**AddS2NoDataBorder V1 USER GUIDE**

**Windows Setup and Processing**

**Overview**

This document covers the process required to set up and run the Python application “AddS2NodataBorder”. The environment is a conda-based “virtual” environment created through Miniconda, a “slimmed down” version of the Anaconda software package and environment management suite. This can also be installed on a Linux-based system or WSL session running under Windows; additional details on this installation path are presented in Appendix A.

* If conda is not currently installed on your system: in a Windows web browser window (e.g. Microsoft Edge, Mozilla Firefox, etc.), go to:
  + <http://docs.conda.io/en/latest/miniconda.html>.
  + In the “Latest Miniconda Installer Links” section, look for and click on the link “**Miniconda3 Windows 64-bit**” to download the 64-bit Windows Miniconda installer (Figure 1). Follow the instructions to complete the installation.

|  |
| --- |
| Graphical user interface, text, application, email  Description automatically generated |
| **Figure 1. Windows 64-bit Miniconda Installer Link** |

* **Optional** **Step**: In the lower left-hand corner of the host’s display screen, go to the “Start” window and left click. Select “Anaconda Prompt (miniconda3).” Select “More” from the popup menu and select “Pin to taskbar”. This will allow easy access to a conda command line window where conda commands can be run. This step can be skipped if desired.
* Bring up a conda command window (Figure 2). If Step 2 was not performed, the first two actions in the description can be used to access the window.

|  |
| --- |
| **Shape, rectangle  Description automatically generated** |
| **Figure 2. “conda” Command Line Window. The “(base)” to the left of the command prompt indicates that the base (or system-level) conda environment is active.** |

* Check for updates to the conda application by entering **conda update conda** at the command prompt (Figure 3).
  + Depending on when conda was installed, there will likely be new and/or updated files to download and install that will bring conda to the latest version. Following the list of files, a [y/n] prompt will be shown. Entering **y** will initiate download and installation of the listed files.

|  |
| --- |
| Text  Description automatically generated |
| **Figure 3. Conda Update** |

* Create a separate “virtual” environment to run the application. **It is highly recommended to do this for any python-based project, in order to avoid potential clashes between project-specific python/supporting library installs and system-wide python/supporting library installs**. To do this, enter **conda create –name <name-of-virtual-environment>** at the command prompt (Figure 4). Once the environment is created, enter **conda activate <name-of-virtual-environment>** at the command prompt to change into the new environment. Remember: ALL supporting package installations and running the application must be done from within this environment!

|  |
| --- |
| Text  Description automatically generated |
| **Figure 4. Create Conda “virtual” environment cmacv11\_env.** |

* In the virtual environment you’ve just created, install the library packages required to support AddS2NodataBorder operation by entering **conda install <module\_name>** at the command prompt. These packages are:
  + Numpy (an array computation/mathematical library) – module\_name = **numpy**
  + GDAL (for image IO operations) – module\_name = **gdal**

If necessary, specific versions of packages can be installed by entering **conda install <module\_name>=version.** For additional information on version-specific package installation, please refer to <https://docs.anaconda.com/anaconda/user-guide/tasks/install-packages/>.

Depending on the order in which packages are installed, conda may flag some previously installed packages for “downgrade” (i.e. installation of an earlier version), in order to allow compatible installation of a new package. It shouldn’t happen very often.

* Test the installation of each package by trying to “import” it. This can be done for each package using the following steps:
  + Enter **python** at the command prompt. This launches the Python interpreter installed in the virtual environment.
  + To test **“numpy”** installation, enter **import numpy** at the Python command prompt.
  + To test **“gdal”** installation, enter **from osgeo import gdal** at the Python command prompt.
  + End the Python interpreter session by entering **quit()** at the Python command prompt.
  + If either package fails to import properly, try to remove it by entering **conda uninstall <package\_name>** at the Windows command prompt, then **conda install <package\_name>**. It’s possible some previously installed packages may be downgraded during the uninstall process. When package re-installation is completed, retest it using the appropriate import command after restarting the Python interpreter.

**Running the Application**

* “Install” the AddS2NodataBorder app source files to the local filesystem.
* Prepare the application to run by specifying the following processing parameters found at the beginning of the main application source file, “AddS2NodataBorder.py”:
  + ***<image\_basename>***: identification string used in the band image filenames with the form **TXXGGG\_YYYYMMDDTHHMMSS,** where **XX** represents the UTM zone number, **GGG** represents the grid ID, and **YYYYMMDDTHHMMSS** represents the UTC acquisition datatake time (which is not necessarily the same as the scene sensing time reported in the product metadata).
  + ***<full\_input\_file\_path>*:** fully qualified path to the input band image files. If processing a TOAR or Sen2Cor-corrected image organized in the .SAFE format, this path must include ALL components, e.g. the spatial resolution folder for a Sen2Cor-corrected image path.
  + ***<full\_output\_file\_path>*:** fully qualified path to the output band image files. If not currently found in the filesystem, it will be created.
  + ***<bands\_to\_process>*:** the list of bands to process. These are the band identifier substrings in the band image filenames, e.g. “B02” for band 2, etc.
  + ***<image\_type>*:** string identifying the type of image being processed. Allowed expressions are the substrings “TOA”, “TOAR”, “SEN2COR”, “S2COR”, “S2C”, or “CMAC”; these can be expressed only in fully upper-case or fully lower-case forms.
* In a Miniconda command window prompt, enter **conda activate <name-of-virtual-environment>.** The environment should switch from the “base” system-level environment to the specified virtual environment (Figure 5).

|  |
| --- |
| Text  Description automatically generated |
| **Figure 5. Switch to the virtual environment set up to run AddS2NodataBorder.** |

* At the command prompt, enter **python </path/to/Source\_Files>/AddS2NodataBorder.py.** As shown in Figure 6, if the configuration information is specified correctly, the program should launch. The program has been successfully launched when processing status messages are scrolled on-screen.

|  |
| --- |
|  |
| **Figure 6. Running AddS2NodataBorder. As the configuration was set up properly, the program has completed execution, with processing status messages displayed on-screen.** |

**APPENDIX A**

**Linux / WSL Linux Setup and Processing**

This appendix presents a procedure to create a virtual Python environment in a “native” Linux system or WSL session running on MS Windows. As with the Windows installation, a conda virtual environment based on Miniconda 3 is considered. If running in a native Linux or WSL command window previously set up on your system, the first 4 steps should be skipped.

* Bring up a cmd prompt window by right-clicking on the window in the lower-left corner of the screen, selecting the “Run” menu option, and entering **cmd** in the Open text box (Figure 7).

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |
| **Figure 7. Bring up a Windows command line window.** |

Right-click and select Run in the menu.

* If WSL is not currently installed and running on your system, enter **wsl –install -d Ubuntu** at the Windows command prompt**.** All the required Windows and Ubuntu components will be downloaded and installed. Any distribution can be specified; for the purposes of this installation procedure, Ubuntu was chosen as the Linux distribution. If a message upon completion of the installation requests a system reboot, go ahead and reboot; this will be needed to establish all the changes made to the system (e.g. new processes to run at startup, etc.).
  + If WSL has already been installed and set up (including having an assigned Linux username and password), simply enter **wsl** at the Windows command prompt. The next step can be skipped.
* Once the WSL main installation and Linux setup has completed, you will be asked to enter a Linux username and password. Do so; it’s recommended that users have a separate account to access the system as needed.
* After setting up your username and password, check to see if any Linux components need to be updated. Do this by entering **sudo apt update** at the Linux command prompt, entering your Linux password, then entering **sudo apt upgrade** at the command prompt. The **sudo** part of the update and upgrade commands provides temporary root-level credentials without having to log in as root; this allows sensitive administrative tasks to be done by “trusted” non-root users. Once the list of component updates has been generated, a yes/no prompt will appear requesting permission to download and install the updates. Enter **y** (or yes) at the command prompt to enable updating.
* At the command prompt, enter **cd ~** to switch to your home Linux directory (Figure 8).

|  |
| --- |
| Text  Description automatically generated |
| **Figure 8. Switch to your Linux home directory. As with the Windows installation described earlier, (base) indicates a system-level conda environment is active. The following “pwd” command displays the current (or “working”) directory.** |

* In your home directory, get the Linux Miniconda installer “Miniconda3 Linux 64-bit” from <https://docs.conda.io/en/latest/minconda.html>. If running Linux in a WSL session, you can do this by downloading the file in Windows and using Windows’ File Manager to copy the file from “Downloads” to your Linux home directory (accessed by clicking on the Linux penguin icon at the bottom of the File Manager filesystem display pane, clicking on the “Ubuntu” folder, clicking on the “home” folder, and finally clicking on your particular user home directory). Alternatively, you can download the file directly to your Linux home directory by entering the command **wget** [**https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh**](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh)at the Linux command prompt (Figure 9). If running on a “native” Linux host, you’ll likely have to use **wget** retrieval.

|  |
| --- |
| Graphical user interface, text  Description automatically generated |
| **Figure 9. Retrieving Linux Miniconda installer in a native Linux or WSL session using ‘wget’ command.** |

* At the Linux command prompt, enter **bash Miniconda3-latest-Linux-x86\_64.sh** to run the installation script. Additional details on installing Miniconda on Linux systems can be found in the web page accessed by the link contained in the ancillary text file **“link\_to\_linux\_miniconda\_installation\_procedure.txt”**, located in the PyCMACV11 main folder.
* Once conda has been installed, check to see if it needs updating by entering **conda update conda** at the Linux command prompt. This step is identical to the step for upgrading the Windows version of conda.
* Enter **conda create –name <name-of-virtual-environment>** at the miniconda command prompt to create a virtual environment. This step is identical to the step creating the virtual environment in the Windows setup procedure. When created switch to this environment by entering **conda activate cmacv11\_env**.
* Install the **“numpy”** and **“gdal”** packages as described earlier by entering **conda install <package\_name>** at the command prompt. Test each installation following the same procedure as described earlier (i.e. start a python session by entering **python** at the command prompt, then importing the module with the import command as described).
* In a Linux text editor (e.g. vi/vim, cmacs, etc.), prepare the main source file “AddS2NodataBorder.py” for running by specifying the <***image\_basename***>, ***<full\_input\_file\_path>***, ***<full\_output\_file\_path>***, ***<bands\_to\_process>***, and ***<image\_type>*** parameters as described earlier.
* Run the application by entering **python <path/to/AddS2NodataBorder\_Source\_Files>/AddS2NoDataBorder.py** at the Minconda window command prompt.