Installing / Upgrading

PyMongo is in the Python Package Index.

Microsoft Windows

We recommend using the *MS Windows installers* available from the Python Package Index.

Installing with pip

We prefer pip to install pymongo on platforms other than Windows:

```
$ pip install pymongo
```

To get a specific version of pymongo:

```
$ pip install pymongo==2.1.1
```

To upgrade using pip:

```
$ pip install --upgrade pymongo
```

Installing with easy_install

If you must install pymongo using setuptools do:

```
$ easy_install pymongo
```

To upgrade do:

```
$ easy_install -U pymongo
```

Dependencies for installing C Extensions on Unix

10gen 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
```

RedHat, CentOS, and Fedora users should issue the following command:

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

OSX

10gen provides pre-built egg packages for Apple provided Python versions on Snow Leopard (2.5, 2.6), Lion (2.5, 2.6, 2.7) and Mountain Lion (2.5, 2.6, 2.7). If you want to install PyMongo for other Python versions (or from source) 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

# For 32-bit-only Python (/usr/bin/python2.5) and some builds
# from python.org
$ env ARCHFLAGS='-arch i386' python -m easy_install pymongo
```

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

Lion (10.7) - PyMongo's C extensions can be built against versions of Python >= 3.2 downloaded from python.org. Building against versions older than 3.2.3 requires **Xcode 4.1**. Any version of Xcode 4 can be used to build the C extensions against 3.2.3 and newer. In all cases Xcode must be installed with 'UNIX Development Support'. See the following for more information:

http://bugs.python.org/issue13590

http://hg.python.org/cpython/file/v3.2.3/Misc/NEWS#l198

Mountain Lion (10.8) - PyMongo's C extensions can be built against versions of Python >= 3.3rc1 downloaded from python.org with no special requirements. If you want to build against the python.org provided 3.2.3 you must have MacOSX10.6.sdk in /Developer /SDKs. See the following for more information:

http://bugs.python.org/issue14499

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 Windows

> **Note:** 10gen provides pre-built exe installers for 32-bit and 64-bit Windows. We recommend that users install those packages (available from pypi).

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

64-bit Windows

For Python 3.3 install Visual Studio 2010. For Python 3.2 and older install Visual Studio 2008. In either case you must use the full version 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 3.3 install Visual C++ 2010 Express.

For Python 2.6 through 3.2 install Visual C++ 2008 Express SP1.

For Python 2.4 or 2.5 you must install MingW32 then run the following command to install:

```
python setup.py build -c mingw32 install
```

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.

In *certain cases*, you might wish to install the driver without the C extensions, even if the extensions build properly. This 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-2.2-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 easy_install pymongo-2.2-py2.7-linux-x86_64.egg
```

Installing a release candidate

10gen may occasionally tag a 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:

```
$ pip install https://github.com/mongodb/mongo-python-driver/tarball/2.2rc1
```

or easy_install:

```
$ easy_install https://github.com/mongodb/mongo-python-driver/tarball/2.2rc1
```