

Installation and Setup of ZeroMQ, MetaTrader 4, Python and IDE on Windows 10

Ruslan Masinjila

October 23, 2018

Requirements

- ▶ Windows 10
- ▶ Python 2.7
- ▶ pyzmq (Python binding for ZeroMQ)
- ▶ Python IDE
- ▶ MetaTrader 4

Installation of Python 2.7

Easiest way to install and configure Python is through Anaconda.

- ▶ Visit Anaconda (<https://www.anaconda.com/download/>)
- ▶ Download Anaconda installer for 32-bit Python 2.7 ¹ and place it on Desktop
- ▶ Create a folder inside the **C** directory called *Anaconda*. This is where installed files will be placed.
- ▶ Run the downloaded installation file and follow instructions. For the destination folder, select the folder you just created using the browse button. Once selected, the destination path should look like **C:\Anaconda**. Click next and proceed with the installation.
- ▶ At the end of installation, click on *Install Microsoft VSCode* (recommended).

¹32-bit is required for compatibility with pyzmq

Configuration of Python 2.7

After installation of Python 2.7 through Anaconda, open **cmd** prompt (Search for **cmd** from bottom left, near **windows** icon). Within the terminal, type *python* followed by **Enter**.

Most likely, *"python" is not recognized...* will be displayed. In order to fix this, a **path** to python has to be set as follows:

- ▶ Click on search icon near bottom left
- ▶ Type **Environment** in the search filed. Then select **Edit the System Environment Variable**
- ▶ In the window that pops up, click on **Environment Variables...**
- ▶ Select **Path** under **System Variables** and click on **Edit...**
- ▶ In the next window that pops up, click on **New** and paste the address of Anaconda, i.e. **C:\Anaconda**.
- ▶ Click **OK** in all open windows. Open a new **cmd** prompt and type *python*. This time it should display python's version and console.

Installation of pyzmq

Pyzmq (<https://pyzmq.readthedocs.io/en/latest/>) provides python bindings for ZeroMQ (<http://zeromq.org/>)
To install pyzmq, first add the path of `pip` to system variables as explained earlier. `Pip` is inside the `Scripts` folder of the `Anaconda` folder, i.e. `C:\Anaconda\Scripts`. Thereafter:

- ▶ Open `cmd` terminal and type `pip`. If environment is configured properly, the terminal should display list of available commands for pip.
- ▶ Type `python -m pip install --upgrade pip` in the terminal and press Enter to upgrade pip.
- ▶ Type `pip install pyzmq` in the terminal and press Enter to install pyzmq.
- ▶ Type `python` in the terminal and press Enter to open python console. Then type `import zmq` and press Enter. Type `print zmq.pyzmq_version()` and press Enter. If no errors occur, the version of pyzmq will be displayed on the terminal.

Installation of Python IDE

Any Python IDE can be used. The following are instructions to install and configure PyDev in Eclipse IDE.

- ▶ If Java Runtime Environment (JRE) is not installed on the machine, download JRE from (<https://www.java.com/en/download/>)
- ▶ Install the JRE using default settings.
- ▶ Download Eclipse IDE from (<https://www.eclipse.org/downloads/>)
- ▶ Install **Eclipse IDE for Java Developers** with the default settings
- ▶ Once Eclipse IDE is installed, launch the IDE using default settings.
- ▶ Install **PyDev** for Eclipse IDE using instructions from here (<https://bit.ly/2yRroHU>). When prompted, choose **Install Anyway**, then **Restart Now** Eclipse IDE.

Testing IDE with Python and ZeroMQ [1/2]

The following are steps to test PyDev and ZeroMQ in Eclipse IDE

- ▶ On Eclipse IDE, click **File**→**New**→**Project**→**PyDev**→**PyDev Project**
- ▶ For the **Project name** write **ZEROMQ**
- ▶ Click on **Please configure an interpreter before proceeding** if the python interpreter is not configured in eclipse.
- ▶ Click on **Quick Auto-Config** in the pop-up window and click **Next** for configuration to complete.
- ▶ Click **Finish** followed by **Open perspective**
- ▶ Python Project is now created. Use **zeromq_server.py** available at <https://bit.ly/2Ja0Ztv> and **zeromq_client.py** available at <https://bit.ly/2Cwf08t> to test "Hello Wolrd" via pyzmq. First run the server, then run the client.

Testing IDE with Python and ZeroMQ [2/2]

This is the most important step for the integration of ZEROMQ/pyzmq with PyDev

- ▶ In the previous slide, you are likely to get an error (**Unidentified variable from import...**) after copy-pasting the zeromq_server.py and zeromq_client.py to the project (**ZEROMQ**)
- ▶ To fix this error, Go to **Windows**→**Preferences**→**PyDev**→**Interpreters**→**Python Interpreter**
- ▶ Click on **Forced Builtins**
- ▶ Click on **New**
- ▶ Type **zmq**
- ▶ Click on **Apply**
- ▶ Click on **Apply and Close**
- ▶ **Restart Eclipse**. All Errors should be gone. Run **zeromq_server.py** then **zeromq_client.py**. The server terminal should print **Hello** and the client terminal should print **World**.

Installation of MetaTrader 4

Log into your favourite broker, in this case darwinex (<https://www.darwinex.com/>) and download MetaTrader 4 (MT 4), which is 32-bit.

- ▶ Install MT4 and Log in with your Credentials.
- ▶ **IMPORTANT:** To enable execution of DLLs, Click on **Tools**→**Options**. Then click on **Expert Advisors** and check **Allow DLL imports (potentially dangerous, enable only for trusted applications)**. Then click **OK**.

Adding mql-zmq bindings to MetaTrader 4 [1/2]

Log into your favourite broker, in this case darwinex (<https://www.darwinex.com/>) and download MetaTrader 4 (MT 4), which is 32-bit.

- ▶ Before proceeding, make sure that hidden folders in Windows are visible. To make them visible, type **folders** in the bottom left **Search** field next to the **Windows** icon. Then click on **File Explorer Options** followed by **View** in the pop-up window. Check **Show hidden files, folders, and drives**. Then click **OK**.
- ▶ Installed MetaTrader 4 files can then be found in C:\Users\user_name\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13

Adding mql-zmq bindings to MetaTrader 4 [2/2]

- ▶ Now copy or download mql-zmq bindings for MQL language from <https://github.com/dingmaotu/mql-zmq> and extract the files.
- ▶ Open the extracted folder, and copy the contents of the **Include** folder into
C:\Users\juser_namej\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13\MQL4\Include
- ▶ Copy the contents of the **Scripts** folder into
C:\Users\user_name\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13\MQL4 \Scripts
- ▶ Copy the contents of **Library\MT4** into
C:\Users\user_name\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13\MQL4\Libraries

Testing ZEROMQ with MetaTrader 4 and Python

- ▶ Clone or Download tools and templates from DarwinexLabs (<https://github.com/darwinex/DarwinexLabs>) then extract the folder.
- ▶ Copy the contents of `tools\MQL4` into `C:\Users\user_name\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13\MQL4\Experts`

Testing Entire Setup

- ▶ Copy the contents of `tools\MQL4` into `C:\Users\user_name\AppData\Roaming\MetaQuotes\Terminal\3B534B10135CFEDF8CD1AAB8BD994B13\MQL4\Experts`.
- ▶ Now open Eclipse IDE and create a python file called `zeromq_test.py` within ZEROMQ project created earlier.
- ▶ Copy the contents of the file in `tools\Python\ZeroMQ_MT4_Python_Template.py` into `zeromq_test.py` file you just created.
- ▶ Within MT4, drag-and-drop `ZeroMQ_MT4_Python_Template` into an open chart.
- ▶ Within Eclipse, run `zeromq_test.py`.
- ▶ If everything goes well, then within terminals of both Eclipse and MT4 you should see outputs about EUR/USD.