

```
=====
MySQLdb Installation
=====
```

#### Prerequisites

-----

##### + Python 2.3.4 or higher

- \* <http://www.python.org/>
- \* Versions lower than 2.2 WON'T WORK.
- \* 2.2.x MIGHT work, or have partial functionality.
- \* 2.4 is tested and works.
- \* Red Hat Linux:
  - Make sure you have the Python development headers and libraries (python-devel).

##### + MySQL 3.23.32 or higher

- \* <http://www.mysql.com/downloads/>
- \* Versions lower than 3.22 definitely WON'T WORK.
- \* Versions lower than 3.22.19 might not work.
- \* MySQL-3.22 is deprecated in favor of 3.23, but still supported.
- \* MySQL-3.23 is supported, but slightly deprecated.
- \* MySQL-4.0 is supported.
- \* MySQL-4.1 is mostly supported; the new prepared statements API is not yet supported, and probably won't be until MySQLdb-1.3 or 2.0.
- \* MySQL-5.0 and newer are not currently supported, but might work.
- \* MaxDB, formerly known as SAP DB (and maybe Adabas D?), is a completely different animal. Use the sapdb.sql module that comes with MaxDB.
- \* Red Hat Linux packages:
  - mysql-devel to compile
  - mysql and/or mysql-devel to run
- \* MySQL.com RPM packages:
  - MySQL-devel to compile
  - MySQL-shared if you want to use their shared library. Otherwise you'll get a statically-linked module, which may or may not be what you want.
  - MySQL-shared to run if you compiled with MySQL-shared installed

- \* Transactions (particularly InnoDB tables) are supported for MySQL-3.23 and up. You may need a special package from your vendor with this support turned on. If you have Gentoo Linux, set either of the berkdb or innodb USE flags on your server, and comment out "skip-innodb" in /etc/mysql/my.cnf for InnoDB table support.

#### + zlib

- \* Required for MySQL-3.23 and newer.
- \* Red Hat Linux
  - zlib-devel to compile
  - zlib to run

#### + openssl

- \* May be needed for MySQL-4.0 or newer, depending on compilation options.

#### + C compiler

- \* Most free software-based systems already have this, usually gcc.
- \* Most commercial UNIX platforms also come with a C compiler, or you can also use gcc.
- \* If you have some Windows flavor, you usually have to pay extra for this, or you can use Cygwin\_.

.. \_Cygwin: <http://www.cygwin.com/>

### Building and installing

The setup.py script uses mysql\_config to find all compiler and linker options, and should work as is on any POSIX-like platform, so long as mysql\_config is in your path.

Depending on which version of MySQL you have, you may have the option of using three different client libraries:

mysqlclient  
mostly but not guaranteed thread-safe

mysqlclient\_r  
thread-safe, use if you can

mysqld  
embedded server

mysqlclient\_r is used by default. To use one of the others, set the environment variable mysqlclient to the name of the library you want to use. In a Bourne-style shell, use::

```
$ export mysqlclient=mysqlclient
```

Only do this if you don't have the thread-safe library (mysqlclient\_r) or you want to use the embedded server (mysqld).

Finally, putting it together::

```
$ tar xzf MySQL-python-1.2.0.tar.gz
$ cd MySQL-python-1.2.0
$ python setup.py build
$ su # or use sudo
# python setup.py install
```

NOTE: You must export environment variables for setup.py to see them. Depending on what shell you prefer, you may need to use "export" or "set -x" (bash and other Bourne-like shells) or "setenv" (csh-like shells).

Windows

.....

I don't do Windows. However if someone provides me with a package for Windows, I'll make it available. Don't ask me for help with Windows because I can't help you.

Generally, though, running setup.py is similar to above::

```
C:\...> python setup.py install
C:\...> python setup.py bdist_wininst
```

The latter example should build a Windows installer package, if you have the correct tools. In any event, you *must* have a C compiler. Additionally, you have to set an environment variable (mysqlroot) which is the path to your MySQL installation. In theory, it would be possible to get this information out of the registry, but like I said, I don't do Windows, but I'll accept a patch that does this.

Zope

....

If you are using a binary package of Zope, you need run setup.py with the python executable that came with Zope. Otherwise, you'll install into the wrong Python tree and Zope (ZMySQLDA) will not be able to find \_mysql.

With zope.org's Zope-2.5.1-linux2-x86 binary tarball, you'd do something like this::

```
$ export ZOPEBIN=".../Zope-2.5.1-linux2-x86/bin" # wherever you unpacked it
$ $ZOPEBIN/python setup.py install # builds and installs
```

Binary Packages

-----

I don't plan to make binary packages any more. However, if someone contributes one, I will make it available. Several OS vendors have their own packages available.

RPMs

....

If you prefer to install RPMs, you can use the bdist\_rpm command with setup.py. This only builds the RPM; it does not install it. You may want to use the --python=XXX option, where XXX is the name of the Python executable, i.e. python, python2, python2.1; the default is python. Using this will incorporate the Python executable name into the package name for the RPM so you have install the package multiple

times if you need to support more than one version of Python.

Red Hat Linux  
.....

MySQL-python is pre-packaged in Red Hat Linux 7.x and newer. This likely includes Fedora Core and Red Hat Enterprise Linux. You can also build your own RPM packages as described above.

Debian GNU/Linux  
.....

Packaged as `python-mysql`\_::

# apt-get install python-mysql

.. \_`python-mysql`: [http://packages.debian.org/cgi-bin/search\\_packages.pl?keywords=python-mysql&searchon=names&subword=1&version=all&release=all](http://packages.debian.org/cgi-bin/search_packages.pl?keywords=python-mysql&searchon=names&subword=1&version=all&release=all)

Gentoo Linux  
.....

Packaged as `mysql-python`\_. Gentoo is also my development platform::

# emerge sync  
# emerge mysql-python  
# emerge zmysqlda # if you use Zope

.. \_`mysql-python`: <http://packages.gentoo.org/search/?sstring=mysql-python>

BSD  
...

MySQL-python is a ported package in FreeBSD, NetBSD, and OpenBSD, although the name may vary to match OS conventions.

License  
-----

GPL or the original license based on Python 1.5.2's license.

:Author: Andy Dustman <andy@dustman.net>  
:Revision: \$Id: README,v 1.11 2005/02/08 01:28:19 adustman Exp \$