Scrapy installation and setup on Windows 7 (32/64)

Matthew J. Holland*

Abstract

In this document we carry out a detailed, step-by-step discussion of getting the Python-driven Scrapy installed and functioning on machine running Windows 7 (either the 32-bit or 64-bit version), a process which can be quite complex and prone to error.

Introduction

The Scrapy framework for building web-crawling software in Python is extremely easy to use, offers great flexibility in terms of functionality, as well as solid performance [1]. Using Scrapy is not a difficult task, but installing it can be exceedingly challenging, potentially even a barrier to many users. The official documentation [2] does indeed elucidate precisely what is needed for a successful install, but there is a real hierarchy of prerequisites that comes into play, and that can make things quite a bit more complicated.

Initial setup

These first few steps are of course critically important, but they are also comparatively straightforward and easy to carry out.

- Install python 2.7X (latest bugfixed version), using the MSI installer.
- Add C:\python27\Scripts and C:\python27 to the `Path` environment variable, accessible from following `control panel -> system` and so forth.
- Download and install Visual C++ 2008 redistributables from the Win32 OpenSSL page [4], and note that we don't need to worry about any service packs or anything, just the version linked on the site, which for reference is "Microsoft Visual C++ 2008 Redistributable Package" with "x86" in parentheses for the 32-bit version, and "x64" for the 64-bit version. These versions can be found at the MS downloads site at [5] and [6] respectively.
- Download and install OpenSSL for windows from the same site as above [4]. Since we're on 64-bit Windows, we go with the appropriate file, called "Win32/Win64 OpenSSL v1.0.0k" as of March 12, 2013. Note that the standard version is needed, do *not* use the light version. **Added note:** upon installation, we are given the option to "copy dlls to bin directory," and in our successful installation attempts, we have always selected the option to copy to bin. As well, do not forget to add `openssl-win32\bin` or `openssl-win64\bin` folder to the PATH environment variable.

*Affiliation: Mathematical Informatics Lab, Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. Contact: `matthew-h ATMK is.naist.jp`

- In the scrapy documentation it is remarked that apparently some prerequisites for Scrapy will have their installs fail "if you don't have Visual Studio installed," which is assuredly very vague. It turns out that Visual Studio 2008/v9.0 is necessary for working in the Python 2.7X environment. The full version of the software is a commercial product, but there is a free "express" version which is satisfactory. Specifically, we want "Microsoft Visual Studio 2008 Express Edition," though that includes a lot of content we don't need. The specific portion of that software suite we want is "Visual C++ 2008 Express," and the setup file is called `vcsetup.exe`. As of early 2013, a search on the MS downloads site will no longer return any useful results, but the file itself can be found using a deep link at [7].
- We now move to the first true prerequisite for using Scrapy, which is `pywin32`. The latest build can be found on sourceforge, and as of March 2013, "build 218" is the newest build. We want the version for Python 2.7, so select the file with either `win32` or `win-amd64` depending on your build.
- The next prerequisite for Scrapy is `Twisted`. Get the MSI installer on the downloads page for either 32/64 [9], and install as per usual.

Installing pip While many users may already have it, we will find it useful to have `pip` installed on our machine. To do so, first download the file `get-pip.py` from [10] into our python27 folder. To run it straight from the command line, we "need setuptools or distribute." As `distribute` is facing deprecation, we install `setuptools`. The downloads page [11] has all the relevant information. If 32-bit, just use the executable. If 64-bit, download `ez_setup.py` as linked on that page, and run it in the command line using, for example

```
python ez_setup.py
```

at the appropriate location. Note that doing this also installs `easy_install`. It is then simply a matter of running

```
python get-pip.py
```

to finish the installation.

Trickier steps

For all the steps in this section, there are generally numerous steps required which are not touched on in the least by the documentation, and thus make for a fairly significant challenge. Fortunately, we found a successful "installation recipe" which we have replicated on three machines, and are thus confident in the methods discussed below.

Zope.Interface (using .EXE) Said in the PyPI documentation to provide an "implementation of object interfaces for Python," `Zope.Interface` is required for the use of Twisted. In the past, the only format available for 64-bit machines was an .EGG file, but now binaries are available for both Windows builds, thus all we need to do is download and run the .EXE files easily accessed on the project PyPI website [12]. That's all for this step[1].

[1] If one elects to use .EGG files for whatever reason, save it to the python27 folder, and simply run `easy_install file_name.egg` in the command prompt.

lxml While the setup us easy, a potentially confusing step relates to `lxml`, a "Pythonic binding for the C libraries `libxml2` and `libxslt`," and one might naturally assume that those two C libraries were also necessary for this guy to be of any use, but on Windows that is not the case. There is plenty of good info on the project site [13], but for our purposes things can just get increasingly complicated. For developing software using C the two libraries noted above are necessary, but here we are in the clear if we have an lxml binary for Windows. The main project itself does *not* offer a Windows binary any longer, but thankfully a generous individual has created a great deal of Python libraries for Windows, and made them available on his website [14]. Using the most recent version executable for Python 2.7 on Windows 32/64 should work fine.

Two more easy steps The first of the next two easy steps is to install `pyOpenSSL`. Initially finding a 64-bit version can be difficult, but thanks to a handy info source on Github [15], one finds that a German software development group has been kind enough to make develop both 32-bit and 64-bit versions, and has made them freely available online [16]. Scroll down the page, select the MSI installer for the Python 2.7 version according to your system's build, and then instead of clicking the "download" button, click the "URL" link, and then scroll down to complete the download.

For the second easy step, we install `w3lib` using easy_install. Navigate to the `\scripts` folder in our python27 folder via the command line, and simply run

`easy_install w3lib` to complete this step.

Dealing with compiling errors

With that, we might feel inclined to use `pip` to go ahead and install Scrapy into `\scripts`, but doing so will either result in Scrapy's install failing with the "u path" error in `msvc9compiler` or install successfully but be totally non-functional with the rather nebulous "cannot find specified module" error which is easy to get stuck on. Put simply, we are very likely to run into compiling errors of various shapes and sizes.

Fortunately, a concise and well-written blog post [17] not at all about Scrapy provided much of the information we need to finish things up.

SDK install We need to download and install "Microsoft Windows SDK for Windows 7 and .NET Framework 3.5 SP1," whose installer `winsdk_web.exe` is available at Microsoft's downloads page [8] as of March 2013. The download can be large, and be aware that we *must* have "Microsoft Visual C++ Compilers 2008" selected in our installation (though the documentation and such is optional).

Very technical tweaks If using 64-bit Windows, the following file

```
\Microsoft Visual Studio 9.0\VC\bin\vcvars64.bat
```

is important, and it needs to be copied. For 32-bit users, the file won't exist and so copying of course will not be necessary. For 64-bit users, copy it to

```
\Microsoft Visual Studio 9.0\VC\bin\amd64\
```

and save it as `vcvarsamd64.bat` so that we end up with the following copied file:

```
\Microsoft Visual Studio 9.0\VC\bin\amd64\vcvarsamd64.bat
```

After that, for both 32-bit and 64-bit users, we need to edit the files `msvc9compiler.py` and `msvccompiler.py`, which are located in the following folder:

```
C:\Python27\Lib\distutils\
```

First, we find the line in `msvc9compiler.py` with

```
ld_args.append('/MANIFESTFILE:' + temp_manifest)
```

in it, and adding a new line directly below it (at the same indentation) we include

```
ld_args.append('/MANIFEST')
```

We then find the section around line 153 in `msvccompiler.py` which has

```
df get_build_version():
```

and add one line with `return 9.0` such that the area of code looks as follows:

```
  def get_build_version():
      """Return the version of MSVC that was used to build Python.

      For Python 2.3 and up, the version number is included in
      sys.version.  For earlier versions, assume the compiler is MSVC 6.
      """
      return 9.0
      prefix = "MSC v."
      i = string.find(sys.version, prefix)
      if i == -1:
          return 6
```

Finally, if it doesn't already exist (it probably will not for 64-bit users), we create a new system environment variable as follows:

```
  Name: VS90COMNTOOLS
  Value: C:\...\Microsoft Visual Studio 9.0\Common7\Tools\
```

where of course the ellipses should be filled in by "Program Files" or "Program Files (x86)" depending on your system. Doing so wraps up the compiler issue, finally. All that remains is to install Scrapy using pip. If there are any remnants of Scrapy left on the system from previous unsuccessful attempts, remove them using the uninstall feature of pip.

Final environment variable check In addition to the new "VS90COMNTOOLS" environment variable that we defined above, the PATH variable needs to include various maps. Here we present the relevant maps we had in a successful 64-bit build:

```
C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;
C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;
C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\bin\;
C:\python27\;
C:\python27\Scripts;
```

```
C:\openssl-win64\bin;
C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE;
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\BIN;
C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\Tools;
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\VCPackages;
```

From that point, following the directions in the official tutorial [3] should see one's first program built using Scrapy work just fine.

References

[1] http://scrapy.org/

[2] http://doc.scrapy.org/en/latest/

[3] http://doc.scrapy.org/en/latest/intro/tutorial.html

[4] http://slproweb.com/products/Win32OpenSSL.html

[5] http://www.microsoft.com/en-us/download/details.aspx?id=29

[6] http://www.microsoft.com/en-us/download/details.aspx?id=15336

[7] http://go.microsoft.com/?linkid=7729279

[8] http://www.microsoft.com/en-us/download/details.aspx?id=3138

[9] http://twistedmatrix.com/trac/wiki/Downloads

[10] https://raw.github.com/pypa/pip/master/contrib/get-pip.py

[11] http://pypi.python.org/pypi/setuptools

[12] http://pypi.python.org/pypi/zope.interface

[13] http://lxml.de/installation.html

[14] http://www.lfd.uci.edu/~gohlke/pythonlibs/

[15] GitHub: "How to Install Scrapy 0.14 in a 64 bit Windows 7 Environment"

[16] http://www.egenix.com/products/python/pyOpenSSL/

[17] Jabur, V. "Compiling python 2.7 modules on windows 32 and 64 using msvc 2008 express." Weblog accessible at: http://blog.victorjabur.com/