ThermoFisher
SCIENTIFIC

Paradigm SDK - Getting Started with Python

Preparation

If possible, install this on a clean PC (newly installed operating system), or in a Virtual Machine.

Overview

The basic steps that are needed before one can use the Paradigm SDK are:

1. Install the base OMNIC Paradigm software for the instrument to which you are connecting.
2. Install the Paradigm SDK Engine and Paradigm SDK Client modules.
3. (Optional but highly recommended) Set up a Python[1] virtual environment for development.
4. Install the paradigm-sdk pip package and dependencies into your Python environment.
5. Copy .NET config files next to your Python executable

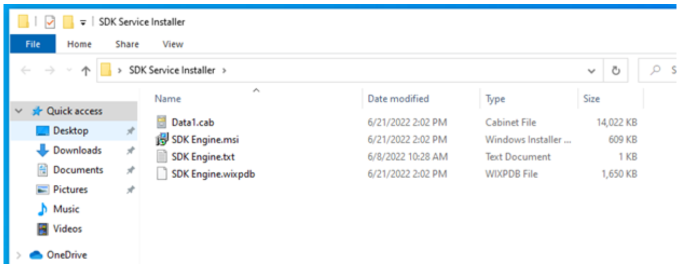
Install OMNIC Paradigm Software

See the OMNIC Paradigm software help topics on knowledge1.thermofisher.com for more information about installing and using OMNIC Paradigm software.

See the OMNIC Paradigm software help topics on knowledge1.thermofisher.com for more information about installing and using OMNIC Paradigm software.

Install the Paradigm SDK

Navigate to the SDK Engine Installer

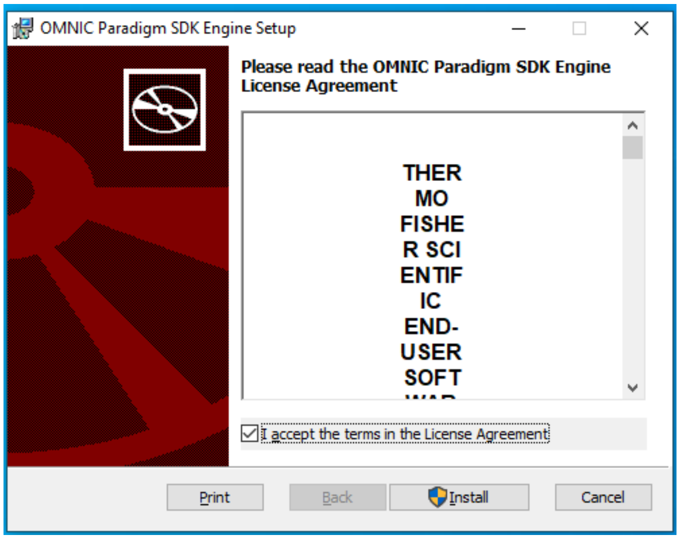


Name	Date modified	Type	Size
Data1.cab	6/21/2022 2:02 PM	Cabinet File	14,022 KB
SDK Engine.msi	6/21/2022 2:02 PM	Windows Installer ...	609 KB
SDK Engine.txt	6/8/2022 10:28 AM	Text Document	1 KB
SDK Engine.wixpdb	6/21/2022 2:02 PM	WIXPDB File	1,650 KB

Run the SDK Engine Installer

Accept the terms of the license agreement.

Install.



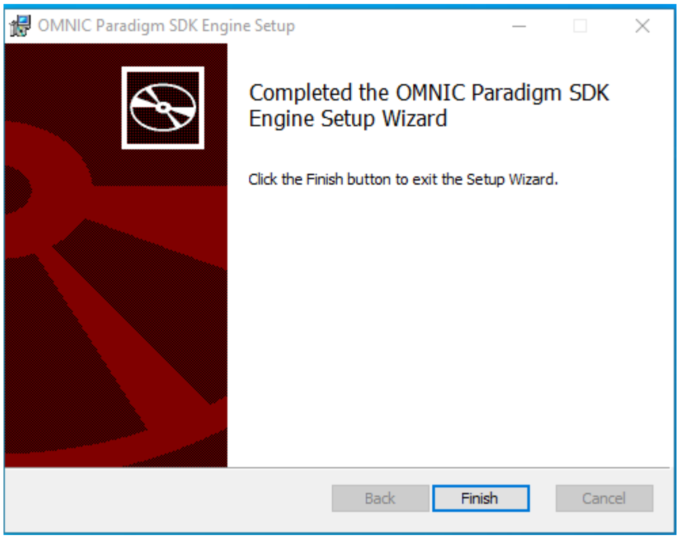
OMNIC Paradigm SDK Engine Setup

Please read the OMNIC Paradigm SDK Engine License Agreement

THER
MO
FISHE
R SCI
ENTIF
IC
END-
USER
SOFT

☑ I accept the terms in the License Agreement

Finish.



OMNIC Paradigm SDK Engine Setup

Completed the OMNIC Paradigm SDK Engine Setup Wizard

Click the Finish button to exit the Setup Wizard.

Now find the SDK Client Installer and take the same steps to finish SDK installation.

The SDK Documentation (including a copy of *this* document) is installed to C:\Users\Public\Documents\Thermo Scientific\SDK

Set up Python

Python is a widely used language so there are many different ways an SDK user might want to have it set up. The instructions in this section will be as general as possible to highlight the essential steps. Additionally, we provide a detailed, step-by-step example of how to set things up for Anaconda[2] in the Appendix at the end of this document. Anaconda is an open-source Python distribution platform used by many data scientists.

Choose a Python Version

The version of Python that can be used is restricted by the pip package used to communicate with the Paradigm SDK client module. This package, called pythonnet, is required because we need to bridge the gap between the Python interpreter and the .NET Common Language Runtime (CLR) that is used by the C# SDK client module. The pythonnet package is compatible with Python 3.8 for its last stable version or 3.10 if using the alpha version. Our recommendation is to use Python 3.10 because Python 3.8 is already close to end of life (bug fix updates ended May 2021, security updates will end in October 2024).

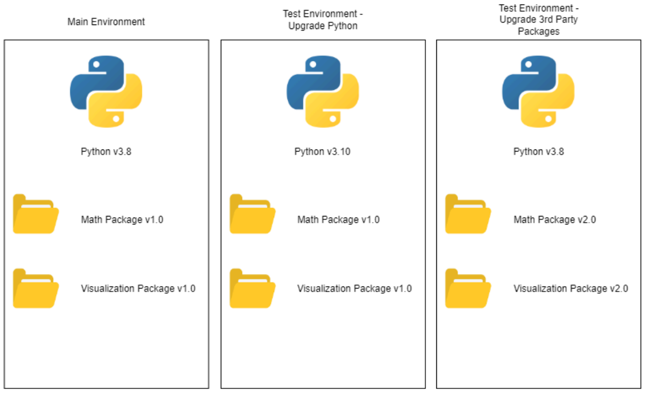
Download Python

Python can be downloaded for free from the python.org website: https://www.python.org/downloads/

Create a Virtual Environment

Using a virtual environment in Python is a common development practice that can make the process of developing your software easier. Essentially, the virtual environment specifies which version of Python is running and what pip packages are installed.

By having a virtual environment, the developer can quickly create a new virtual environment and test installing different packages, running their program using different versions of Python, and many other things. Once the test or investigation is complete, the developer can return to their main environment without needing to undo the actions taken during the test.


Main Environment
Test Environment - Upgrade Python
Test Environment - Upgrade 3rd Party Packages
Python v3.8
Math Package v1.0
Visualization Package v1.0
Python v3.10
Math Package v1.0
Visualization Package v1.0
Python v3.8
Math Package v2.0
Visualization Package v2.0

Another benefit is that virtual environments isolate your programs from other Python programs running on the same computer. For example, the Thermo Fisher Cloud Connector service uses a certain version of Python (e.g. 3.8) that is added to the Windows PATH variable. As a result, Python programs that run without a virtual environment will default to using that version rather than the latest version that you've downloaded.

If you are using Python with an IDE or platform like Anaconda, there is often built-in functionality that will do most of the work of setting up the virtual environment for you. Follow the instructions specific to that platform rather than running the batch file mentioned below.

The Python language comes with a package for creating virtual environments called venv. Included in your installation of the Paradigm SDK is a Windows batch file `create_venv.bat` that serves as an example of how to create a virtual environment. It also copies the Python files for the paradigm-sdk package and the sample_client.py into the virtual environment's Scripts folder. Make sure to check the file "bat_errors.txt" to see if any errors occurred.

Install the paradigm-sdk Pip Package

There is a pip package that is included with your SDK installation to be installed into your Python environment. The file "install_paradigm_sdk.bat" executes the steps necessary. Those steps are:

1. Install the "build" package using "pip install –upgrade build"
2. Install the alpha version of the pythonnet package using "pip install –upgrade pythonnet==3.0.0a2"
3. Install the paradigm-sdk package using "pip install <filepath to paradigm_sdk folder>"

If running the .bat file, make sure to check the file "bat_errors.txt" to see if any errors occurred.

Copy in the .NET Config Files

The SDK Client module requires an ".exe.config" file in order to find all the .dll files it needs. When writing an application in a .NET language (e.g. C#, VB.NET, etc.), this file is automatically generated and included with the final program. Since Python is not one of these languages, we need to manually provide this file and place it next to the Python executable that is running our code.

Included in your installation are two files: "python.exe.config" and "pythonw.exe.config". Copy both of these files into the folder containing the Python executable that is running your program. The two files are for the two different executables used by Python: python.exe is used for command-line applications, pythonw.exe is used for all other programs like GUI applications or automated scripts with no UI.

An easy method of finding your Python executable is to use these lines of Python code:

```
import sys
print(sys.executable)
quit()
```

Note that in this example python.exe is located at:

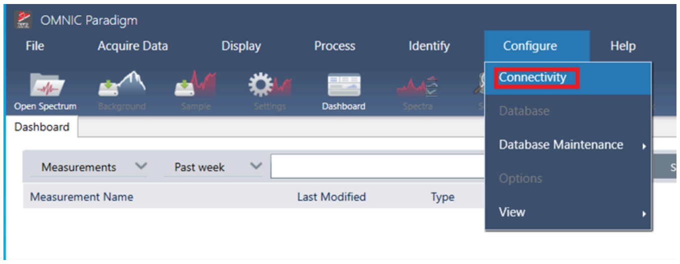
`"C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe"`

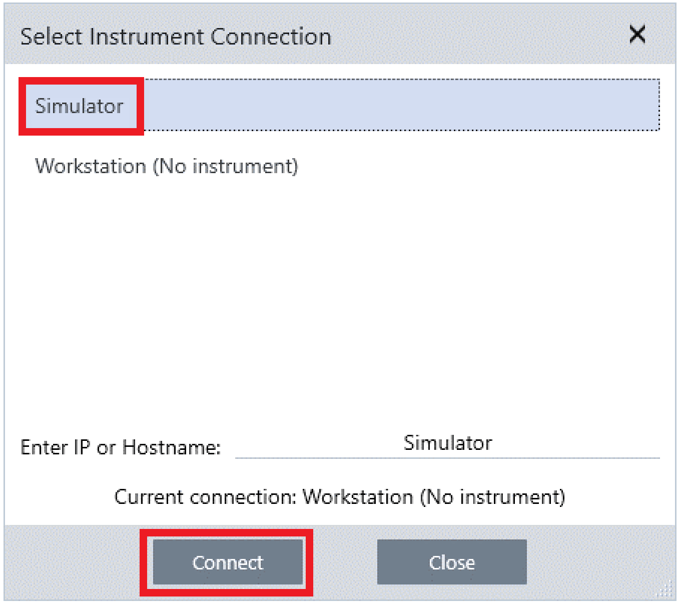
Configure OMNIC Paradigm software

Start OMNIC Paradigm from the desktop

Simulator (Optional)

If you do not have an instrument, you can use the simulator:



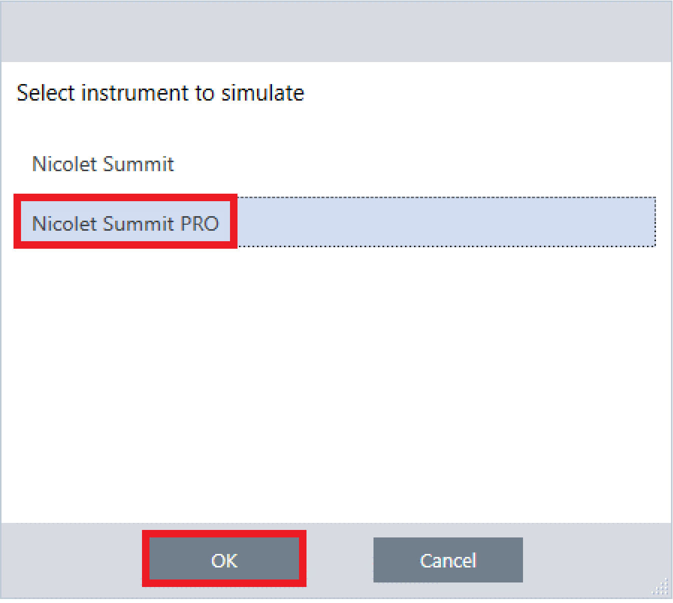

Select Instrument Connection

Simulator

Workstation (No instrument)

Enter IP or Hostname: Simulator

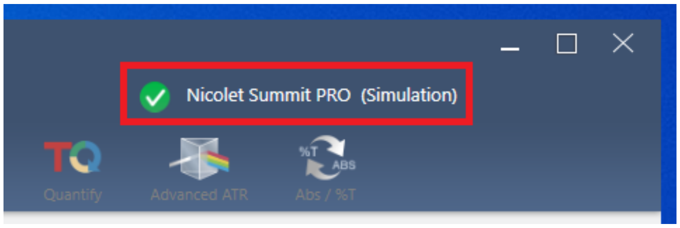
Current connection: Workstation (No instrument)



Select instrument to simulate

Nicolet Summit

Nicolet Summit PRO



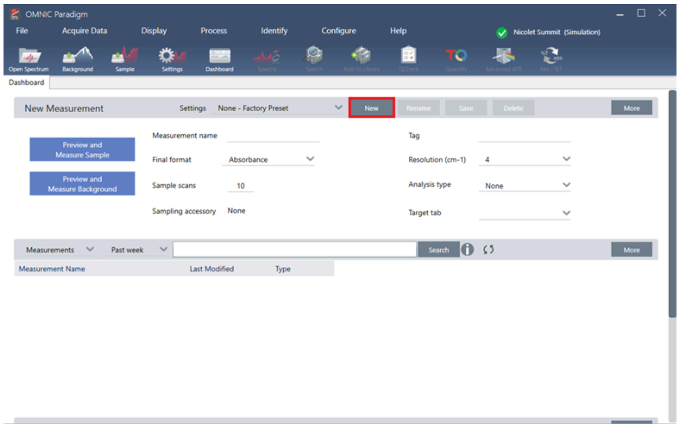
Nicolet Summit PRO (Simulation)

Notice that Paradigm indicates it is now simulating an instrument:

Configuring Named Parameter Sets

The SDK will use named parameter sets from OMNIC Paradigm software. We will create one now named “Example Parameters”.

Select New



OMNIC Paradigm

Nicolet Summit (Simulation)

New Measurement

Settings: None - Factory Preset

Measurement name:

Final format: Absorbance

Sample scans: 10

Sampling accessory: None

Tag:

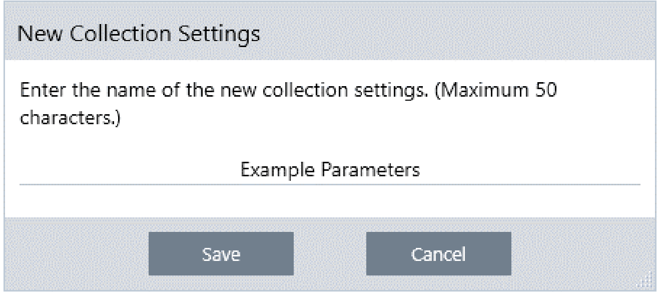
Resolution (cm-1): 4

Analysis type: None

Target tab:

Measurement Name	Last Modified	Type

Enter the name “Example Parameters” and select save

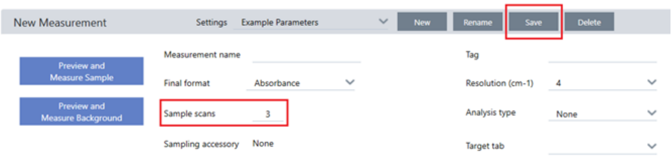


New Collection Settings

Enter the name of the new collection settings. (Maximum 50 characters.)

Example Parameters

Make any changes to the Parameters that you want. For example, change sample scans from 10 to 3. Notice that the Save button is now active. Save the changes.



New Measurement

Settings: Example Parameters

Measurement name:

Final format: Absorbance

Sample scans: 3

Sampling accessory: None

Tag:

Resolution (cm-1): 4

Analysis type: None

Target tab:

USE THE SDK

We have provided an example python application that acquires spectra. It is located in this folder:

`“C:\Users\Public\Documents\thermo scientific\SDK\Python”`

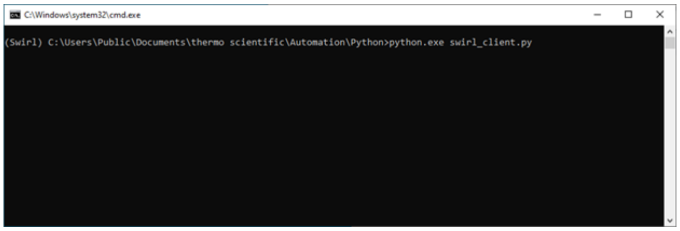
It is named *sample_client.py*

Go to:

`“C:\Users\Public\Documents\thermo scientific\SDK\Python using the ExampleEnv command window.”`

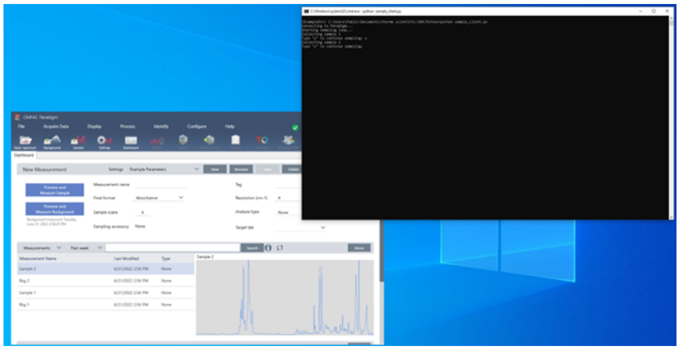
Run:

```
$ python.exe sample_client.py
```



C:\Windows\system32\cmd.exe

```
(Swirl) C:\Users\Public\Documents\thermo scientific\Automation\Python>python.exe swirl_client.py
```



```
(ExampleEnv) C:\Users\Public\Documents\Thermo Scientific\SDK\Python>python sample_client.py
Connecting to Paradigm...
Starting sampling loop...
Collecting sample 1
Type "y" to continue sampling: y
Collecting sample 2
Type "y" to continue sampling:
```

Appendix

Troubleshooting Tips

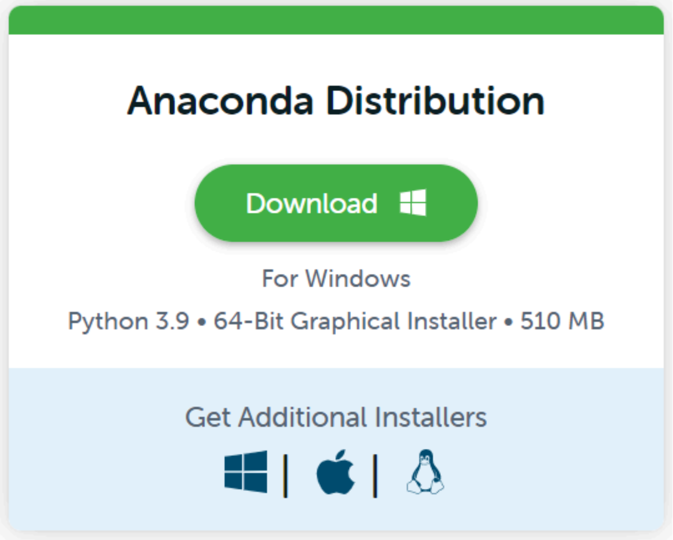
- Be sure to check that the version of Python you are running is what you expect. Follow the steps outlined in “Copy in the .NET Config Files” to see which executable is running.
- If you are getting the error message “System.IO.FileNotFoundException: Unable to find assembly ‘ParadigmLib’”
  - Check that the SDK .dll’s are installed into C:\Program Files\Thermo Scientific\SDK Client
- If you are getting an error message like “System.IO.FileNotFoundException: Could not load file or assembly ‘Microsoft.Extensions.Logging, Version=2.0.0.0, Culture=neutral, PublicKeyToken=adb9793829ddae60’ or one of its dependencies. The system cannot find the file specified.”
  - Check that the python.exe.config and pythonw.exe.config files are present in same directory as the Python executable you are using.

Install Anaconda Detailed

Follow the link

Anaconda - https://www.anaconda.com/products/distribution

Download Anaconda



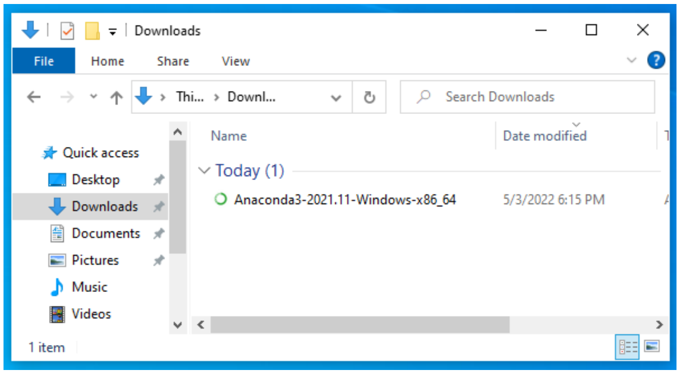
Anaconda Distribution

For Windows

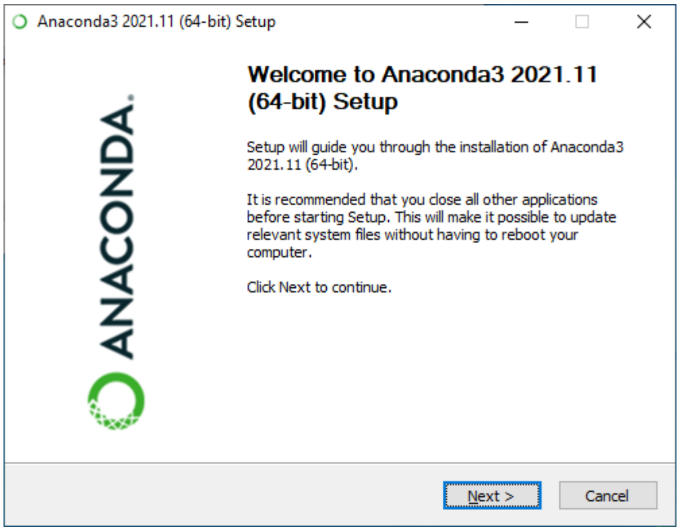
Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers

Run the installer



Name	Date modified
Today (1)	
Anaconda3-2021.11-Windows-x86_64	5/3/2022 6:15 PM



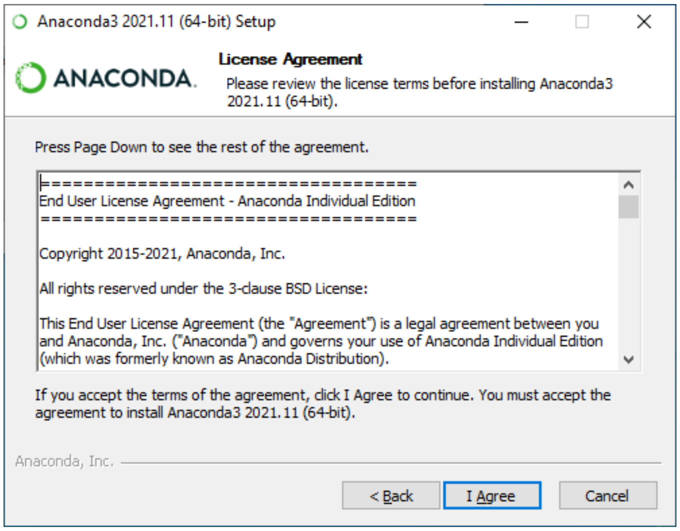
Anaconda3 2021.11 (64-bit) Setup

Welcome to Anaconda3 2021.11 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2021.11 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



License Agreement

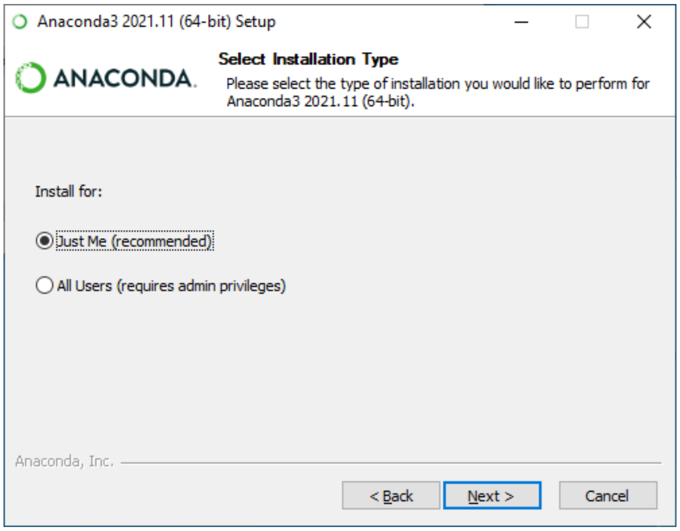
Please review the license terms before installing Anaconda3 2021.11 (64-bit).

Press Page Down to see the rest of the agreement.

```
===================================
End User License Agreement - Anaconda Individual Edition
===================================

Copyright 2015-2021, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you
and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Individual Edition
(which was formerly known as Anaconda Distribution).
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2021.11 (64-bit).

Anaconda, Inc.



Select Installation Type

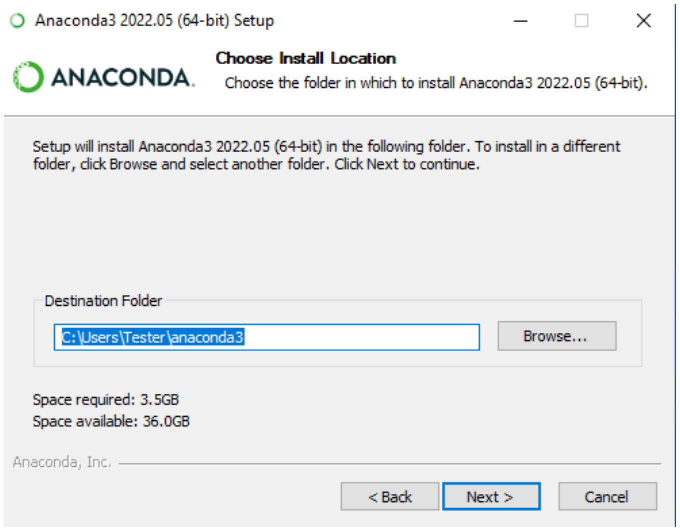
Please select the type of installation you would like to perform for Anaconda3 2021.11 (64-bit).

Install for:

● Just Me (recommended)

○ All Users (requires admin privileges)

Anaconda, Inc.



Choose Install Location

Choose the folder in which to install Anaconda3 2022.05 (64-bit).

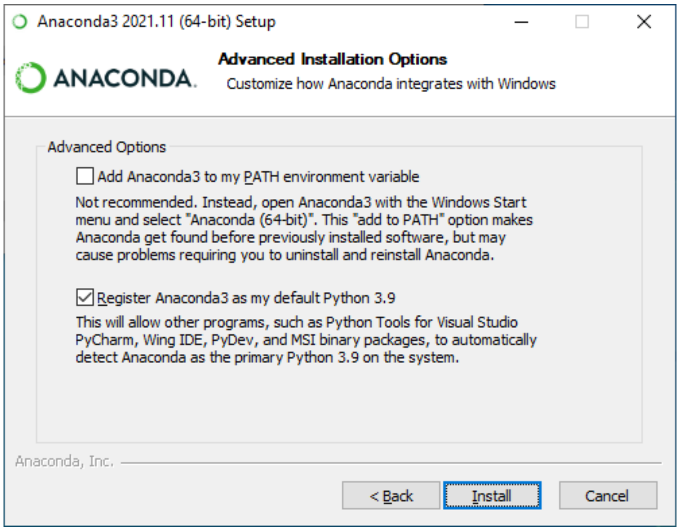
Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Users\Tester\anaconda3

Space required: 3.5GB
Space available: 36.0GB

Anaconda, Inc.



Advanced Installation Options

Customize how Anaconda integrates with Windows

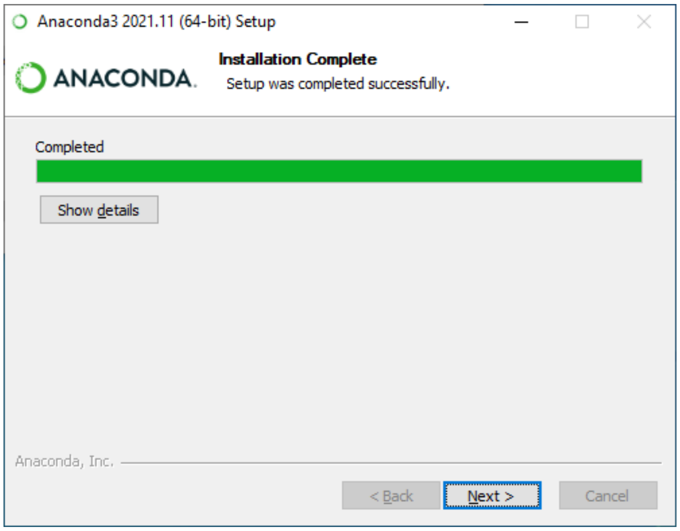
Advanced Options

□ Add Anaconda3 to my PATH environment variable

Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda3 as my default Python 3.9

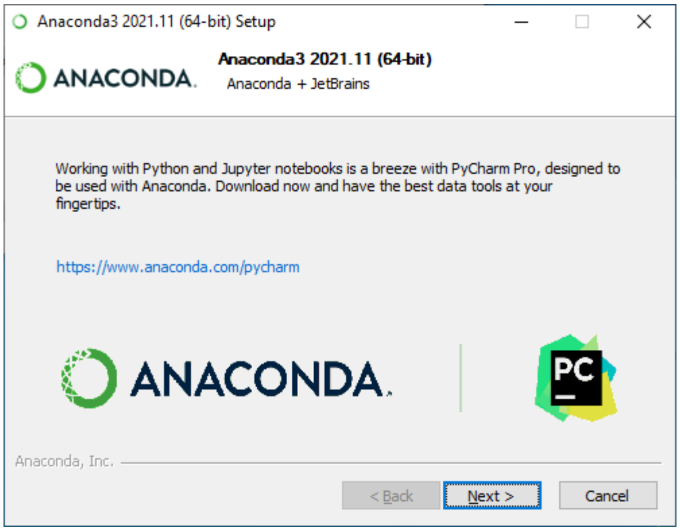
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.



Installation Complete

Setup was completed successfully.

Completed

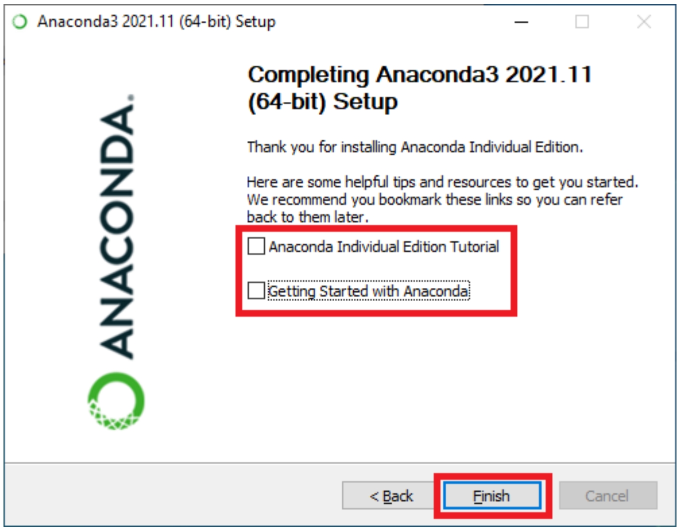


Anaconda3 2021.11 (64-bit)

Anaconda + JetBrains

Working with Python and Jupyter notebooks is a breeze with PyCharm Pro, designed to be used with Anaconda. Download now and have the best data tools at your fingertips.

https://www.anaconda.com/pycharm



Anaconda3 2021.11 (64-bit) Setup

Completing Anaconda3 2021.11 (64-bit) Setup

Thank you for installing Anaconda Individual Edition.

Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

□ Anaconda Individual Edition Tutorial

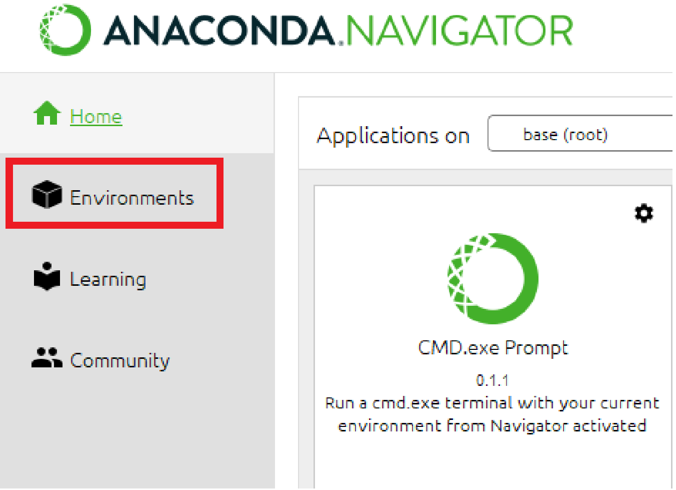
□ Getting Started with Anaconda

Configure Anaconda Detailed

Run the Anaconda Navigator

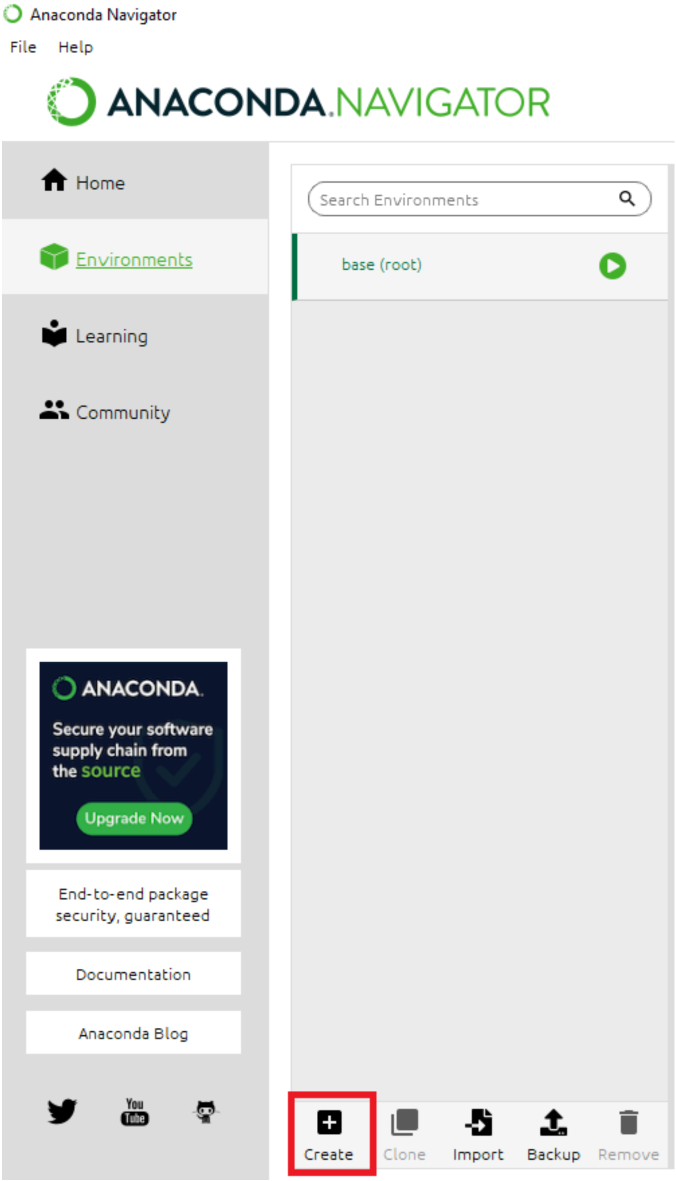
Create a Virtual Environment

Select Environments

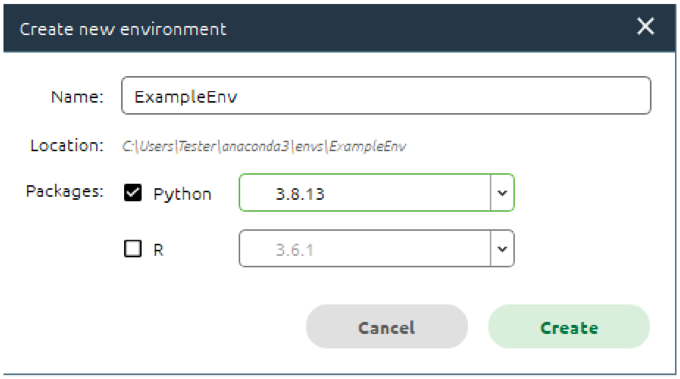


Run a cmd.exe terminal with your current environment from Navigator activated

Select Create



Fill in the name and select the version of Python you will be using. Create the environment.



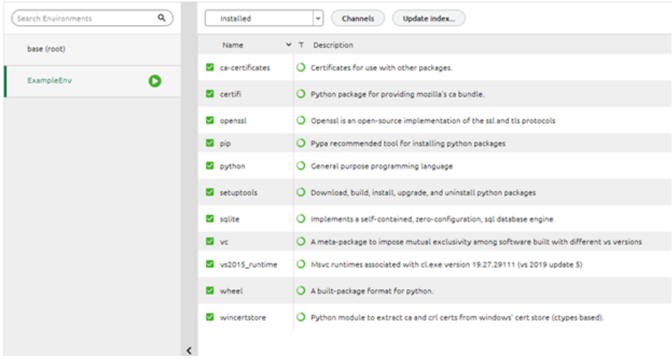
Name: ExampleEnv

Location: C:\Users\Tester\anaconda3\envs\ExampleEnv

Packages: ☑ Python 3.8.13

☐ R 3.6.1

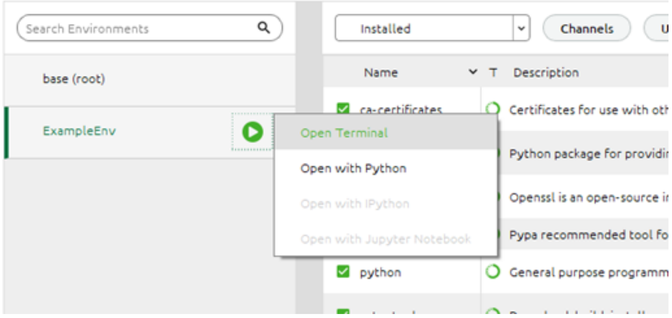
You now have a virtual environment with the following packages installed:



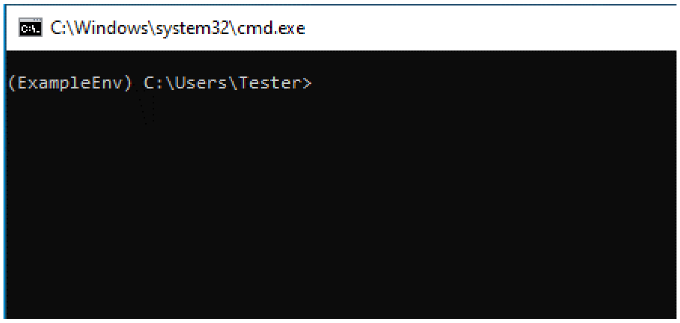
Name	T	Description
☑ ca-certificates	○	Certificates for use with other packages.
☑ certifi	○	Python package for providing mozilla's ca bundle.
☑ openssl	○	Openssl is an open-source implementation of the ssl and tls protocols
☑ pip	○	Pypa recommended tool for installing python packages
☑ python	○	General purpose programming language
☑ setuptools	○	Download, build, install, upgrade, and uninstall python packages
☑ sqlite	○	Implements a self-contained, zero-configuration, sql database engine
☑ vc	○	A meta-package to impose mutual exclusivity among software built with different vs versions
☑ vs2015_runtime	○	Msvc runtimes associated with cl.exe version 19.27.29111 (vs 2019 update 5)
☑ wheel	○	A built-package format for python.
☑ wincertstore	○	Python module to extract ca and crl certs from windows' cert store (ctypes based).

Set up the Config files for Python

Open a terminal for your virtual environment



Notice the (ExampleEnv) name in the command prompt.



```
(ExampleEnv) C:\Users\Tester>
```

Determine where python is installed:

In the command prompt enter:

> ```
> $ python
> ```

Python is now running in the command prompt

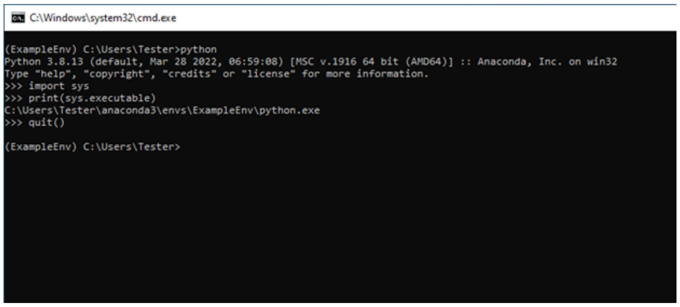
Enter each line of code below:

> ```
> import sys
> print(sys.executable)
> ```

```
quit()
```

Note that in this example python.exe is located at:

“C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe"



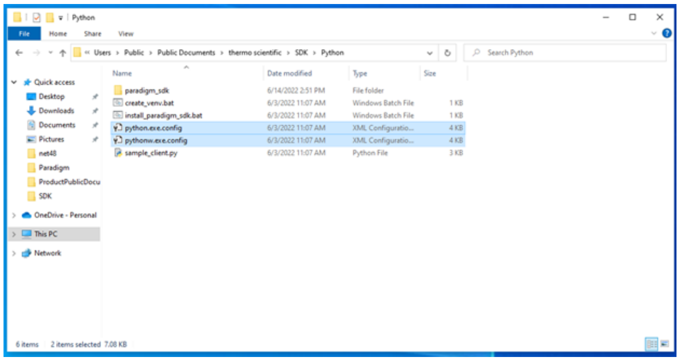
C:\Windows\system32\cmd.exe

```
(ExampleEnv) C:\Users\Tester>python
Python 3.8.13 (default, Mar 28 2022, 06:59:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> print(sys.executable)
C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe
>>> quit()

(ExampleEnv) C:\Users\Tester>
```

Go to:

“C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe”.

Find the two .config files there and copy them to the clipboard.



« Users › Public › Public Documents › thermo scientific › SDK › Python

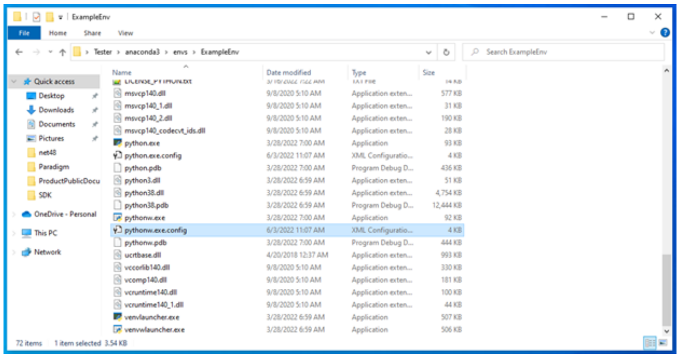
Name	Date modified	Type	Size
paradigm_sdk	6/14/2022 2:51 PM	File folder	
create_venv.bat	6/3/2022 11:07 AM	Windows Batch File	1 KB
install_paradigm_sdk.bat	6/3/2022 11:07 AM	Windows Batch File	1 KB
python.exe.config	6/3/2022 11:07 AM	XML Configuratio...	4 KB
pythonw.exe.config	6/3/2022 11:07 AM	XML Configuratio...	4 KB
sample_client.py	6/3/2022 11:07 AM	Python File	3 KB

6 items 2 items selected 7.08 KB

Go to the python.exe install folder. In this example it is

“C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe”.

Copy the two config files to this folder.



› Tester › anaconda3 › envs › ExampleEnv

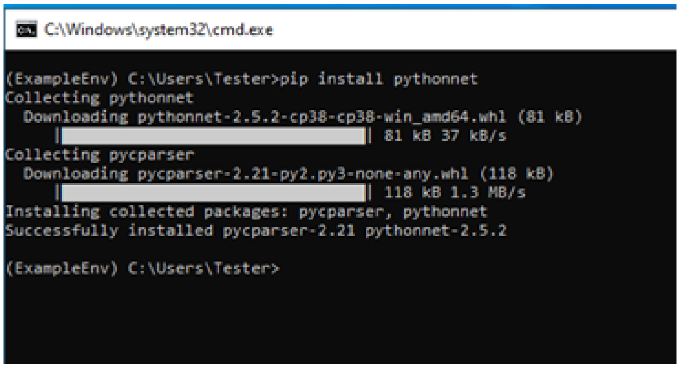
Name	Date modified	Type	Size
msvcp140.dll	9/8/2020 5:10 AM	Application exten...	577 KB
msvcp140_1.dll	9/8/2020 5:10 AM	Application exten...	31 KB
msvcp140_2.dll	9/8/2020 5:10 AM	Application exten...	190 KB
msvcp140_codecvt_ids.dll	9/8/2020 5:10 AM	Application exten...	28 KB
python.exe	3/28/2022 7:00 AM	Application	93 KB
python.exe.config	6/3/2022 11:07 AM	XML Configuratio...	4 KB
python.pdb	3/28/2022 7:00 AM	Program Debug D...	436 KB
python3.dll	3/28/2022 6:59 AM	Application exten...	51 KB
python38.dll	3/28/2022 6:59 AM	Application exten...	4,754 KB
python38.pdb	3/28/2022 6:59 AM	Program Debug D...	12,444 KB
pythonw.exe	3/28/2022 7:00 AM	Application	92 KB
pythonw.exe.config	6/3/2022 11:07 AM	XML Configuratio...	4 KB
pythonw.pdb	3/28/2022 7:00 AM	Program Debug D...	444 KB
ucrtbase.dll	4/20/2018 12:37 AM	Application exten...	993 KB
vccorlib140.dll	9/8/2020 5:10 AM	Application exten...	330 KB
vcomp140.dll	9/8/2020 5:10 AM	Application exten...	181 KB
vcruntime140.dll	9/8/2020 5:10 AM	Application exten...	100 KB
vcruntime140_1.dll	9/8/2020 5:10 AM	Application exten...	44 KB
venvlauncher.exe	3/28/2022 6:59 AM	Application	507 KB
venvwlauncher.exe	3/28/2022 6:59 AM	Application	506 KB

72 items 1 item selected 3.54 KB

Install Python Support Files

Install pythonnet:

```
$ pip install pythonnet
```



```
(ExampleEnv) C:\Users\Tester>pip install pythonnet
Collecting pythonnet
  Downloading pythonnet-2.5.2-cp38-cp38-win_amd64.whl (81 kB)
     |████████████████████████████████| 81 kB 37 kB/s
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
     |████████████████████████████████| 118 kB 1.3 MB/s
Installing collected packages: pycparser, pythonnet
Successfully installed pycparser-2.21 pythonnet-2.5.2

(ExampleEnv) C:\Users\Tester>
```

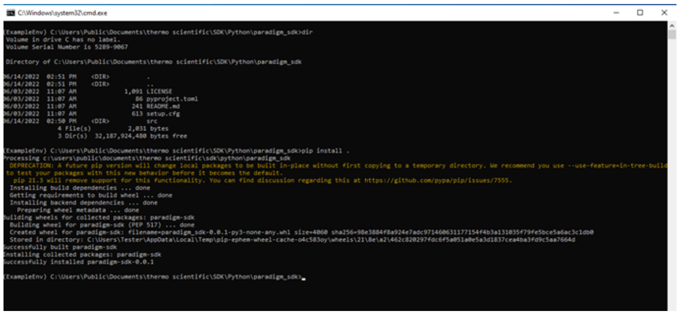
Navigate to:

"C:\Users\Tester\anaconda3\envs\ExampleEnv\python.exe"

Install the paradigm sdk:

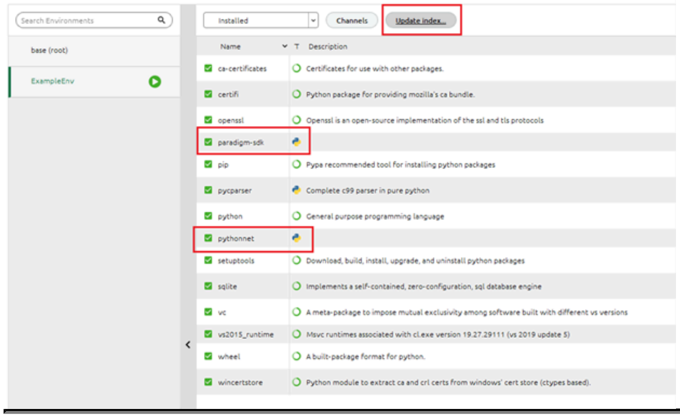
```
$ pip install .
```

You can ignore the deprecation warning. Notice that final line says that paradigm-sdk was successfully installed.



```
(ExampleEnv) C:\Users\Public\Documents\thermo scientific\SDK\Python\paradigm_sdk>dir
 Volume in drive C has no label.
 Volume Serial Number is 5289-9067

 Directory of C:\Users\Public\Documents\thermo scientific\SDK\Python\paradigm_sdk

06/14/2022  02:51 PM    <DIR>          .
06/14/2022  02:51 PM    <DIR>          ..
06/03/2022  11:07 AM             1,091 LICENSE
06/03/2022  11:07 AM                86 pyproject.toml
06/03/2022  11:07 AM               241 README.md
06/03/2022  11:07 AM               613 setup.cfg
06/14/2022  02:50 PM    <DIR>          src
               4 File(s)          2,031 bytes
               3 Dir(s)  32,187,924,480 bytes free

(ExampleEnv) C:\Users\Public\Documents\thermo scientific\SDK\Python\paradigm_sdk>pip install .
Processing c:\users\public\documents\thermo scientific\sdk\python\paradigm_sdk
  DEPRECATION: A future pip version will change local packages to be built in-place without first copying to a temporary directory. We recommend you use --use-feature=in-tree-build
 to test your packages with this new behavior before it becomes the default.
   pip 21.3 will remove support for this functionality. You can find discussion regarding this at https://github.com/pypa/pip/issues/7555.
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
    Preparing wheel metadata ... done
Building wheels for collected packages: paradigm-sdk
  Building wheel for paradigm-sdk (PEP 517) ... done
  Created wheel for paradigm-sdk: filename=paradigm_sdk-0.0.1-py3-none-any.whl size=4060 sha256=98e3884f8a924e7adc9714606311177154f4b3a131035f79fe5bce5a6ac3c1db0
  Stored in directory: C:\Users\Tester\AppData\Local\Temp\pip-ephem-wheel-cache-o4c583oy\wheels\21\8e\a2\462c820297fdc6f5a051a0e5a3d1837cea4ba3fd9c5aa7664d
Successfully built paradigm-sdk
Installing collected packages: paradigm-sdk
Successfully installed paradigm-sdk-0.0.1

(ExampleEnv) C:\Users\Public\Documents\thermo scientific\SDK\Python\paradigm_sdk>
```

Return to the Anaconda navigator and update the index. Notice that pythonnet and paradigm-sdk are now installed:



	Name	T	Description
☑	ca-certificates	○	Certificates for use with other packages.
☑	certifi	○	Python package for providing mozilla's ca bundle.
☑	openssl	○	Openssl is an open-source implementation of the ssl and tls protocols
☑	paradigm-sdk		
☑	pip	○	Pypa recommended tool for installing python packages
☑	pycparser		Complete c99 parser in pure python
☑	python	○	General purpose programming language
☑	pythonnet		
☑	setuptools	○	Download, build, install, upgrade, and uninstall python packages
☑	sqlite	○	Implements a self-contained, zero-configuration, sql database engine
☑	vc	○	A meta-package to impose mutual exclusivity among software built with different vs versions
☑	vs2015_runtime	○	Msvc runtimes associated with cl.exe version 19.27.29111 (vs 2019 update 5)
☑	wheel	○	A built-package format for python.
☑	wincertstore	○	Python module to extract ca and crl certs from windows' cert store (ctypes based).

1. Python is a registered trademark of the Python Software Foundation.↩?

2. Anaconda, the Anaconda Logo, Conda, the Conda Logo and Miniconda are registered trademarks of Anaconda Inc. Nucleus is a trademark of Anaconda Inc.↩?