miniconda3\python.exe) を入力する。

- Windows 環境変数の追加

以下のようにユーザー変数 SNAP\_HOME を新規作成する。

変数名(N) : SNAP\_HOME

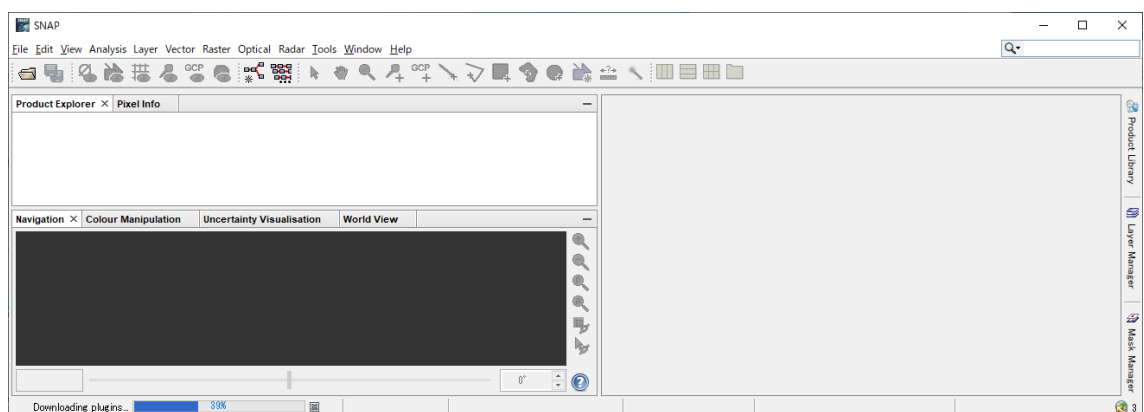
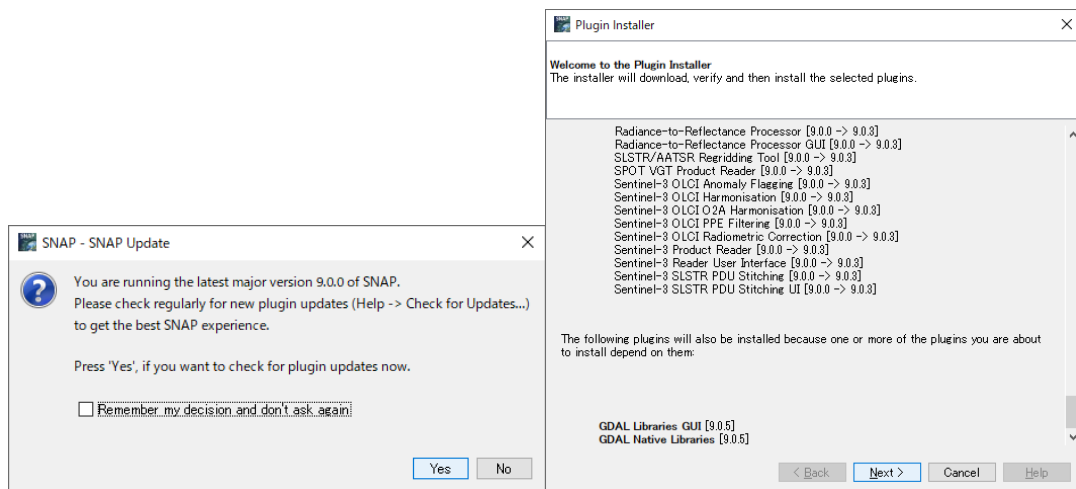
変数値(V) : C:\Program Files\snap

- 環境変数 Path の設定

手順 1 の(2)と同様の方法で、ユーザー環境変数 Path に %SNAP\_HOME%\bin を追加する。(SNAP のインストーラにより既に追加されている場合は不要。)

Path を設定したら、一度 Anaconda PowerShell Prompt を閉じて再起動する。

- SNAP を立ち上げるとアップデートするかどうか尋ねられるので、Yes を選択して SNAP をアップデートする。



アップデートを完了するには一度 SNAP を再起動する必要がある。(SNAP を再起動すると下のような画面が表示される。)



- %USERPROFILE%\¥snap¥snap-python フォルダの下に作成された snappy フォルダを%USERPROFILE%\¥miniconda3¥Lib フォルダの下にコピーする。
- Anaconda Powershell Prompt で ipython または python を立ち上げ、  
import snappy  
と入力して snappy をインポートできるかどうか確認する。

```

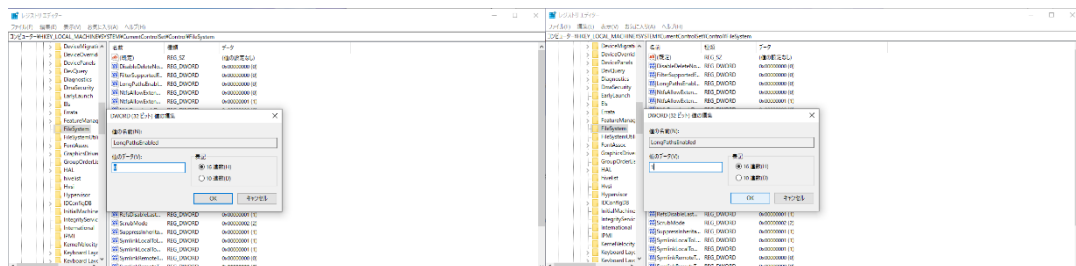
Anaconda Powershell Prompt (miniconda3)
(base) PS C:\Users\hongo-26> python
Python 3.9.16.1 packaged by conda-forge | (main, Feb 1 2023, 21:28:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import snappy
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external tool adapters
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new updates for the best SNAP experience.
>>>

```

## 5. SatelliteTool のインストール

- Windows のファイルパス文字制限の解除

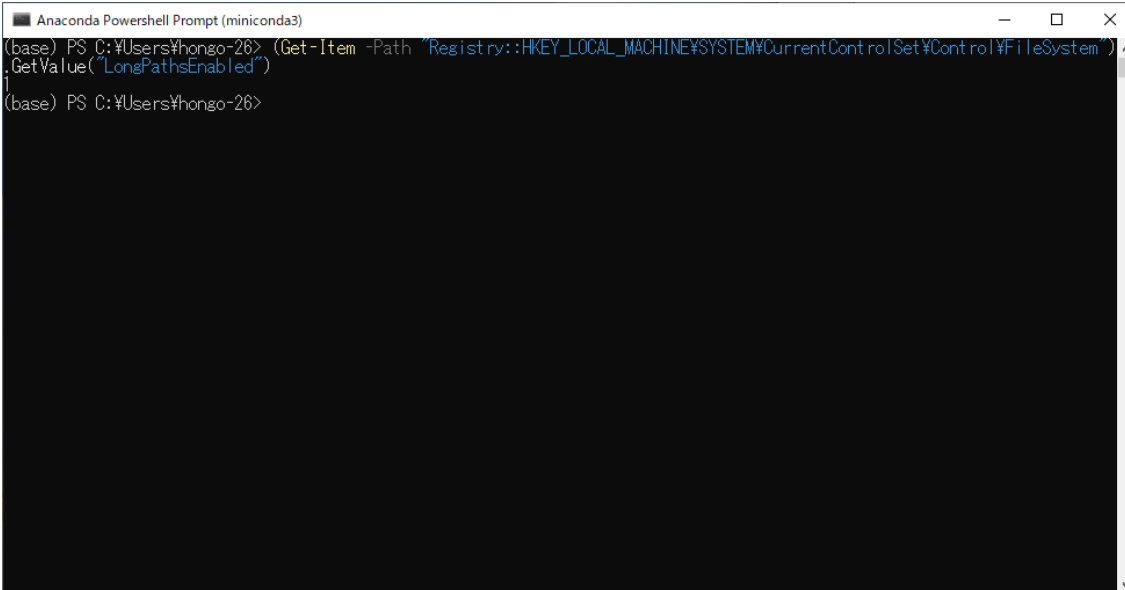
Windows の regedit を実行し、コンピューター > HKEY\_LOCAL\_MACHINE > SYSTEM > CurrentControlSet > Control > FileSystem にある LongPathsEnabled の値を 0 から 1 に変更する。



変更後、Anaconda Powershell Prompt で以下のコマンドを実行し、文字制限が解除されたことを確認する。(文字制限が解除されていれば 1、解除されていなければ 0 が出力される。)

(Get-Item -Path

"Registry::HKEY\_LOCAL\_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥FileSystem").GetValue("LongPathsEnabled")



```
Anaconda Powershell Prompt (miniconda3)
(base) PS C:\Users\hongo-26> (Get-Item -Path "Registry::HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥FileSystem").GetValue("LongPathsEnabled")
1
(base) PS C:\Users\hongo-26>
```

- Windows 環境変数の追加

手順 1 の(1)と同様の方法で、ユーザー変数 CDIR を新規作成する。

変数名(N) : CDIR

変数値(V) : %USERPROFILE%¥miniconda3

- スクリプトのコピー

SatelliteTool のソースファイルをホームフォルダ (%USERPROFILE%) の下で展開する。自動化も行う場合は Automation のソースファイルもホームフォルダの下に展開する。(それぞれ%USERPROFILE%¥SatelliteTool、 %USERPROFILE%¥Automation というフォルダの下に Python スクリプトが置かれるようにする。)

※ 最新版のソースファイルは GitHub から入手することができる。

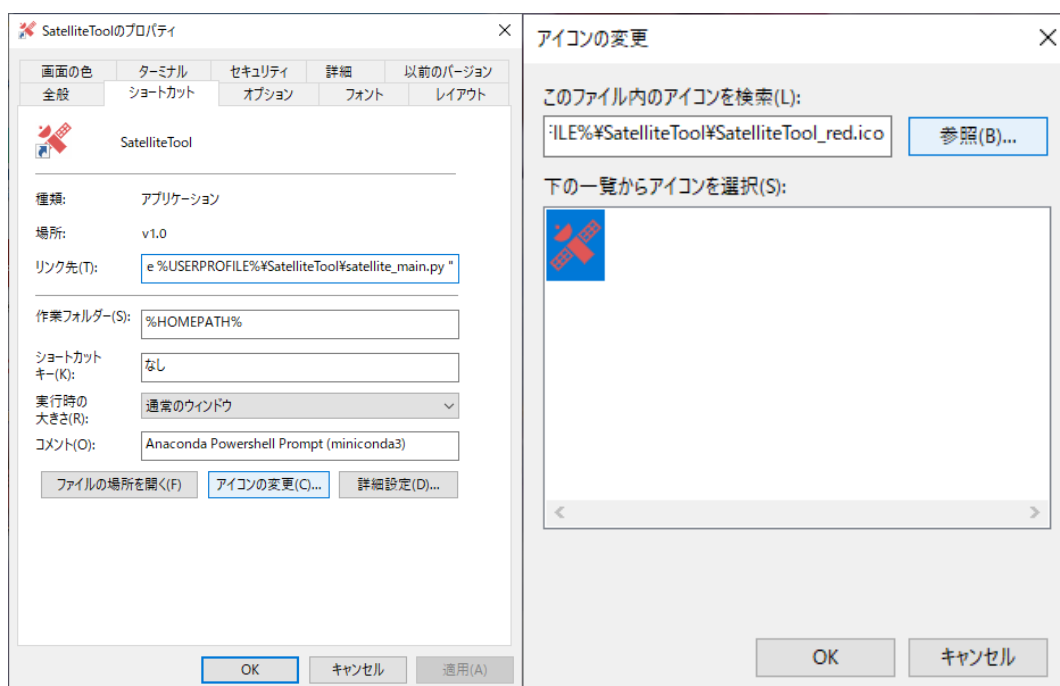
SatelliteTool: [https://github.com/nahiro/satellite\\_analysis](https://github.com/nahiro/satellite_analysis)

Automation: [https://github.com/nahiro/calc\\_trans](https://github.com/nahiro/calc_trans)



- ショートカットのコピー

SatelliteTool.lnk をデスクトップにコピーする。コピーしたショートカット上で右クリックし、アイコンの変更で%USERPROFILE%\SatelliteTool\SatelliteTool\_red.ico を選択する。



※ SatelliteTool.lnk の代わりに SatelliteTool\_noexit.lnk を使うと、スクリプト実行後にコンソール画面が開いたままになる。(SatelliteTool.lnk を使った場合、スクリプト実行後にコンソール画面が閉じられる。)

- データのコピー

必要に応じて、以下のファイルをローカルマシンに保存する。

- NAS サーバからのデータダウンロードに必要なファイル：  
%USERPROFILE%\netrc

- Bojongsoang の解析に必要なファイル・フォルダ：  
%WORK%\Shapefile\Bojongsoang (フォルダごと)  
%WORK%\WorldView\wv2\_190816\_mul.tif  
%WORK%\Sentinel-2\_Data\Bojongsoang\parcel\_mask.tif  
%WORK%\Sentinel-2\_Data\Bojongsoang\studyarea\_mask.tif

%WORK%\Sentinel-

2\_Analysis\Bojongsoang\Current\formula\pm\_formula\_score\_mean.csv

※ Copernicus サーバから Sentinel-2 L2A データをダウンロードする場合は以下のファイルも必要

%WORK%\Sentinel-2\_Data\Bojongsoang\bojongsoang.json

※ 作付日推定を行う場合は以下のファイルも必要

%WORK%\Sentinel-1\_Data\Bojongsoang\bojongsoang.json

%WORK%\Sentinel-1\_Data\Bojongsoang\Bojongsoang.shp (.shx 等を含む)

%WORK%\Sentinel-1\_Data\Bojongsoang\incidence\_list.dat

%WORK%\Sentinel-1\_Data\Bojongsoang\pixel\_area\_block.dat

%WORK%\Sentinel-1\_Data\Bojongsoang\x\_profile.npy

%WORK%\Sentinel-1\_Data\Bojongsoang\y\_profile.npy

– Cihea の解析に必要なファイル・フォルダ

%WORK%\Shapefile\All\_area\_polygon\_20210914 (フォルダごと)

%WORK%\WorldView\wv2\_180629\_mul.tif

%WORK%\Sentinel-2\_Data\Cihea\parcel\_mask.tif

%WORK%\Sentinel-2\_Data\Cihea\studyarea\_mask.tif

%WORK%\Sentinel-2\_Analysis\Cihea\Current\formula\pm\_formula\_score\_mean.csv

※ 作付日推定を行う場合は以下のファイルも必要

%WORK%\Sentinel-1\_Data\Cihea\cihea.json

%WORK%\Sentinel-1\_Data\Cihea\Cihea.shp (.shx 等を含む)

%WORK%\Sentinel-1\_Data\Cihea\Cihea\_outline (フォルダごと)

%WORK%\Sentinel-1\_Data\Cihea\incidence\_list.dat

%WORK%\Sentinel-1\_Data\Cihea\find\_nearest.npz

%WORK%\Sentinel-1\_Data\Cihea\paddy\_mask.tif

%WORK%\Sentinel-1\_Data\Cihea\paddy\_mask\_studyarea.dat

%WORK%\Sentinel-1\_Data\Cihea\paddy\_mask\_studyarea.tif

%WORK%\Sentinel-1\_Data\Cihea\pixel\_area\_block.dat

– Testsite の解析に必要なファイル・フォルダ

%WORK%\Shapefile\Testsite\_polygon\_20210914 (フォルダごと)

%WORK%\WorldView\wv2\_180629\_mul.tif

%WORK%\Sentinel-2\_Data\Testsite\parcel\_mask.tif

%WORK%\Sentinel-2\_Data\Testsite\studyarea\_mask.tif