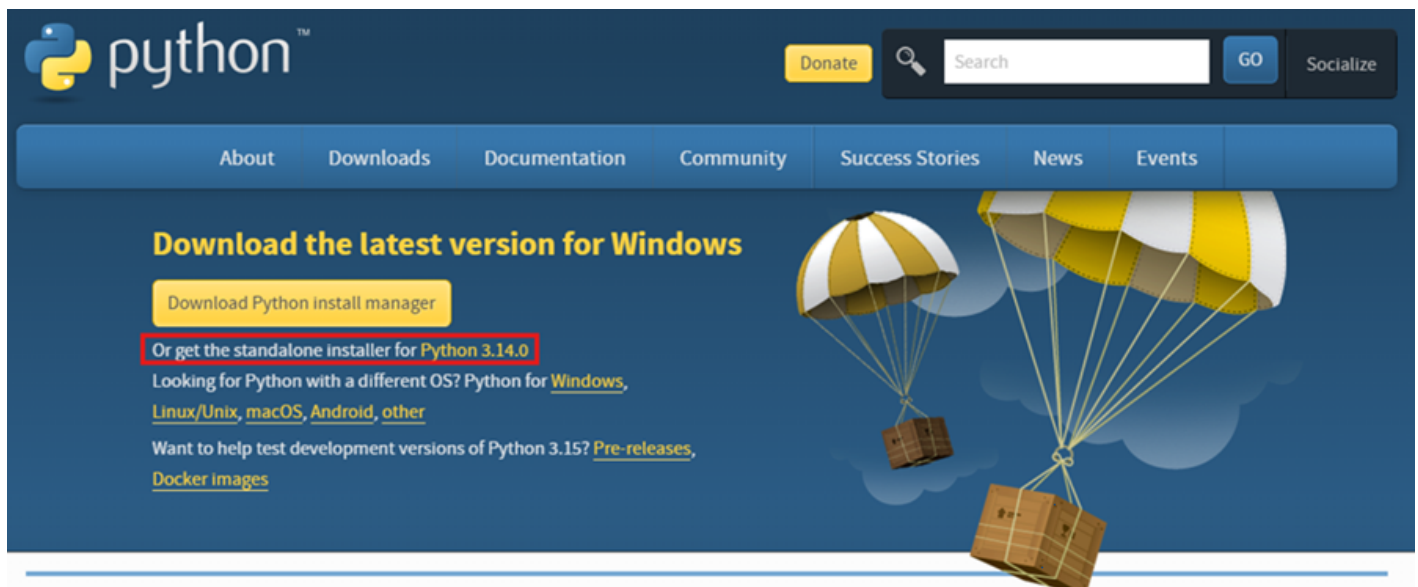


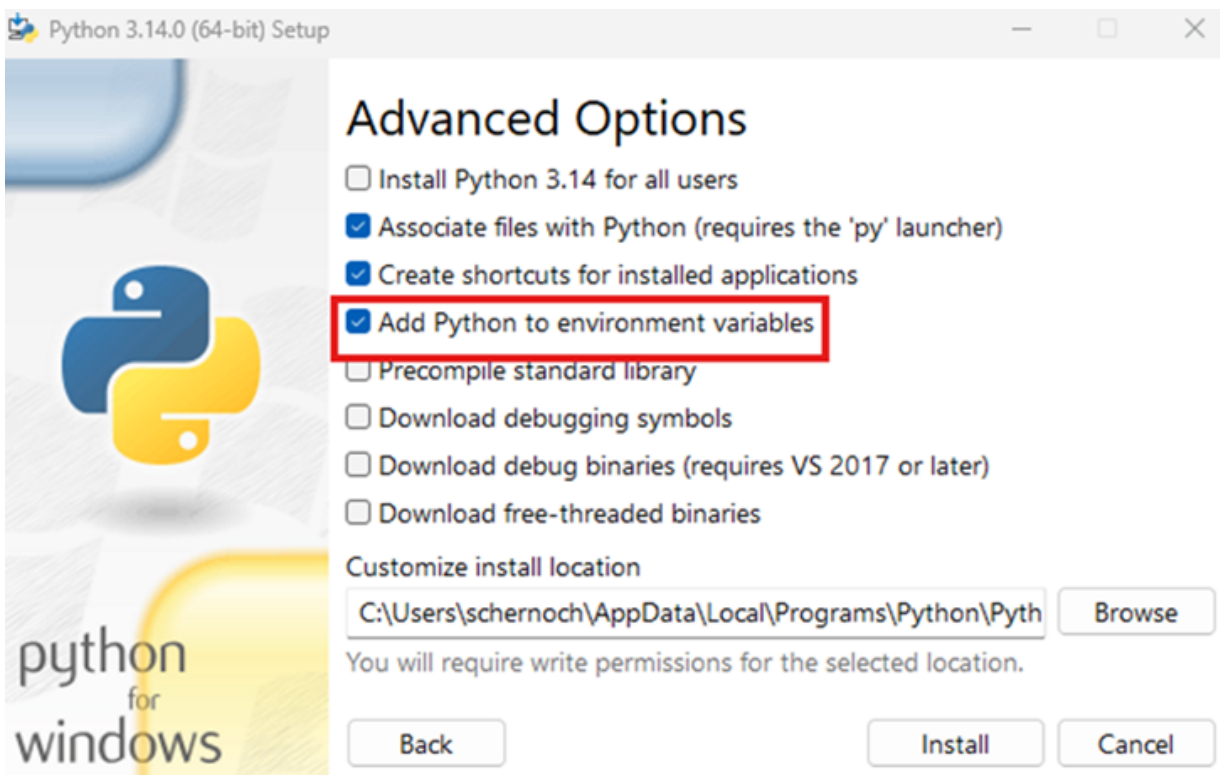
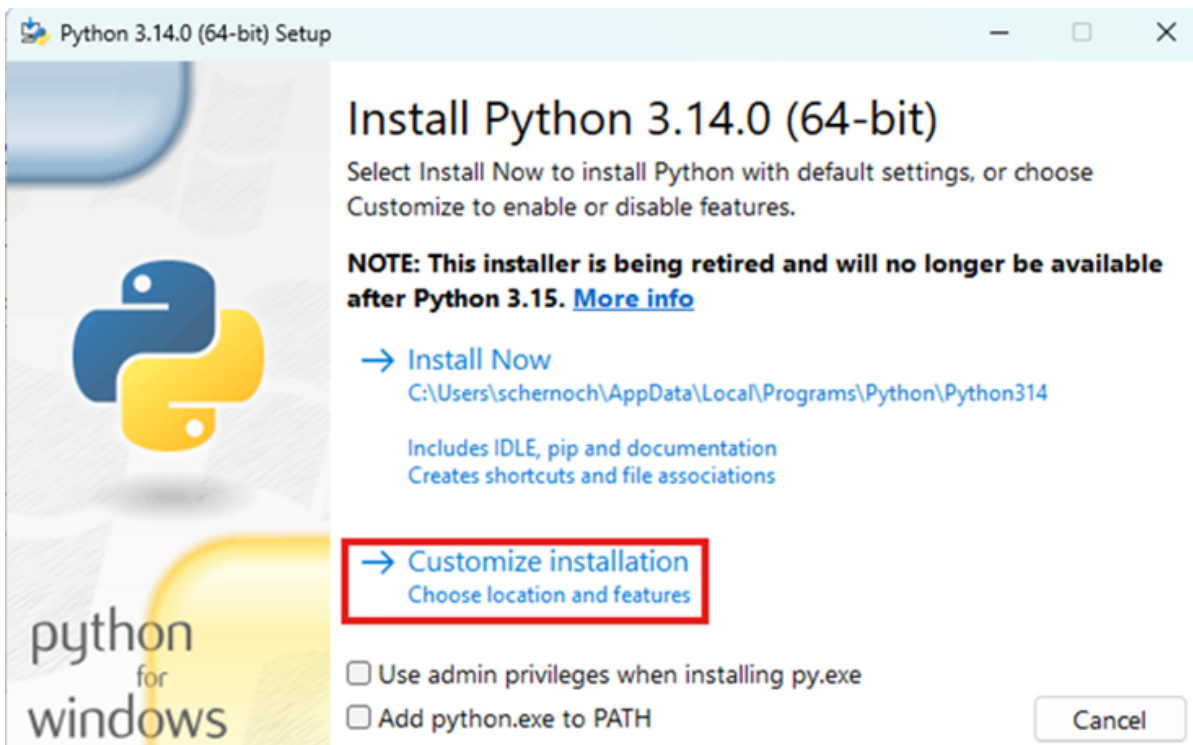
WIBL and YDVR data processing

Getting Set Up

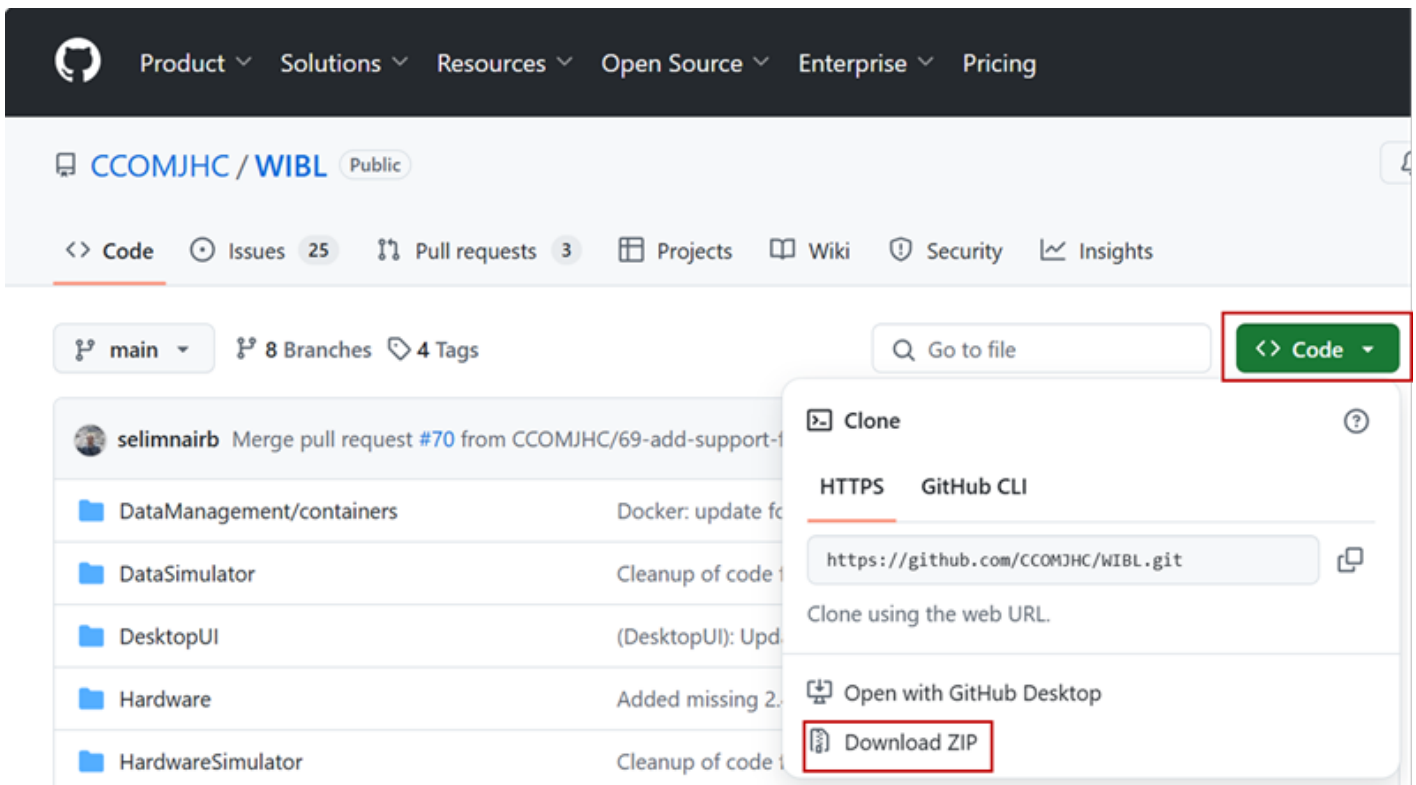
You will need to install the following programs to run the WIBL processing scripts:

- **Python 3.14:** download [here](#). Choose *Customize Installation*. Under Advanced Options check the box "Add Python to Environmental Variables." [More detailed instructions are provided [here](#)]





- **Windows PowerShell 7:** download [here](#)
- **Miniconda:** download [here](#)
- **Visual Studio Code:** download [here](#)
- **WIBL GitHub Repository:** [Linked here](#). Save the repository to Desktop (or Documents) - do not leave it in Downloads.



Perform the following steps the first time you run the WIBL scripts:

Note: Edit file paths as needed to match the location where your files are saved locally

```
#Open Anaconda Prompt

#Run the following code
conda init powershell

#Close Anaconda Prompt
-----

#Run Windows PowerShell 7 as an administrator (see screenshot below)

#Run the following code
Set-ExecutionPolicy Unrestricted

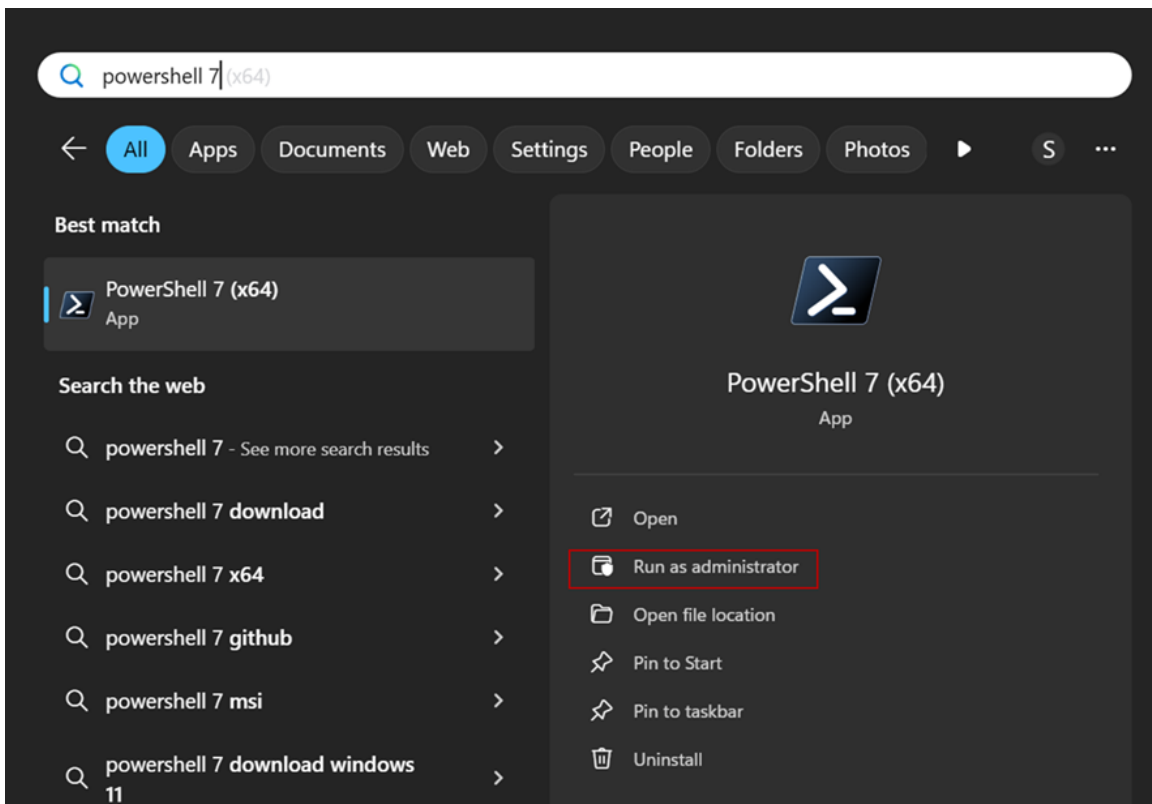
#Close Windows PowerShell 7, then reopen (not as admin)

#Set directory to the wibl-python folder within the WIBL repository
cd .\Desktop\WIBL-main\wibl-python

#Install packages
pip install .
```

```
#Create environment
conda env create -f environment.yml

#Execute the environment
conda activate wibl-python
```



Metadata - Configuration - Authentication

If you did not preload metadata onto the logger before installing it, you will need to create your metadata and configuration files before you can process raw data with the WIBL scripts.

Example metadata and configuration files can be found in the WIBL repository. Guidance for how to properly format each of these file types can be found in the README file located in the same folder.

Metadata, configuration, and README file locations in the WIBL repository:

WIBL-Main\wibl-python\examples\ship-metadata-complete.json
WIBL-Main\wibl-python\examples\configure-submission.json
WIBL-Main\wibl-python\examples\configure-submission-test.json
WIBL-Main\wibl-python\examples\README.md

To create metadata and configuration files specific to your dataset:

1. Copy the example files to a new folder in Documents (or Desktop) - don't work within the WIBL repository.
2. Open the copied files in **Visual Studio Code** to edit.

3. In the configuration file, make sure to set `"local": false`, or else your files will not be transmitted to the DCDB when you run **submitDCDB.ps1**.
4. The `configure-submission-test.json` can be used to submit data to the test-API on the DCDB (does not officially submit data).
5. Save the edited files with appropriate names (e.g., `chance-maritime-metadata-complete.json` and `configure-submission-USF.json`).

Further information on mandatory and recommended metadata can be found in [IHO's Guidance to Crowdsourced Bathymetry B-12 Edition 3.0.0](#)

The more comprehensive the metadata, the more valuable CSB products are to end-users! The DCDB will accept submissions meeting the mandatory metadata requirements, but organizations should strive to meet the highest standards recommended whenever possible.

You will need to download your organization's Trusted Node authentication token in order to submit processed data to the DCDB.

Running the WIBL Scripts

Note: There is a README file with more detailed information about the WIBL data management scripts and how to use them in the WIBL repository.

README file location in the WIBL repository:

WIBL-main\wibl-python\scripts\data-management\README.md

Once you're ready to start processing, perform the following steps:

```
#Run Windows PowerShell 7 as an administrator

#Run the following code
Set-ExecutionPolicy Unrestricted

#Close Windows PowerShell 7, then reopen (not as admin)

#Set directory to folder where raw files are saved (change folder path as needed)
cd .\Documents\CSB\Chance-Maritime\1-DataProcessing\

#Rename raw files to .wibl
Get-ChildItem * | Rename-Item -NewName { $ .Name + ".wibl" }

#Set directory to WIBL data-management scripts folder
```

```
cd .\Desktop\WIBL-main\wibl-python\scripts\data-management\  
  
#Activate environment  
conda activate wibl-python  
  
#Process .wibl to .geojson  
.\processWibl.ps1 [[-wiblPath] <string>] [[-metadataFile] <string>] [[-wiblConfig] <string>]  
  
#Validate .geojson against csbschema  
.\validateWibl.ps1 [[-inPath] <string>] [[-extension] GeoJSON]
```

processWibl.ps1: -wiblPath is the path to the folder where .wibl files are saved, -metadataFile is the path to the metadata file you created, and -wiblConfig is the path to the configuration file you created.

Example: `.\processWibl.ps1 -wiblPath C:\Users\schernoch\Documents\CSB\Chance-Maritime\1-DataProcessing\ -metadataFile C:\Users\schernoch\Documents\CSB\Chance-Maritime\0-Metadata\chance-maritime-metadata-complete.json -wiblConfig C:\Users\schernoch\Documents\CSB\Chance-Maritime\0-Metadata\configure-submission-USF.json`

validateWibl.ps1: -inPath is the folder where processed .geojson files are saved, -extension should be set to GeoJSON

Example: `.\validateWibl.ps1 -inPath C:\Users\schernoch\Documents\CSB\Chance-Maritime\1-DataProcessing\ -extension GeoJSON`

Information on the most up-to-date csbschema can be found [here](#). Once the .geojson files have been validated against csbschema, they are ready to be submitted to the DCDB.

There is also a new CSB validator tool written by Clint Campbell (available for download through [this link](#)) that allows CSB data to be validated by time stamp (data not collected from the future) and GPS (GPS location exists) to ensure data will actually be accepted into DCDB. This tool enables a CSB manager to validate not only that the data has something in every B-12 required section like the WIBL validator (e.g., there is a number in the “depth” category), but will actually reflect the DCDB’s data validation process and signal to practitioners if the data will be flagged in the system once submitted (e.g., the depth is 1E34, which is both beyond the transducer readable range and is in a format that cannot be utilized by DCDB, resulting in the submission being rejected).

Revision #9

Created 2026-01-14 21:56:33 UTC by Sophie Chernoch

Updated 2026-03-03 02:32:37 UTC by Sophie Chernoch