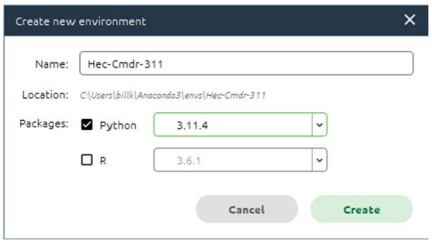
Quick Start Guide for RAS-Commander

Install Python using Anaconda Navigator

Download via https://www.anaconda.com/

Then, create a Python 3.11 Environment:

1. Open Anaconda Navigator
2. Environments > Create
3. Create Python 3.11 Environment

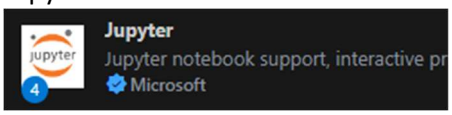


Install Visual Studio Code (VSCode) + Jupyter and Python Extensions

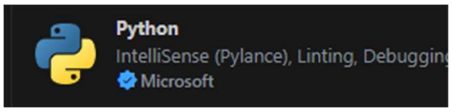
Download via https://code.visualstudio.com/Download

After installing, Install the following Visual Studio Code Extensions (Ctrl+Shift+X):

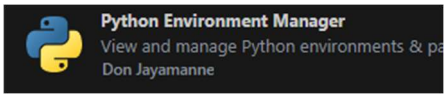
Jupyer:



Python:



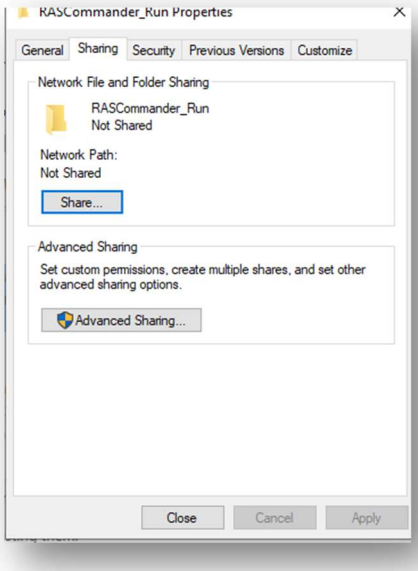
Python Environment Manager:



Create Local Windows File Shares to Support Remote Execution

Assuming you have a local workstation for remote execution, which you have already established working Remote Desktop and administrative privileges:

1. Log into the remote machine
2. Create a folder (Example: *C:\RASCommander_Run*)
3. Right click folder and go to "Properties"
4. Navigate to "Sharing" tab and click "Share"
5. Add read/write user permissions for each user or user group that will be executing runs remotely



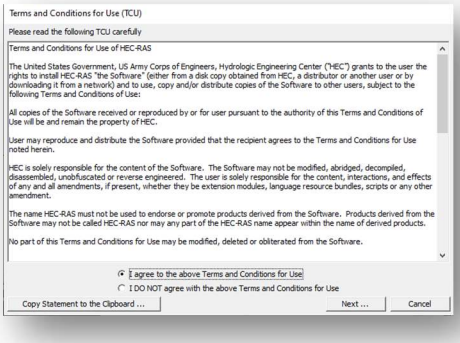
Note: When setting up multiple machines for remote execution, *ensure that each shared folder is placed at the same path on each machine, preferably outside of the user profile folders.*

Specific Windows Active Directory Security Policy and settings needed for Remote Desktop and PsExec is not within the scope of this document. It is suggested to set up a user group for HEC-RAS users that need elevated privileges on multiple machines to support this functionality.

Open HEC-RAS to Accept Terms and Conditions of Use for Each HEC-RAS Version

This is a necessary step, to prevent the program from hanging at first execute with a terms and conditions prompt. For each version of HEC-RAS you want to automate on a remote machine, the Terms and Conditions to be opened by at least one user of the machine and accepted, and HEC-RAS subsequently closed to force recordation of that acceptance.

Unfortunately, there is no direct command-line workaround for this step.



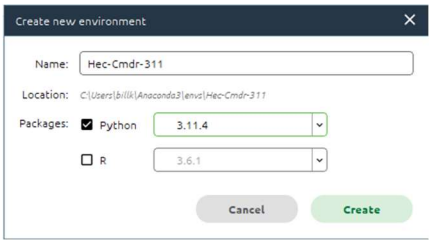
Quick Start Guide for HMS-Commander

Install Python using Anaconda Navigator

Download via https://www.anaconda.com/

Then, create a Python 3.11 Environment:

1. Open Anaconda Navigator
2. Environments > Create
3. Create Python 3.11 Environment

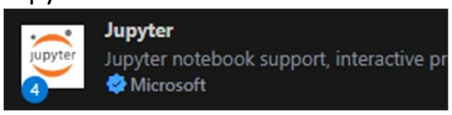


Install Visual Studio Code (VSCode) + Jupyter and Python Extensions

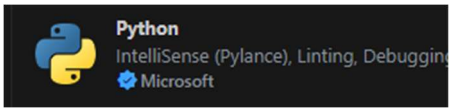
Download via https://code.visualstudio.com/Download

After installing, Install the following Visual Studio Code Extensions (Ctrl+Shift+X):

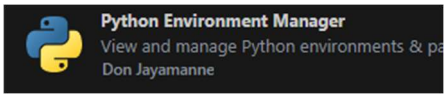
Jupyer:



Python:



Python Environment Manager:



Install Java Software Development Kit

Download latest version via https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.msi

NOTE: For HEC-HMS 4.9, JDK version 20.0.1 must be installed.

Install Jython

Download Jython Installer at https://www.jython.org/download.html

Install to the default location (C:\jython2.7.3)

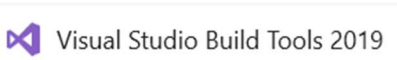
Quick Start Guide for DSS-Commander

Assuming VSCode and Anaconda are already installed,

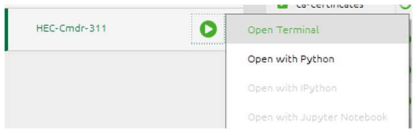
Install C++ Build Tools for Visual Studio 2019

Download Link: https://aka.ms/vs/17/release/vs_BuildTools.exe

When installing, select the "C++ Build Tools for Visual Studio 2019" option



1. Open a Terminal in the new environment



2. Install Required Dependencies with this command:

   "conda install pandas keyring chardet tqdm numpy affine bokeh ipykernel"

3. Download PyDss Wheel for Python 3.11 using web browser (skip this step if you don't need DSS-Commander)
   Direct Download Link:
   https://github.com/gyanz/pydsstools/blob/dcb25828ab32277cf72819362f68395a8564680a/dist/pydsstools-2.3-cp311-cp311-win_amd64.whl?raw=true

4. Install pydss wheel with this command in Anaconda terminal:

   "cd Downloads && pip install pydsstools-2.3-cp311-cp311-win_amd64.whl"