**Windows Build Instructions**

Warning: Building Armory on Windows is fairly involved!

Make sure you have Microsoft Visual Studio 2012 (other versions may take some tinkering). Regardless of whether you are on a 64 bit system, use the x32 compilers in MSVS and 32-bit versions of everything you download. Also, make sure that MSVS builds in Release mode.

* **Install Git and TortoiseGit**

» <https://code.google.com/p/msysgit/downloads/list>

» <http://code.google.com/p/tortoisegit/downloads/list>

» Select all default options if you don’t know what they are.

* **Clone the repo:**

» Right click on the directory where you want the code to go: select “Git clone…”

» URL: git://github.com/etotheipi/BitcoinArmory.git

» Click OK

* **Download and install Python (2.7.5):**

» <http://www.python.org/getit/>

* **Download and install Python-Twisted (13.1.0.win32-py2.7):**

» <http://twistedmatrix.com/trac/wiki/Downloads>

* **Download Python-Psutil (1.0.1.win32-py2.7)**

» <https://code.google.com/p/psutil/downloads/list>

* **Download and install zope.interface (for python-twisted) (3.8.0.win32-py2.7):**

» <http://pypi.python.org/pypi/zope.interface/3.8.0#downloads>

After you install Zope, go to C:\Python27\Lib\site-packages\zope, and create a new empty file called “\_\_init\_\_.py”.

* **Download and install pywin32 (218.win32-py2.7):**

» <http://sourceforge.net/projects/pywin32/files/>

* **Download and full-install PyQt4 (4.10.3-gpl-Py2.7-Qt4.8.5-x32):**

» <http://www.riverbankcomputing.co.uk/software/pyqt/download>

(If you only want to run Armory through python or use armoryengine.py, then you can stop here. The remaining steps help you build \_CppBlockUtils.pyd and CppBlockUtils.py, but if you have installed Armory, you can simply grab them from the installation directory and put them into the base BitcoinArmory directory)

* **Download and UNPACK SWIG (swigwin-2.0.4, it’s the second link on the page).**

» <http://www.swig.org/download.html>

» Move the inner-most swigwin-2.0.4 directory to BtcoinArmory/cppForSwig directory, rename it to just “swigwin”

* **Add C:\Python27 to your PATH environment variable:**

» python.exe needs to be in your PATH for both compilation, and running Armory. You can add it by going to Start->Control Panel->System->Advanced(Tab)->Environment Variables(Button). In Window 7, you can simply use the search box in the upper right in the control panel to search for “Environment”

» In the bottom list box, scroll to “Path”, click “Edit”, and add “;C:\Python27\” to the end.

» REBOOT. There is no other way to enable the environment variable.

**Finally!** All steps above are complete. Go into the cppForSwig directory and open the ArmoryEngine.sln file. Make sure to select “Release” mode and “x64″ (or Win32 if 32-bit OS). Rebuild solution. This will compile all the C++ into a swig module, and copy/rename it to the correct location (make sure swigwin is in the right place!). Once this is done, you only need to go back to the root directory and double-click ArmoryQt.py.

Final Note: The MSVS project actually tries to execute one more step after compiling, which was not covered here — for producing a standalone .exe. For just using Armory, this part is unnecessary. If you see the line “***1 file(s) copied.”*** you actually *succeeded.*Ignore the error and double-click ArmoryQt.py!

To enable the post build steps to create the installer .exe file add these downloads:

* **Download and install Resource Tuner Console (1.93):**

» <http://www.heaventools.com/download-rtc.htm>

* **Nullsoft Scriptable Install System (3.0a1):**

» <http://nsis.sourceforge.net/Download>

**Add C:\Resource Tuner Console\;C:\Program Files (x86)\NSIS to your PATH environment variable:**

» Follow the same steps that you did for adding Python27 to your PATH.