Axiometrix Solutions
9290 SW Nimbus Avenue
Beaverton, OR 97008
www.axiometrixsolutions.com

Tel: 503-627-0832
Toll Free: 800-231-7350

Controlling APx500 Using Python

When writing a script to control the APx500 software, the APx500 API library must be referenced. Because this library is built into a .NET assembly file, the standard Python implementation cannot be used; support for .NET libraries must be added by installing the Python.NET module. Autocomplete in Visual Studio Code can also optionally be enabled via the Python.NET Stub Generator.

To get started controlling APx500 using Python, follow the steps below.

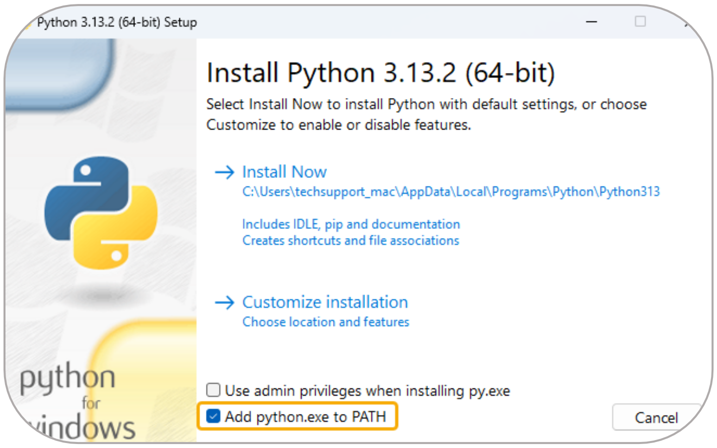
> Note that although Python is a cross-platform language, API control of APx500 is only available on operating systems supported by the APx500 application.

Step 1: Install Python

Before installing Python, check which versions are currently supported by Python.NET using the Python.NET GitHub repository. As of March 19th, 2025, supported versions are 3.7 through 3.13.

python 3.7 | 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | 3.13

Once compatibility has been confirmed, download and install the desired version of python. Make sure to check the box to add Python to the Windows PATH.



Python 3.13.2 (64-bit) Setup

Install Python 3.13.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\techsupport_mac\AppData\Local\Programs\Python\Python313

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☐ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

Step 2: Install Python.NET Using a Package Manager

To install Python.NET using pip, a package manager for Python:

1. Open a Windows Command Prompt.
2. Type “`pip install pythonnet`”.

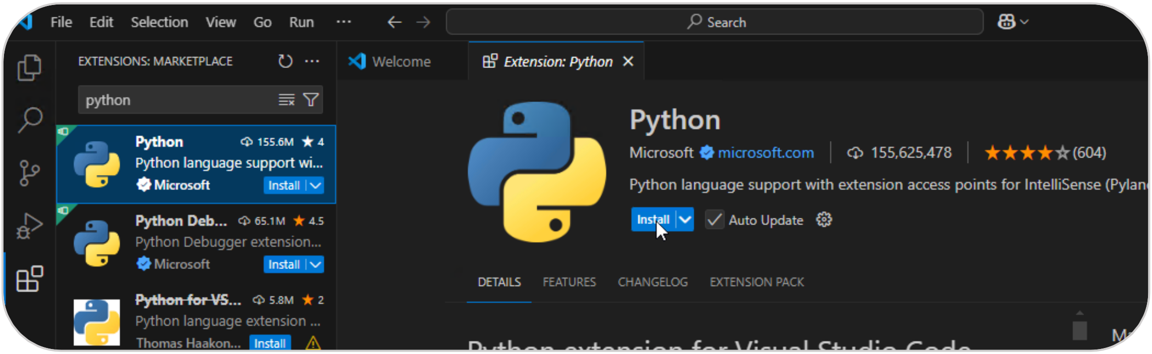
And that’s it! You are now ready to write code to control APx500 using Python and Python.NET in your preferred development environment.

Step 3 (Optional): Enable Autocomplete in Visual Studio Code

To enable autocomplete in Visual Studio Code, follow these steps:

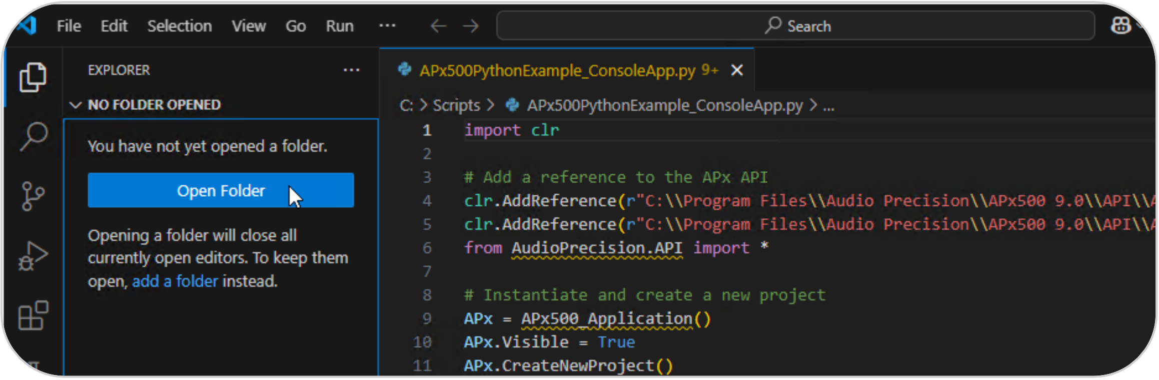
1. Install Visual Studio Code
   Download and install Visual Studio Code.

2. Install the Python Extension
   Add the Microsoft Python extension to Visual Studio Code.



Python language support with extension access points for IntelliSense (Pylan

3. From the Explorer tab, click the "Open Folder" button and select the directory containing your script.


File
Edit
Selection
View
Go
Run
Search
EXPLORER
NO FOLDER OPENED
Open Folder
APx500PythonExample_ConsoleApp.py 9+
C: > Scripts > APx500PythonExample_ConsoleApp.py > ...

You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

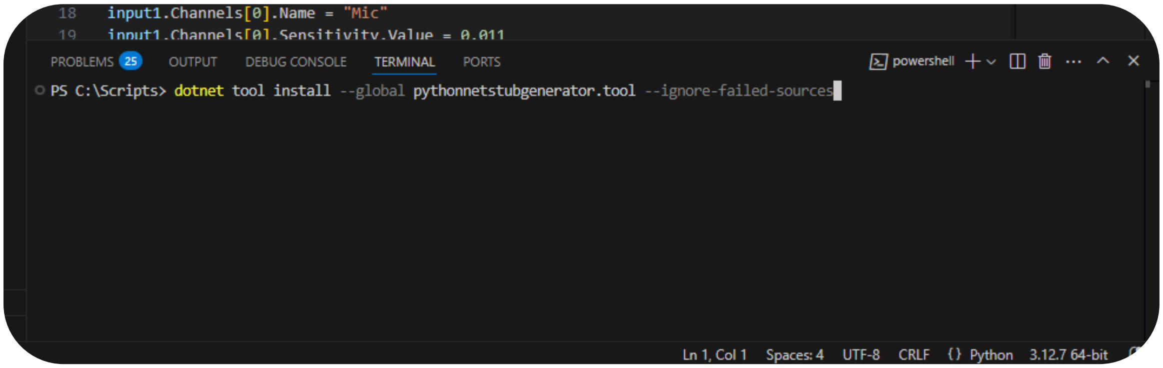
```
import clr

# Add a reference to the APx API
clr.AddReference(r"C:\\Program Files\\Audio Precision\\APx500 9.0\\API\\A
clr.AddReference(r"C:\\Program Files\\Audio Precision\\APx500 9.0\\API\\A
from AudioPrecision.API import *

# Instantiate and create a new project
APx = APx500_Application()
APx.Visible = True
APx.CreateNewProject()
```

4. Install the Python.NET Stub Generator
   Open the Visual Studio Code Terminal (enable from the View->Terminal if necessary) and install the pythonnet-stub-generator by running the following command:

```
dotnet tool install --global pythonnetstubgenerator.tool --ignore-failed-sources
```



```
input1.Channels[0].Name = "Mic"
input1.Channels[0].Sensitivity.Value = 0.011
```

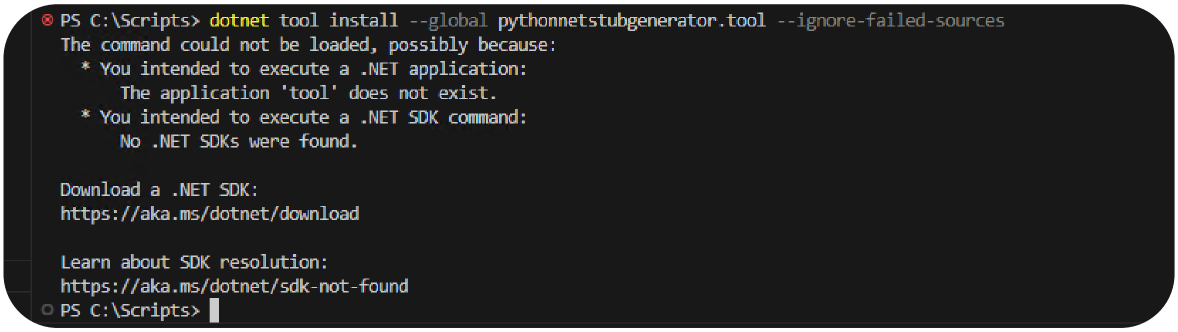
PROBLEMS 25
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
Ln 1, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.12.7 64-bit

```
PS C:\Scripts> dotnet tool install --global pythonnetstubgenerator.tool --ignore-failed-sources
```

When the stub generator successfully installs, you should see a message like the following:

Tool 'pythonnetstubgenerator.tool' (version '1.2.1') was successfully installed.

If an error related to “No .NET SDKs” is shown, or the dotnet command is not found, download and install the.NET SDK, then restart Visual Studio Code and retry the dotnet tool install.



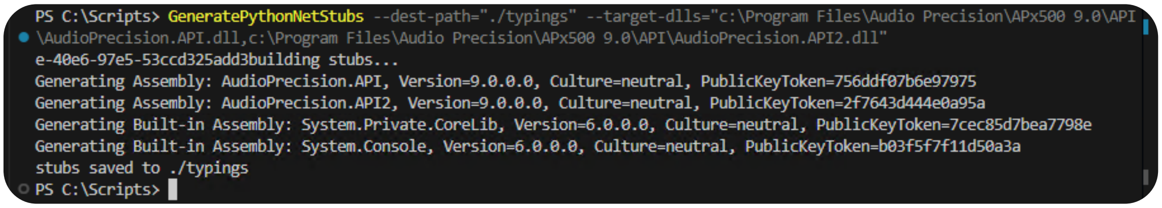
```
PS C:\Scripts> dotnet tool install --global pythonnetstubgenerator.tool --ignore-failed-sources
The command could not be loaded, possibly because:
  * You intended to execute a .NET application:
      The application 'tool' does not exist.
  * You intended to execute a .NET SDK command:
      No .NET SDKs were found.

Download a .NET SDK:
https://aka.ms/dotnet/download

Learn about SDK resolution:
https://aka.ms/dotnet/sdk-not-found
PS C:\Scripts>
```

5. **Generate API Stubs**
   To generate the stubs needed for autocomplete (into sub-directory named “typings”), run the command below, updating the highlighted areas to match the version of APx500 software you are controlling from the API:

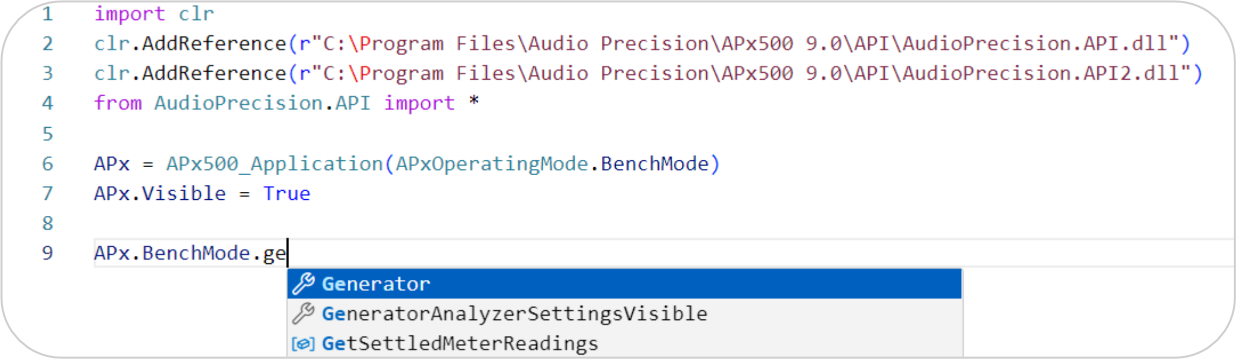
   GeneratePythonNetStubs --dest-path="./typings" –target-dlls="c:\Program Files\Audio Precision\APx500 9.1\API\AudioPrecision.API.dll,c:\Program Files\Audio Precision\APx500 9.1\API\AudioPrecision.API2.dll"



```
PS C:\Scripts> GeneratePythonNetStubs --dest-path="./typings" --target-dlls="c:\Program Files\Audio Precision\APx500 9.0\API
\AudioPrecision.API.dll,c:\Program Files\Audio Precision\APx500 9.0\API\AudioPrecision.API2.dll"
e-40e6-97e5-53ccd325add3building stubs...
Generating Assembly: AudioPrecision.API, Version=9.0.0.0, Culture=neutral, PublicKeyToken=756ddf07b6e97975
Generating Assembly: AudioPrecision.API2, Version=9.0.0.0, Culture=neutral, PublicKeyToken=2f7643d444e0a95a
Generating Built-in Assembly: System.Private.CoreLib, Version=6.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e
Generating Built-in Assembly: System.Console, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
stubs saved to ./typings
PS C:\Scripts>
```

Note that stubs will need to be re-generated (or copied over) each time you start a python script in a new directory, or if the version of APx500 API you reference in your script changes.

If setup was successful and the APx500 API is properly referenced in your script, auto-completion will now appear as you type APIs in Visual Studio Code.



```
import clr
clr.AddReference(r"C:\Program Files\Audio Precision\APx500 9.0\API\AudioPrecision.API.dll")
clr.AddReference(r"C:\Program Files\Audio Precision\APx500 9.0\API\AudioPrecision.API2.dll")
from AudioPrecision.API import *

APx = APx500_Application(APxOperatingMode.BenchMode)
APx.Visible = True

APx.BenchMode.ge
```

Example Script

Included with this guide are two example python scripts: one for a console-style application, and another that utilizes Windows Forms to display a simple user interface. These scripts were created and tested with version 3.13.2 of Python and 3.0.3 of Python.NET. They demonstrate how to:

- Reference the Audio Precision API
- Instantiate the Apx500 software
- Create a new project
- Configure Signal Path Setup
- Add and configure a measurement
- Run a sequence
- Get data acquired in the sequence.

For more examples, please contact the Audio Precision Technical Support team using the contact information on our website.