Stand Alone use of check_esxi_hardware.py on Windows

The following procedure allows the use of Claudio Kuenzler's popular Nagios ESXi monitoring Python script (https://www.claudiokuenzler.com/nagios-plugins/check_esxi_hardware.php) in the Windows environment. This makes monitoring ESX servers from Windows trivial, assuming you can capture and process the resulting data, e.g. by piping the output to a text file.

The following has been tested on Windows server 2008R2 & 2012, Windows 7 and 10. It should be platform independent; no specific Windows components or prerequisites are required.

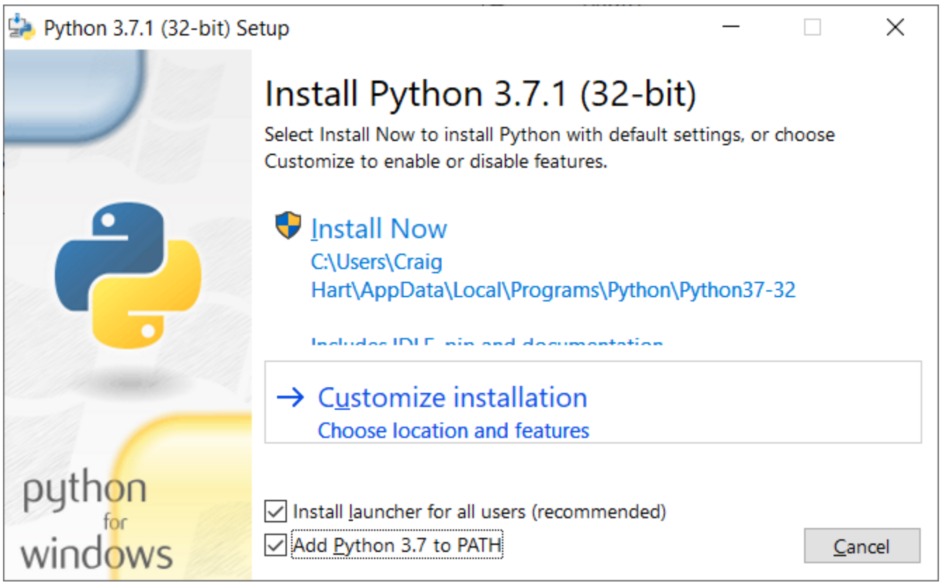
Requirements

Download Python 3.xx (Grab the current version – 3.7.1 at the time of writing) for Windows from https://www.python.org/downloads/windows/

Download the ZIP file of the PyWBEM client from https://github.com/pywbem/pywbem

Step 1: Install Python

Run the Python installer and choose **'Add Python 3.7 to PATH'** + choose the **Customize installation** option



Install Python 3.7.1 (32-bit)

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

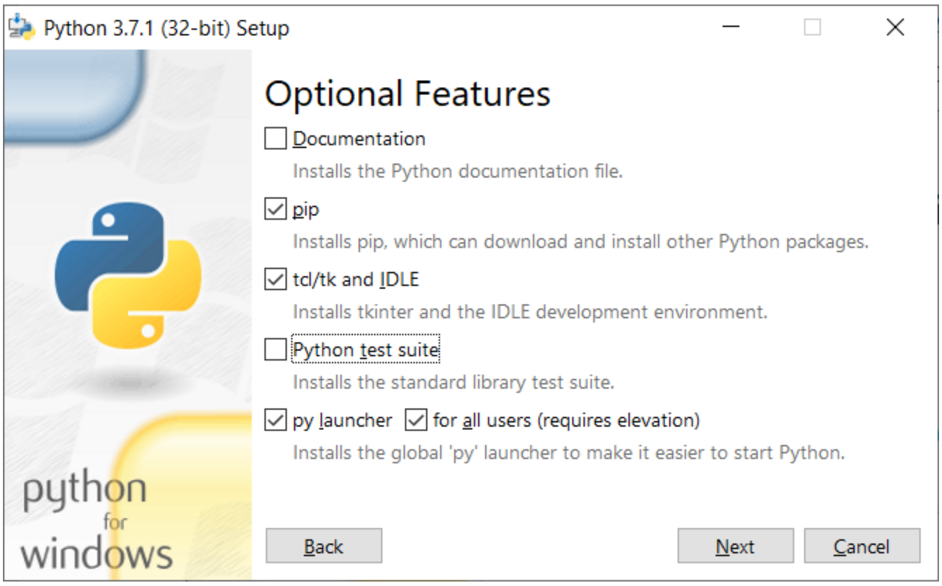
Install Now
C:\Users\Craig Hart\AppData\Local\Programs\Python\Python37-32

Includes IDLE, pip and documentation

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

You can omit the documentation and test suite – they are not required, but cause no harm if left in. PIP and the py launcher ARE required:



Optional Features

☐ Documentation
Installs the Python documentation file.

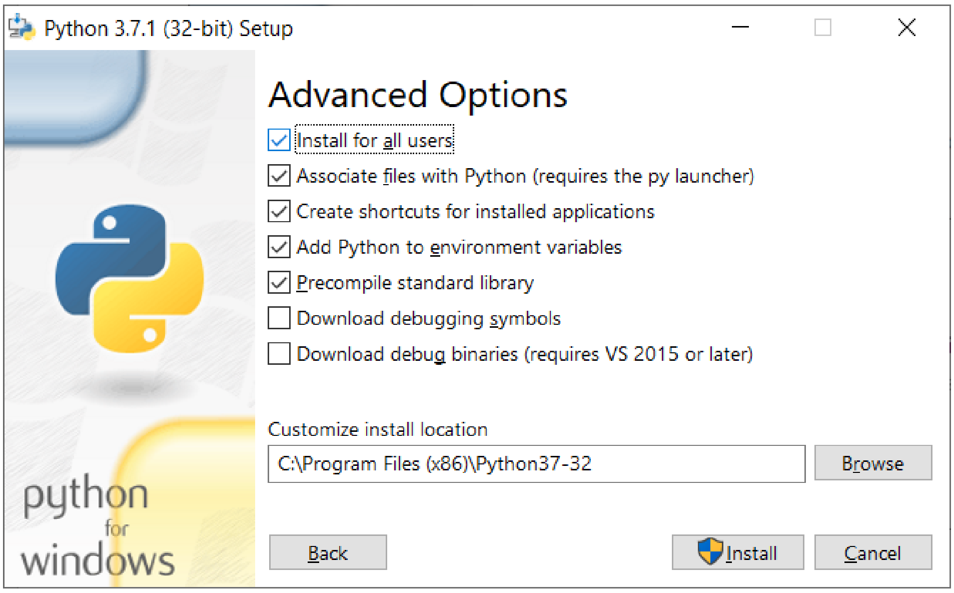
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☐ Python test suite
Installs the standard library test suite.

☑ py launcher ☑ for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.

It is recommended to 'install for all users' so that Python can run even if launched as a scheduled task or by a different user. This causes python to be installed in program files, rather than the current user's personal profile.



Python 3.7.1 (32-bit) Setup

Advanced Options

- ☑ Install for all users
- ☑ Associate files with Python (requires the py launcher)
- ☑ Create shortcuts for installed applications
- ☑ Add Python to environment variables
- ☑ Precompile standard library
- □ Download debugging symbols
- □ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Program Files (x86)\Python37-32

Click Install and allow the install to complete.

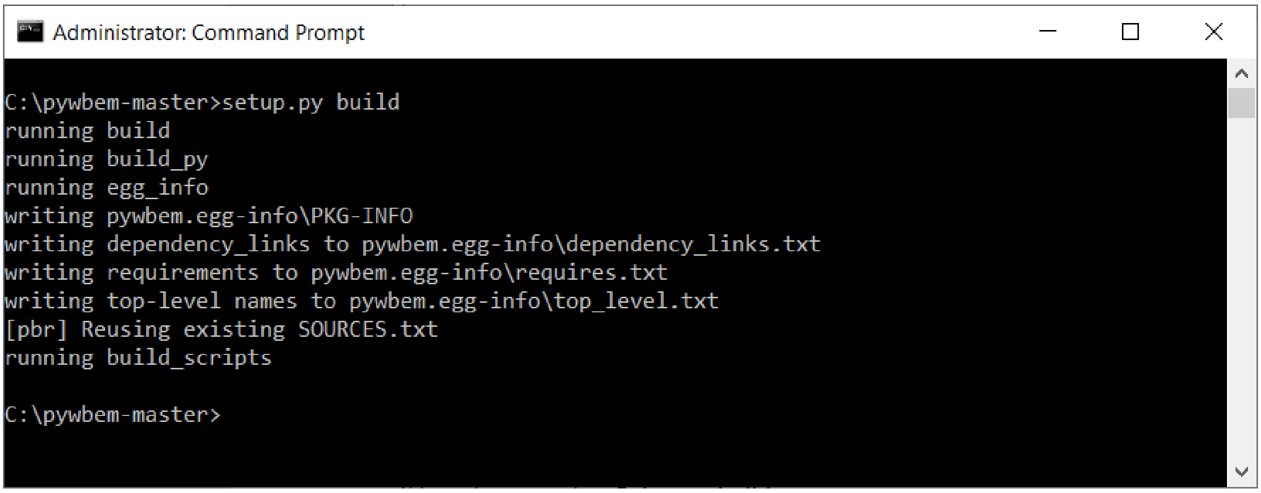
Step 2: Compile and Install PyWBEM into Python

Unzip the PyWBEM download into any folder. We will use **c:\pywbem-master**

Open a command prompt and move into that folder with **cd c:\pywbem-master**

Compile the PyWBEM code with the following command:

C:\pywbem-master>**Setup.py build**

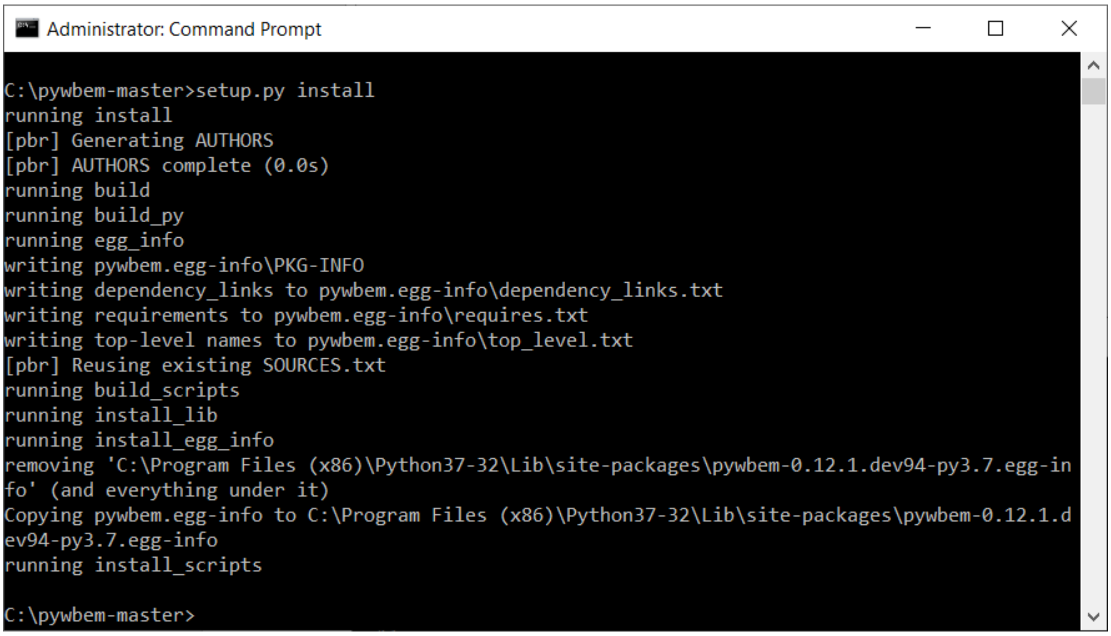


Administrator: Command Prompt

```
C:\pywbem-master>setup.py build
running build
running build_py
running egg_info
writing pywbem.egg-info\PKG-INFO
writing dependency_links to pywbem.egg-info\dependency_links.txt
writing requirements to pywbem.egg-info\requires.txt
writing top-level names to pywbem.egg-info\top_level.txt
[pbr] Reusing existing SOURCES.txt
running build_scripts

C:\pywbem-master>
```

Install PyWBEM into Python with the following command:

C:\pywbem-master>**setup.py install**



```
C:\pywbem-master>setup.py install
running install
[pbr] Generating AUTHORS
[pbr] AUTHORS complete (0.0s)
running build
running build_py
running egg_info
writing pywbem.egg-info\PKG-INFO
writing dependency_links to pywbem.egg-info\dependency_links.txt
writing requirements to pywbem.egg-info\requires.txt
writing top-level names to pywbem.egg-info\top_level.txt
[pbr] Reusing existing SOURCES.txt
running build_scripts
running install_lib
running install_egg_info
removing 'C:\Program Files (x86)\Python37-32\Lib\site-packages\pywbem-0.12.1.dev94-py3.7.egg-in
fo' (and everything under it)
Copying pywbem.egg-info to C:\Program Files (x86)\Python37-32\Lib\site-packages\pywbem-0.12.1.d
ev94-py3.7.egg-info
running install_scripts

C:\pywbem-master>
```

You will see various messages from each command, this is normal as long as no errors are shown.

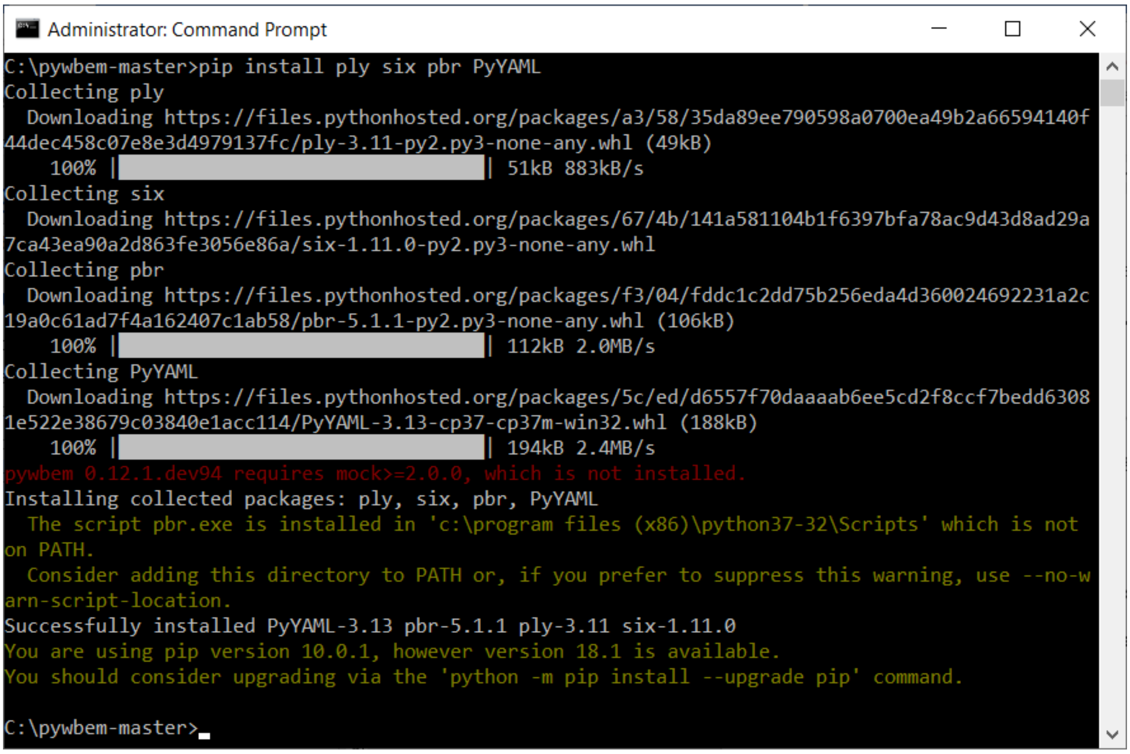
PyWBEM is now installed.

Step 3: Add required Libraries to Python

The script requires a few extra libraries to be added to Python. Install them using PIP as follows:

C:\pywbem-master>**pip install ply six pbr PyYAML**

You will see various messages from each command, this is normal as long as no errors are shown (The brown messages are *not* errors). Regarding the red error message - I did not find mock as being required, however you can also issue a **pip install mock** command if you wish.



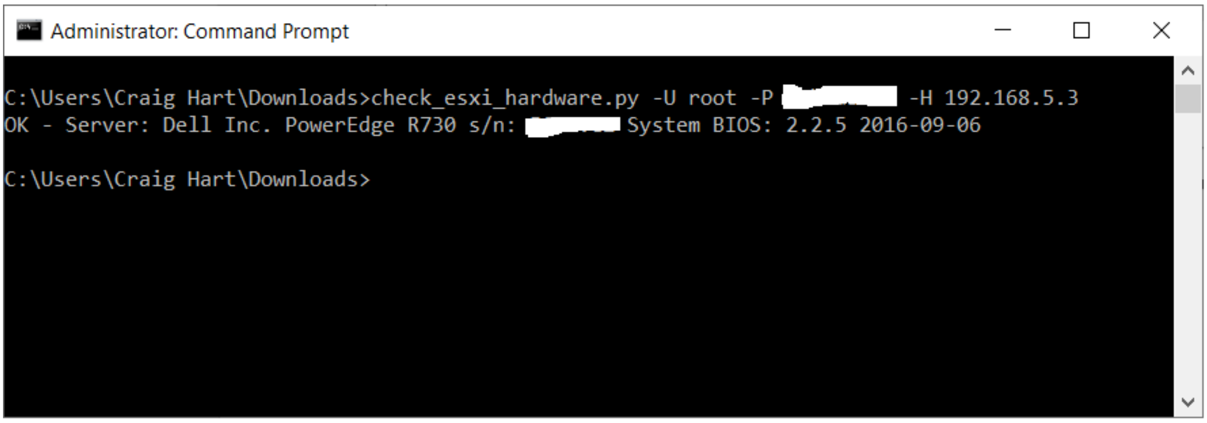
```
C:\pywbem-master>pip install ply six pbr PyYAML
Collecting ply
  Downloading https://files.pythonhosted.org/packages/a3/58/35da89ee790598a0700ea49b2a66594140f
44dec458c07e8e3d4979137fc/ply-3.11-py2.py3-none-any.whl (49kB)
    100% |████████████████████████████████| 51kB 883kB/s
Collecting six
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a
7ca43ea90a2d863fe3056e86a/six-1.11.0-py2.py3-none-any.whl
Collecting pbr
  Downloading https://files.pythonhosted.org/packages/f3/04/fddc1c2dd75b256eda4d360024692231a2c
19a0c61ad7f4a162407c1ab58/pbr-5.1.1-py2.py3-none-any.whl (106kB)
    100% |████████████████████████████████| 112kB 2.0MB/s
Collecting PyYAML
  Downloading https://files.pythonhosted.org/packages/5c/ed/d6557f70daaaab6ee5cd2f8ccf7bedd6308
1e522e38679c03840e1acc114/PyYAML-3.13-cp37-cp37m-win32.whl (188kB)
    100% |████████████████████████████████| 194kB 2.4MB/s
pywbem 0.12.1.dev94 requires mock>=2.0.0, which is not installed.
Installing collected packages: ply, six, pbr, PyYAML
  The script pbr.exe is installed in 'c:\program files (x86)\python37-32\Scripts' which is not
on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-w
arn-script-location.
Successfully installed PyYAML-3.13 pbr-5.1.1 ply-3.11 six-1.11.0
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\pywbem-master>_
```

Step 4: Run check_esxi_hardware.py

Download the script from https://www.claudiokuenzler.com/nagios-plugins/check_esxi_hardware.php#Download – you'll need to right-click on the link found on this webpage and choose 'save as…' or 'save as file…', otherwise the web browser will just open the file onscreen.

Once saved to your machine, simply run the script as per the instructions:

C:\>**check_esxi_hardware.py –U <user> -P <password> -H <server IP address>**



```
C:\Users\Craig Hart\Downloads>check_esxi_hardware.py -U root -P [redacted] -H 192.168.5.3
OK - Server: Dell Inc. PowerEdge R730 s/n: [redacted] System BIOS: 2.2.5 2016-09-06

C:\Users\Craig Hart\Downloads>
```

Output can be piped to a text file e.g.:

C:\>**check_esxi_hardware.py –U <user> -P <password> -H <server IP address> > ESXiHealth.txt**

The resulting ESXIHealth.txt can then be processed to recover the required information in the monitoring platform of your choice.

That's it!!

I place this document in the public domain, for the use of all check_esxi_hardware.py users without restriction.

Written by Craig Hart, Revision 1.0, 30 November 2018. craighart71 at gmail.com