



Automating any of Copper Mountain Technology's VNAs in Python follows a similar structure to the automation procedure for other environments; the key step lies in setting up the COM server connection. The following example is based on Python 3.4, though later versions of Python involve a similar procedure.

Before starting, install the VNA software application and ensure that the COM server for the VNA you are using is registered during installation. The latest version of the VNA software is always available for download at www.coppermountaintech.com.

To register the COM server if the VNA software installation has already completed, open up a command prompt and execute the following commands:

```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\CMT>cd C:\UNA\S2UNA
C:\UNA\S2UNA>S2UNA.exe -Regserver
C:\UNA\S2UNA>
```

COM server registration

COM server registered successfully

OK

(Note: this path is for the S2VNA. Check the programming manual for specific file paths of other instruments.)

This command will register or re-register the COM server and the confirmation dialogue box will appear. If an error occurs, be sure you have administrator privileges. For assistance, please contact support@coppermountaintech.com.

In order to communicate with the VNA through its COM interface, pywin32 needs to be installed. You can find download it here: <http://sourceforge.net/projects/pywin32/files/> (you will need to search the files for the pywin32 matching your version of Python).

After successfully installing pywin32, you can start programming in Python to automate your VNA. Please refer to the Python programming example vna.py for examples of correct syntax. An excerpt of vna.py is shown below. For additional assistance, please contact support@coppermountaintech.com and we will be glad to help!



COPPER MOUNTAIN
TECHNOLOGIES

Python Setup Guide

Jan. 2016

```
# [This is a simple example of setting an ActiveX property in Python. Note
# that when indexed parameters are referenced, the Get prefix and SCPI
# capitalization must be used (e.g. GetSENSe(1) rather than simply sense(1) ]
app.scpI.GetSENSe(1).frequency.center = f1_hz
app.scpI.GetSENSe(1).frequency.span = f2_hz
= else:
    app.scpI.GetSENSe(1).frequency.start = f1_hz
    app.scpI.GetSENSe(1).frequency.stop = f2_hz

app.scpI.GetSENSe(1).sweep.points = num_points

= if instrument[0] != "R":
    app.scpI.GetSOURce(1).power.level.immediate.amplitude = power_level_dbm

#Configure the measurement
app.scpI.GetCALCulate(1).GetPARAmeter(1).define = parameter
app.scpI.GetCALCulate(1).GetPARAmeter(1).select()
app.scpI.GetCALCulate(1).selected.format = format
app.scpI.trigger.sequence.source = "bus"
```