

Programming PPS Power Sources using Python and pyVISA

Applicable to ADF, AFX, AZX, LMX and LSX Series



1 Abstract

This application note explains how to install and use the Python scripting language and available PyVISA instrument control library. Python is a scripting language available on Windows, Linux and Mac OS. It is also an interpreted and high-level programming language for the purpose of general programming requirements. It was designed and developed by the Software Developer named Guido van Rossum. It was first released in the year 1991. More information can be found at <https://www.python.org/>

PyVISA is a Python package that enables you to control all kinds of measurement devices independently of the interface (e.g. GPIB, RS232, USB, Ethernet). As an example, reading self-identification from a Pacific Power AC power source with GPIB number 1 is as easy as three lines of Python code:

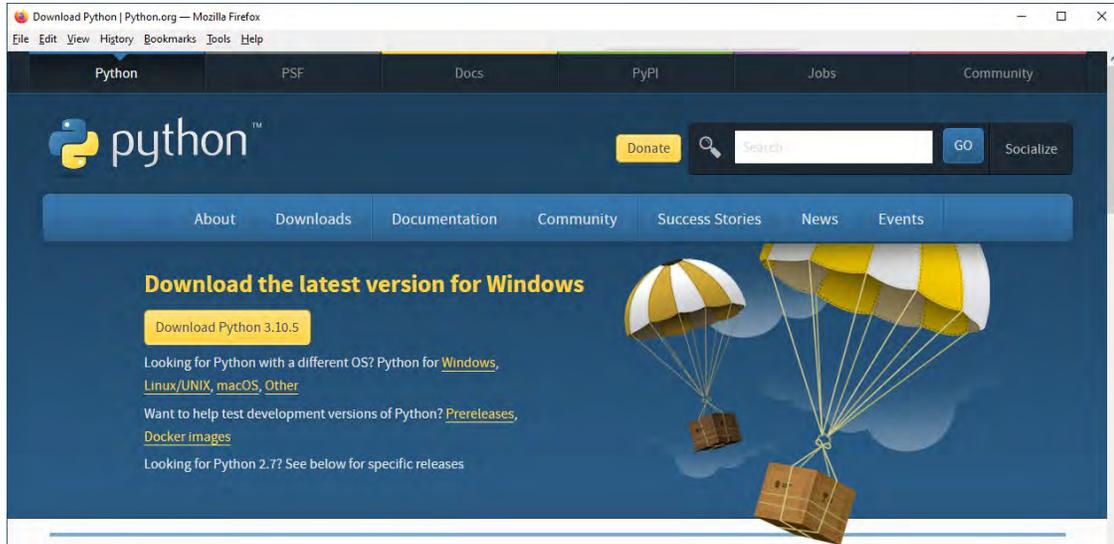
```
>>> import pyvisa
>>> rm = pyvisa.ResourceManager()
>>> rm.list_resources()
('GPIB0::1::INSTR')
>>> inst = rm.open_resource('GPIB0::1::INSTR')
>>> print(inst.query("*IDN?"))
```

More information on PyVISA can be found at <https://pyvisa.readthedocs.io/en/latest/>



2 Installing Python for Windows

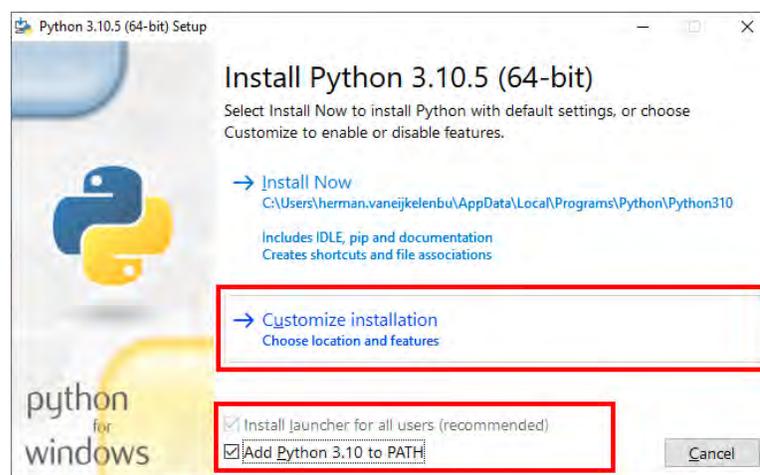
Python can be downloaded from the python.org site <https://www.python.org/downloads/> for a variety of operating systems. For this application note, we will use the Windows version.



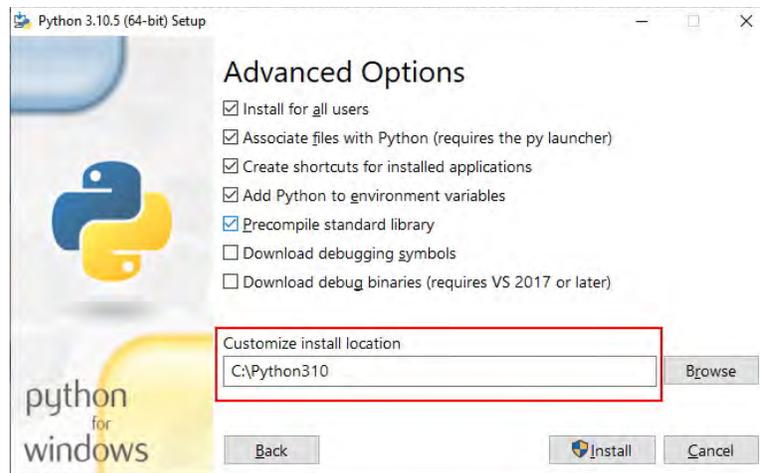
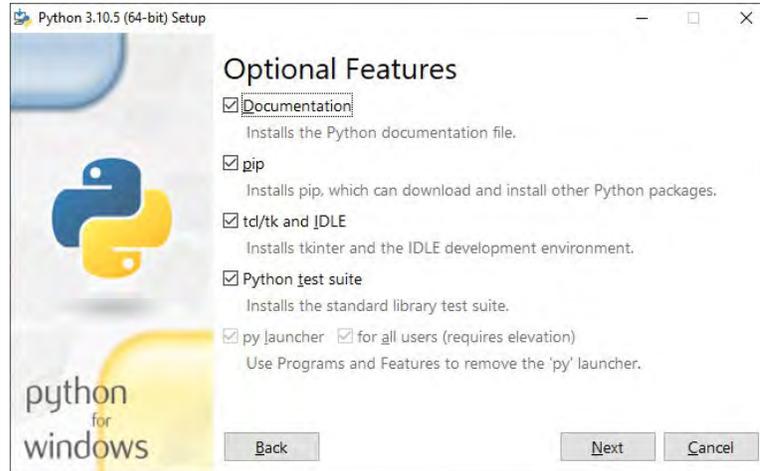
Download the installer from the Python site and run the “python-3.10.5-amd64.exe” file to install. A newer version than the 3.10.5 shown here may be available.

You will be prompted to run a standard or custom installation. We will use a custom installation so we can place the program at the root of the C:\ drive which will make it easier to use the Command Line prompt later.

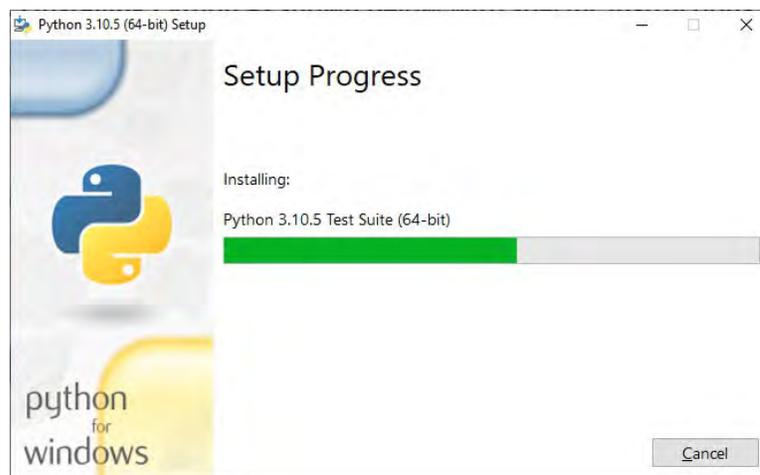
Follow the installation prompt screens shown below to install the Python scripting program. Note the items highlighted in the red box for selections and the installation path.

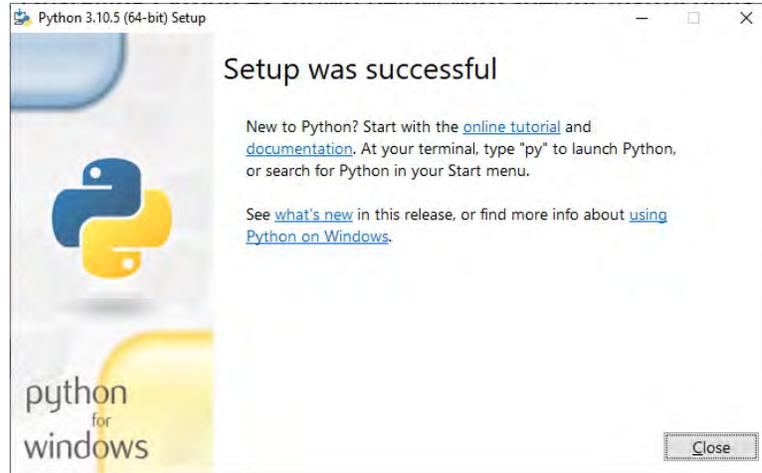


*Note: Make sure the check the “Add Python 3.10 to PATH setting
Select “Customize installation” so the program does **not** install under C:\Program Files*



Change install location to C:\Python310





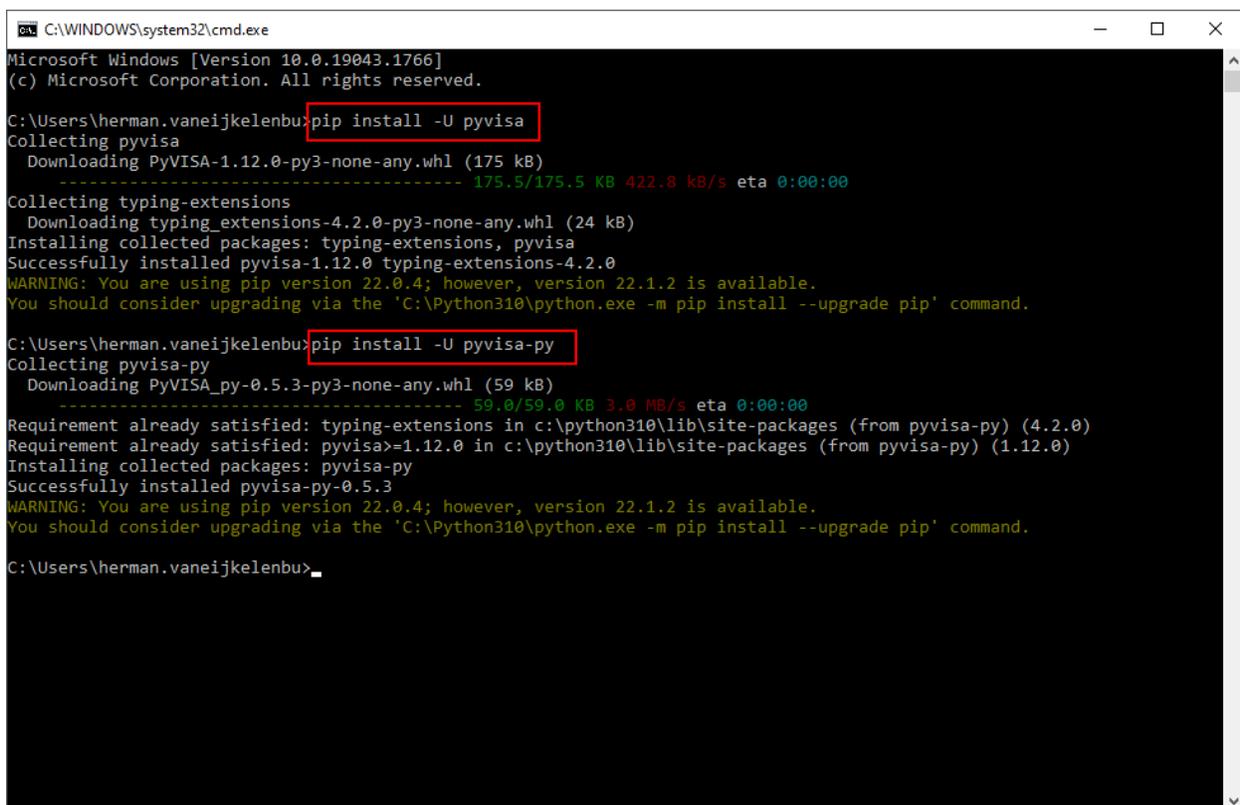
3 Installing PyVISA

In order for PyVISA to work, you need to have a suitable backend. PyVISA includes a backend that wraps the National Instruments' VISA library. However, you need to download and install the library yourself (See NI-VISA Installation). There are multiple VISA implementations from different vendors. PyVISA is tested against National instruments' VISA and Keysight IO Library Suite which can both be downloaded for free (you do not need a development environment only the driver library).

Note that both 32 and 64 bit versions are available for both VISA and PyVISA. You need to make sure that the Python and VISA library have the same bitness, i.e. both 32 bits or both 64 bits.

To install, use the pip installer from the command prompt (cmd.exe).

- On the command prompt, type "pip install -U pyvisa" to install the frontend of pyvisa.
- On the command prompt, type "pip install -U pyvisa-py" to install the backend of pyvisa



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\herman.vaneijkelenbu> pip install -U pyvisa
Collecting pyvisa
  Downloading PyVISA-1.12.0-py3-none-any.whl (175 kB)
----- 175.5/175.5 KB 422.8 kB/s eta 0:00:00
Collecting typing-extensions
  Downloading typing_extensions-4.2.0-py3-none-any.whl (24 kB)
Installing collected packages: typing-extensions, pyvisa
Successfully installed pyvisa-1.12.0 typing-extensions-4.2.0
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'C:\Python310\python.exe -m pip install --upgrade pip' command.

C:\Users\herman.vaneijkelenbu> pip install -U pyvisa-py
Collecting pyvisa-py
  Downloading PyVISA_py-0.5.3-py3-none-any.whl (59 kB)
----- 59.0/59.0 KB 3.0 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions in c:\python310\lib\site-packages (from pyvisa-py) (4.2.0)
Requirement already satisfied: pyvisa>=1.12.0 in c:\python310\lib\site-packages (from pyvisa-py) (1.12.0)
Installing collected packages: pyvisa-py
Successfully installed pyvisa-py-0.5.3
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'C:\Python310\python.exe -m pip install --upgrade pip' command.

C:\Users\herman.vaneijkelenbu>
```

4 Sample Power Source Control Script

The following sample script will display a simple dialog box with an Output On and Off control button.

```

from tkinter import *
from tkinter.font import BOLD
import pyvisa

rm = pyvisa.ResourceManager()

root = Tk()
root.title("AFX Python Example")

def OutputOnOff():
    rscName = ipTxtBox.get()
    afx = rm.open_resource(rscName)
    afx.write('OUTP?')
    state = afx.read()

    if (int(state) == 0):
        afx.write('OUTP 1')
        outLabel.config(text = "Output On", fg='Red')
    else:
        afx.write('OUTP 0')
        outLabel.config(text = "Output Off", fg='Black')
    afx.close()

def ConnectSrc():
    rscName = ipTxtBox.get()
    print(rscName)

    print ("\n")
    print (rm.list_resources())
    print ("\n")

    afx = rm.open_resource(rscName)
    idn = ""
    afx.write('*IDN?')
    idn = afx.read()
    print(idn)

    idnTxtBox.delete(0, END)
    idnTxtBox.insert(0, idn)

    afx.write('OUTP 0')
    outLabel.config(text = "Output On", fg='Black')
    afx.write('FORM 3')
    afx.write('FREQ 50')
    afx.write('VOLT 150')
    afx.write('OUTP 1')
    outLabel.config(text = "Output On", fg='Red')

    print(afx.query('MEAS:FREQ?'))
    print(afx.query('MEAS:VOLT1?'))
    print(afx.query('MEAS:VOLT2?'))
    print(afx.query('MEAS:VOLT3?'))
    afx.close()

#===== Define objects
rowLabel1 = Label(root, text=" ")
colLabel = Label(root, text=" ")
rscLabel = Label(root, justify=LEFT, text="Enter Resource Name : ")
ipTxtBox = Entry(root, width=35, borderwidth=5)
ipTxtBox.insert(28, "TCPIP0::192.168.14.31::inst0::INSTR")
#ipTxtBox.insert(28, "GPIB0::1::INSTR")
idnTxtBox = Entry(root, width=50, borderwidth=5, bg = 'PeachPuff')
btnConnect = Button(root, text="Connect ...", width=10, borderwidth=5, command=ConnectSrc)
rowLabel2 = Label(root, text=" ")
  
```

```

outLabel = Label(root, text="Output Off", width=15, font=BOLD)
btnOnOff = Button(root, text="Output ...", width=10, borderwidth=5, command=OutputOnOff)
rowLabel3 = Label(root, text=" ")
#####

##### Define location
rowLabel1.grid(row=0, column=0)
colLabel.grid(row=0, column=3)
rscLabel.grid(row=1, column=0)
ipTxtBox.grid(row=2, column=0, padx=10, pady=10)
btnConnect.grid(row=2, column=1)
idnTxtBox.grid(row=3, column=0, padx=10, pady=10, columnspan=3)
rowLabel2.grid(row=4, column=0)
outLabel.grid(row=5, column=0)
btnOnOff.grid(row=5, column=1)
rowLabel3.grid(row=6, column=0)
#####

root.mainloop()

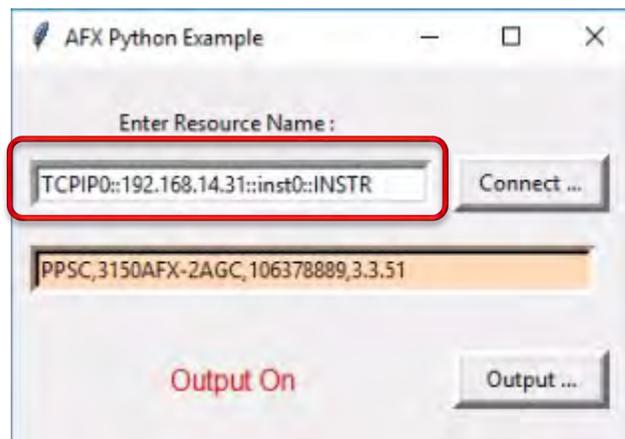
```

5 Running the Sample Script

For this application note, the sample Python script file is placed in a subdirectory called “examples” under the Python310 program directory. Copy the example program in the following folder: C:\Python310\examples\.

On the command prompt, type “python C:\python310\examples\afx_python_example.py”

Type the resource name on the resource name textbox then click the “**Connect ...**” button to connect to the power source. Click the “**Output ...**” button to enable and disable the power source output state as shown below.

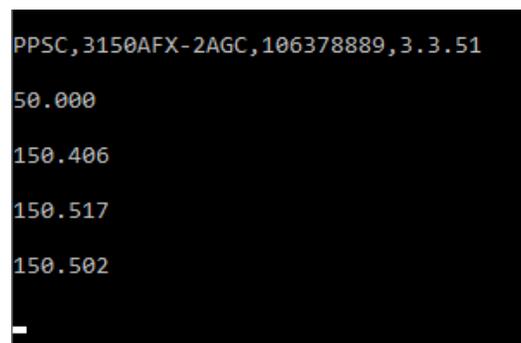


The followings are the possible resource names:

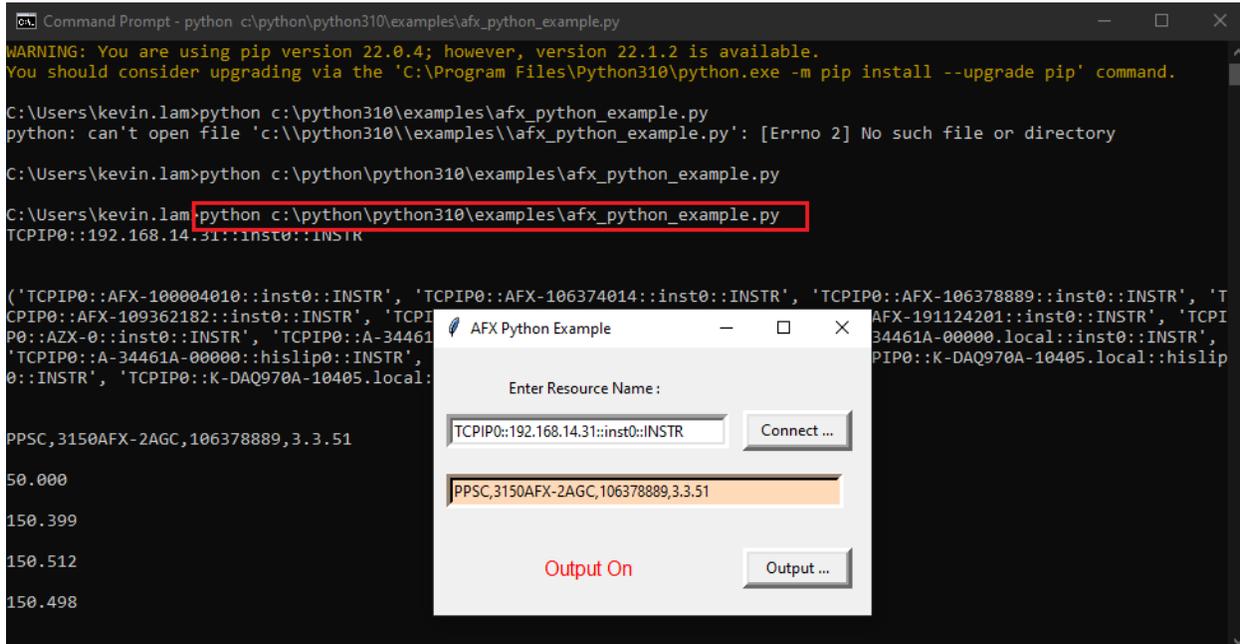
TCPIP0::192.168.14.31::inst0::INSTR	(LAN)
GPIB0::1::INSTR	(GPIB)
ARSL::1::INSTR	(SERIAL)

Once connection is established, the script will query the power source id using the *IDN? Command and display this information in the dialog screen below the VISA resource. Also, the output state of the power source will be displayed at the bottom as either **Output On** or **Output Off**. To change output state, click the “**Output...**” button to the right.

If the output state is ON, the scrip will query the frequency and vltage for each output phase and display this information in the command window.



The cmd.exe windows and dialog are displayed in full below.



```
Command Prompt - python c:\python\python310\examples\afx_python_example.py
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'C:\Program Files\Python310\python.exe -m pip install --upgrade pip' command.

C:\Users\kevin.lam>python c:\python310\examples\afx_python_example.py
python: can't open file 'c:\\python310\\examples\\afx_python_example.py': [Errno 2] No such file or directory

C:\Users\kevin.lam>python c:\python\python310\examples\afx_python_example.py
C:\Users\kevin.lam>python c:\python\python310\examples\afx_python_example.py
TCPiP0:192.168.14.31::inst0::INSTR

('TCPiP0:AFX-100004010::inst0::INSTR', 'TCPiP0:AFX-106374014::inst0::INSTR', 'TCPiP0:AFX-106378889::inst0::INSTR', 'T
CPIP0:AFX-109362182::inst0::INSTR', 'TCPI
P0:AZX-0::inst0::INSTR', 'TCPiP0:A-34461
P0:A-34461A-00000::hislip0::INSTR',
'TCPiP0:A-34461A-00000::hislip0::INSTR',
0::INSTR', 'TCPiP0:K-DAQ970A-10405.local:
PPSC,3150AFX-2AGC,106378889,3.3.51
50.000
150.399
150.512
150.498
```

6 Customer Support

For application support, contact Pacific Power Source's Customers Service - Toll Free US: +1 (800) 854-2433 / support@pacificpower.com or your local authorized Pacific Power Source distributor.



2802 Kelvin Ave, Suite 100
Irvine, CA 92614 USA
Phone: +1 949.251.1800
Fax: +1 949.756.0756
Toll Free: 800.854.2433
E-mail: sales@pacificpower.com
www.pacificpower.com