Installing / Upgrading

PyMongo is in the Python Package Index.

Warning: **Do not install the "bson" package.** PyMongo comes with its own bson package; doing "pip install bson" or "easy_install bson" installs a third-party package that is incompatible with PyMongo.

Installing with pip

We recommend using pip to install pymongo on all platforms:

```
$ python -m pip install pymongo
```

To get a specific version of pymongo:

```
$ python -m pip install pymongo==3.1.1
```

To upgrade using pip:

```
$ python -m pip install --upgrade pymongo
```

Note: pip does not support installing python packages in .egg format. If you would like to install PyMongo from a .egg provided on pypi use easy_install instead.

Installing with easy_install

To use `easy_install` from setuptools do:

```
$ python -m easy_install pymongo
```

To upgrade do:

```
$ python -m easy_install -U pymongo
```

Dependencies

PyMongo supports CPython 2.6, 2.7, 3.3+, PyPy, and PyPy3.

Optional dependencies for GSSAPI and TLS:

GSSAPI authentication requires pykerberos on Unix or WinKerberos on Windows. The correct dependency can be installed automatically along with PyMongo:

```
$ python -m pip install pymongo[gssapi]
```

TLS / SSL support may require ipaddress and certifi or wincertstore depending on the Python version in use. The necessary dependencies can be installed along with PyMongo:

```
$ python -m pip install pymongo[tls]
```

You can install both dependencies automatically with the following command:

```
$ python -m pip install pymongo[gssapi,tls]
```

Other optional packages:

- backports.pbkdf2, improves authentication performance with SCRAM-SHA-1, the default authentication mechanism for MongoDB 3.0+. It especially improves performance on Python older than 2.7.8, or on Python 3 before Python 3.4.
- monotonic adds support for a monotonic clock, which improves reliability in environments where clock adjustments are frequent. Not needed in Python 3.3+.

Dependencies for installing C Extensions on Unix

MongoDB, Inc. does not provide statically linked binary packages for Unix flavors other than OSX. To build the optional C extensions you must have the GNU C compiler (gcc) installed. Depending on your flavor of Unix (or Linux distribution) you may also need a python development package that provides the necessary header files for your version of Python. The package name may vary from distro to distro.

Debian and Ubuntu users should issue the following command:

```
$ sudo apt-get install build-essential python-dev
```

Users of Red Hat based distributions (RHEL, CentOS, Amazon Linux, Oracle Linux, Fedora, etc.) should issue the following command:

```
$ sudo yum install gcc python-devel
```

Installing from source

If you'd rather install directly from the source (i.e. to stay on the bleeding edge), install the C extension dependencies then check out the latest source from github and install the driver from the resulting tree:

```
$ git clone git://github.com/mongodb/mongo-python-driver.git pymongo
$ cd pymongo/
$ python setup.py install
```

Installing from source on OSX

If you want to install PyMongo from source on OSX you will have to install the following to build the C extensions:

Snow Leopard (10.6) - Xcode 3 with 'UNIX Development Support'.

Snow Leopard Xcode 4: The Python versions shipped with OSX 10.6.x are universal binaries. They support i386, PPC, and (in the case of python2.6) x86_64. Xcode 4 removed support for PPC, causing the distutils version shipped with Apple's builds of Python to fail to build the C extensions if you have Xcode 4 installed. There is a workaround:

```
# For Apple-supplied Python2.6 (installed at /usr/bin/python2.6) and
# some builds from python.org
$ env ARCHFLAGS='-arch i386 -arch x86_64' python -m easy_install pymongo
```

See http://bugs.python.org/issue11623 for a more detailed explanation.

Lion (10.7) and newer - PyMongo's C extensions can be built against versions of Python 2.7 >= 2.7.4 or Python 3.3+ downloaded from python.org. In all cases Xcode must be installed with 'UNIX Development Support'.

Xcode 5.1: Starting with version 5.1 the version of clang that ships with Xcode throws an error when it encounters compiler flags it doesn't recognize. This may cause C extension builds to fail with an error similar to:

```
clang: error: unknown argument: '-mno-fused-madd' [-Wunused-command-line-argument-hard-e
```

There are workarounds:

```
# Apple specified workaround for Xcode 5.1
# easy_install
$ ARCHFLAGS=-Wno-error=unused-command-line-argument-hard-error-in-future easy_install py
# or pip
$ ARCHFLAGS=-Wno-error=unused-command-line-argument-hard-error-in-future pip install pym

# Alternative workaround using CFLAGS
# easy_install
$ CFLAGS=-Qunused-arguments easy_install pymongo
# or pip
$ CFLAGS=-Qunused-arguments pip install pymongo
```

Installing from source on Windows

If you want to install PyMongo with C extensions from source the following requirements apply to both CPython and ActiveState's ActivePython:

64-bit Windows

For Python 3.5 and newer install Visual Studio 2015. For Python 3.3 and 3.4 install Visual Studio 2010. For Python 2.6 and 2.7 install Visual Studio 2008, or the Microsoft Visual C++ Compiler for Python 2.7. You must use the full version of Visual Studio 2010 or 2008 as Visual C++ Express does not provide 64-bit compilers. Make sure that you check the "x64 Compilers and Tools" option under Visual C++.

32-bit Windows

For Python 3.5 and newer install Visual Studio 2015.

For Python 3.3 and 3.4 install Visual C++ 2010 Express.

For Python 2.6 and 2.7 install Visual C++ 2008 Express SP1.

Installing Without C Extensions

By default, the driver attempts to build and install optional C extensions (used for increasing performance) when it is installed. If any extension fails to build the driver will be installed anyway but a warning will be printed.

If you wish to install PyMongo without the C extensions, even if the extensions build properly, it can be done using a command line option to *setup.py*:

```
$ python setup.py --no_ext install
```

Building PyMongo egg Packages

Some organizations do not allow compilers and other build tools on production systems. To install PyMongo on these systems with C extensions you may need to build custom egg packages. Make sure that you have installed the dependencies listed above for your operating system then run the following command in the PyMongo source directory:

```
$ python setup.py bdist_egg
```

The egg package can be found in the dist/ subdirectory. The file name will resemble “pymongo-3.4-py2.7-linux-x86_64.egg” but may have a different name depending on your platform and the version of python you use to compile.

Warning: These “binary distributions,” will only work on systems that resemble the environment on which you built the package. In other words, ensure that operating systems and versions of Python and architecture (i.e. “32” or “64” bit) match.

Copy this file to the target system and issue the following command to install the package:

```
$ sudo python -m easy_install pymongo-3.4-py2.7-linux-x86_64.egg
```

Installing a beta or release candidate

MongoDB, Inc. may occasionally tag a beta or release candidate for testing by the community before final release. These releases will not be uploaded to pypi but can be found on the github tags page. They can be installed by passing the full URL for the tag to pip:

```
$ python -m pip install https://github.com/mongodb/mongo-python-driver/archive/3.4rc0.ta
```

or easy_install:

```
$ python -m easy_install https://github.com/mongodb/mongo-python-driver/archive/3.4rc0.ta
```