MySQL Programming

MySQL Installation and Configuration
Contributed by McGraw−Hill/Osborne
2004−04−28
[Send Me Similar Content When Posted]
[Add Developer Shed Headlines To Your Site]

DISCUSS NEWS SEND PRINT PDF

Article Index:

In this chapter, Vikram discusses how to obtain, install, configure, and test the MySQL server on Unix and Windows. It also explains the differences between the different MySQL sever versions available, with a view to helping you select the right one for your needs. This excerpt comes from chapter three of *MySQL: The Complete Reference*, by Vikram Vaswani (McGraw−Hill/Osborne, ISBN 0−07−222477−0, 2004).

One of the nicer things about MySQL is the fact that it’s an open−source application licensed under the General Public License (GPL); users are encouraged to download, modify, and use it free of charge. This open licensing policy has played an important role in MySQL’s widespread acceptance and popularity in the developer community; according to statistics published on the MySQL web site, MySQL is in use at more than 4 million sites worldwide, with more than 25,000 copies of the MySQL database server downloaded every day!

In this chapter, I’m going to help add to those numbers, by taking you through the process of obtaining, installing, configuring, and testing the MySQL server on your workstation. This chapter covers installation of both binary and source versions on UNIX and Microsoft Windows, and it also explains the differences between the MySQL server versions available, with a view to helping you select the right one for your needs.

Obtaining MySQL

The first order of business is to drop by the official MySQL web site (http://www.mysql.com/downloads/mysql.html) and get yourself a copy of the last stable release of the software. This isn’t necessarily as easy as it sounds−−like ice−cream, MySQL comes in many flavors, and you’ll need to select the one that’s most appropriate for your needs.

You’ll need to make two primary decisions when selecting which MySQL distribution to download and use:

- Choosing which version to install
- Choosing between binary and source distributions

Choosing Which Version to Install

MySQL AB currently makes the following two versions of the MySQL database server available on its web site:

- ***MySQL Standard*** This is the standard version of the MySQL database server, which includes support for both the regular, non−transactional tables and the newer, transaction−safe tables. It is suited for production environments requiring a stable, flexible, and robust database engine.

- ***MySQL Max*** This version includes the feature set of the standard version, together with newer, more experimental capabilities. It is not always best suited for production environments, since it usually includes a fair share of not−completely−stable enhancements.

Both these versions of MySQL are licensed under the GPL and may be freely downloaded and used under the terms of that license.

In most cases, MySQL Standard is the version you should use−−it’s the version used in all the examples in this book, and it’s stable, feature−rich, and well−suited for most common applications. You should select MySQL Max only if that version includes new features that you need or are keen to try out−−or if you’re a geek with a penchant for living life on the bleeding edge all the time.

Remember: this is chapter three of *MySQL: The Complete Reference*, by Vikram Vaswani (McGraw−Hill/Osborne, ISBN 0−07−222477−0, 2004). Vikram is the founder of Melonfire, and has had numerous articles featured on Dev Shed.
Buy this book now.

MySQL AB makes both source and binary distributions of the MySQL database server available for download on its web site. As of this writing, binary distributions are available for Linux, Solaris, FreeBSD, Mac OS X, Windows 95/98/2000/XP/NT/ ME, HP−UX, IBM AIX, SCO OpenUNIX, and SGI Irix, and source distributions are available for both Windows and UNIX platforms.

In most cases, it’s preferable to use a precompiled binary distribution rather than a source distribution−−MySQL AB puts in a lot of time and effort to create binary distributions for various platforms, and using these distributions is generally considered a Good Idea. These precompiled binary distributions are also easier to install than source distributions, and they are optimized by MySQL’s developers to ensure maximum performance on the target systems.

That said, for a number of possible reasons, a source distribution might work better for you than a binary distribution:

- You’d like to recompile MySQL with compile−time options different from the defaults provided by the MySQL team. (For example, you might like to use a different value for the installation path.)
- You’d like to compile a smaller, lighter version of MySQL that doesn’t include all the features (and overhead) of the standard binary distribution.
- You’d like to enable support for newer, experimental features that are disabled by default in the standard binaries.
- You’d like to view or make modifications to the source code of the application.
- You have waaaaay too much time on your hands, and you’d like to impress the pretty girl next door with your deep and profound knowledge of MySQL’s internals. By the way, according to surveys, only six percent of MySQL users are women.

If you’re new to MySQL, I’d recommend that you go with a binary distribution−−source distributions are typically used only by experienced developers who either need to tweak MySQL’s default values for their own purposes or who are interested in studying the source code to see how it works. Such users usually also have the time, inclination, and expertise to diagnose and troubleshoot compilation and configuration issues that may arise during the installation process.

Pay It Forward

It should be noted that, in addition to the two versions discussed in the section “Choosing Which Version to Install,” MySQL AB also offers two other versions of the MySQL database server, MySQL Pro and MySQL Classic, which are licensed commercially and are therefore not freely available for download.

What’s the difference between the two? MySQL Pro includes support for transactions (with the exception of the licensing terms, it is identical in every way to MySQL Standard), while MySQL Classic does not.

If you’re interested in purchasing a commercial license for these versions of the MySQL database server, drop by http://order.mysql.com and find out more about what your greenbacks buy you.

Once you’ve figured out which version of MySQL you need, point your browser to http://www.mysql.com/downloads/mysql.html and select the appropriate file from the choices available. The MySQL software is also mirrored on a number of other sites around the world, and you can obtain a list of mirrors from http://www.mysql.com/downloads/mirrors.html. You can make your download more efficient by selecting the site that is geographically closest to you. Once your download is complete, save it to a directory on your hard drive and skip to the section titled “Installing and Configuring MySQL,” a bit later in this chapter.

What’s in a Name?

You’ve probably noticed that MySQL distributions are named using a particular naming scheme−−for example, `mysql−4.0.9−gamma.tar.gz`. The numbers in the file name indicate the version number of the software, with the additional suffix indicating the stability level of the software (*alpha*, *beta*, or *gamma*).

Note that we will assume, throughout this chapter, that you are using MySQL version 4.0.15, and that all your downloads are saved in the `/tmp` (UNIX) or `c:\temp` (Windows) directory, and I will reference that directory in subsequent instructions.

Once you’ve got yourself a copy of MySQL, it’s time to install and configure it for your specific platform. This section includes information on how to do just that, for both Windows and UNIX platforms, using both binary and source distributions.

It should be noted at the outset that this section is designed merely to provide an overview and general guide to the process of installing and configuring MySQL. It is not intended as a replacement for the installation documentation that ships with MySQL. If you encounter difficulties installing or configuring the various programs described here, drop by the MySQL web site or search the mailing lists for detailed troubleshooting information and advice.

Installing and Configuring MySQL on UNIX

MySQL is available in binary form for almost all versions of UNIX and can even be compiled for those UNIX variants for which no binary distribution exists. This section will discuss installing and configuring MySQL on Linux using both source and binary distributions; the process for other UNIX variants is similar, although you should refer to the documentation included with the MySQL distribution for platform−specific notes.

Installing MySQL from a Binary RPM Distribution

The recommended way to install MySQL on a Linux system is via RPM. MySQL AB makes the following RPMs available for download on its web site:

- ***MySQL*** The MySQL database server, which manages databases and tables, controls user access, and processes SQL queries

- ***MySQL−client*** MySQL client programs, which makes it possible to connect to, and interact with, the server

- ***MySQL−devel*** Libraries and header files that come in handy when compiling other programs that use MySQL
- ***MySQL−shared*** Shared libraries for the MySQL client
- ***MySQL−bench*** Benchmark and performance testing tools for the MySQL database server

Needful Things

MySQL software distributions are usually packaged in Zip, tar (tape archive), or RPM (RPM Package Manager) format, and they can range from 7 to 20 MB in size in compressed form, and up to 100 MB in size in uncompressed form. Depending on the format you select, you will need appropriate unpackaging tools to extract the files from the source archive. For Zip and tar files, you will need GNU tar and GNU gunzip, available from http://www.gnu.org.

For RPM files, you will need rpm, available from http://www.rpm.org. Additionally, if you're planning on compiling and installing MySQL from a source distribution, you will need a C++ compiler like gcc on UNIX or Visual C++ on Windows. The gcc compiler is available from http://www.gnu.org/software/gcc, while Visual C++ is available from http://msdn.microsoft.com/visualc.

The MySQL RPMs listed here are all built on a SuSE Linux system, but they'll usually work on other Linux variants with no difficulty.

Installing an RPM distribution of MySQL is extremely simple, and it involves running only a single command−−the rpm command−−for each RPM you wish to install. Here's how you go about doing it:

1. First ensure that you're logged in as root:

```
[user@host]# su – root
```

2. Switch to the directory containing the RPMs:

```
[root@host]# cd /tmp
```

3. Install the MySQL database server by executing the following command (remember to replace the filename in italics with the file name of your RPM):

```
[root@host]# rpm −i MySQL−4.0.9−0.i386.rpm
```

RPM does the following things to get MySQL up and running on your system:

- Copies the MySQL binaries to appropriate locations on your system (usually, binaries go to `/usr/bin and /usr/sbin`, while databases and tables are stored in `/var/lib/mysql`)
- Adds a `mysql` user/group to the system to handle all MySQL−related operational

and administrative tasks

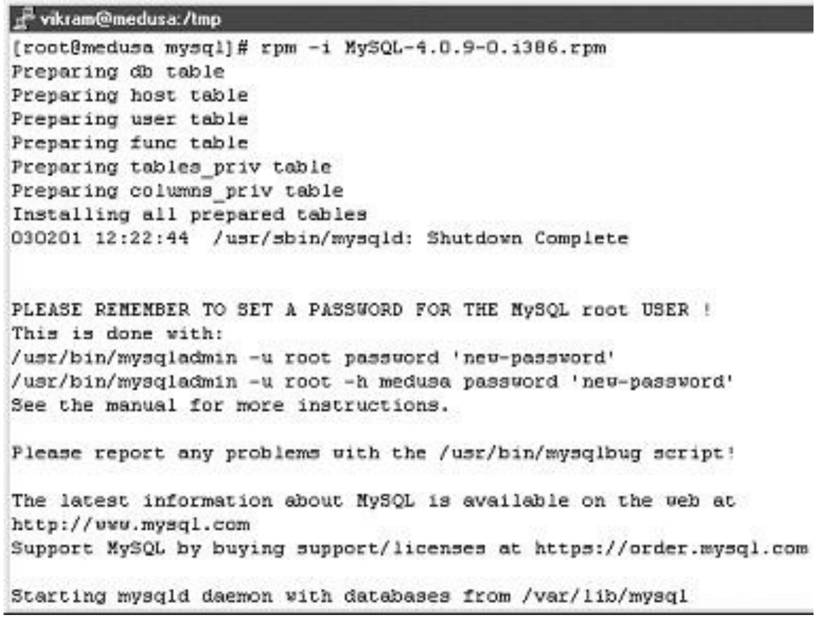
- Alters ownership of the MySQL binaries so that they are owned by the `mysql` user/group

- Creates and initializes the MySQL grant tables

- Adds appropriate entries to your system's startup scripts so that the MySQL server starts up automatically at boot time

- Starts the server so that you can begin using it immediately

Figure 1 shows a snippet of what you might see during the installation process:



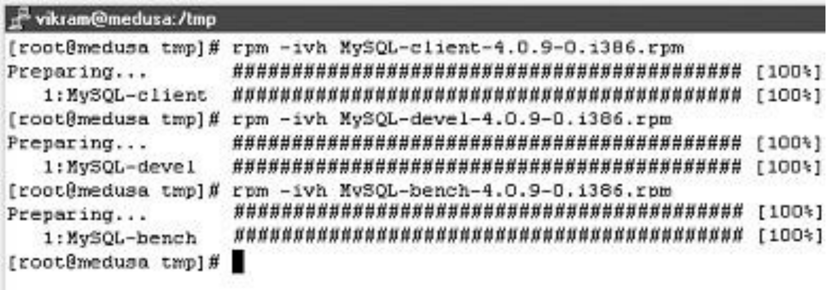
```
[root@medusa mysql]# rpm -i MySQL-4.0.9-0.i386.rpm
Preparing db table
Preparing host table
Preparing user table
Preparing func table
Preparing tables_priv table
Preparing columns_priv table
Installing all prepared tables
030201 12:22:44  /usr/sbin/mysqld: Shutdown Complete


PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
This is done with:
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h medusa password 'new-password'
See the manual for more instructions.

Please report any problems with the /usr/bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at https://order.mysql.com

Starting mysqld daemon with databases from /var/lib/mysql
```

FIGURE 1 *Installation of the MySQL server via RPM*

4. Now install the remaining RPMs in a similar manner:

```
[root@host]# rpm -i MySQL-client-4.0.9-0.i386.rpm
[root@host]# rpm -i MySQL-devel-4.0.9-0.i386.rpm
[root@host]# rpm -i MySQL-shared-4.0.9-0.i386.rpm
[root@host]# rpm -i MySQL-bench-4.0.9-0.i386.rpm
```

Figure 2 displays what you should see while performing this task.



```
[root@medusa tmp]# rpm -ivh MySQL-client-4.0.9-0.i386.rpm
Preparing...                ########################################### [100%]
   1:MySQL-client           ########################################### [100%]
[root@medusa tmp]# rpm -ivh MySQL-devel-4.0.9-0.i386.rpm
Preparing...                ########################################### [100%]
   1:MySQL-devel            ########################################### [100%]
[root@medusa tmp]# rpm -ivh MySQL-bench-4.0.9-0.i386.rpm
Preparing...                ########################################### [100%]
   1:MySQL-bench            ########################################### [100%]
[root@medusa tmp]# █
```

FIGURE 2 *Installation of ancillary MySQL tools and utilities via RPM*

Note that it’s *necessary* to install only the server; however, I would recommend thatyou install the client as well so that you can interact with the server from the system console. The benchmark utilities should be installed only if you plan to test MySQL performance, while the libraries and header files come in handy when you’re compiling other utilities or tools that use MySQL (for example, the PHP scripting language).

Once installation has been successfully completed, you should move later in the chapter to the section titled “Testing MySQL” to verify that everything is working as it should.

In case you’re using a Linux distribution that doesn’t support RPM, you can also install MySQL using a binary tarball from the MySQL web site.

Installing from a binary distribution essentially means that you need to perform the installation steps manually rather than letting RPM automatically take care of it for you.

Here’s how you go about doing it:

1. Ensure that you’re logged in as root:

```
[user@host]# su − root
```

2. Extract the content of the tarball to an appropriate directory on your system−−I’ll assume this location is `/usr/local/`. Remember to replace the file name in italics with the file name of your tarball.

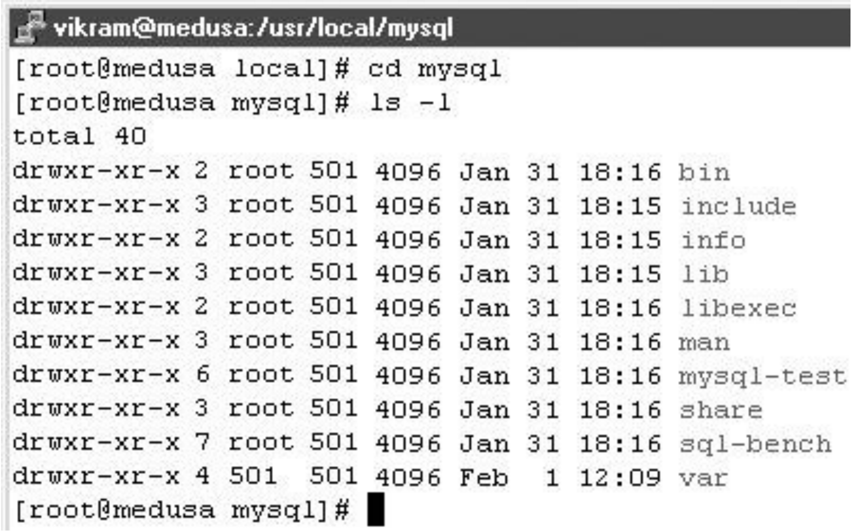
```
[root@host]# cd /usr/local
[root@host]# tar −xzvf
mysql−standard−4.0.9−gamma−pc−linux−i686.tar.gz
```

The MySQL files should get extracted into a directory named according to the format `mysql−`*`version−os−architecture`*−−for example, `mysql−standard−4.0.9−gamma−pc−linux−i686`.

3. Now you’ll notice that the directory created in the previous step has a somewhat long and cumbersome directory name−−something like `mysql−standard−4.0.9−gamma−pc−linux−i686`. For ease of use, create a soft link to this directory named mysql in the same location.

```
[root@host]# ln −s
mysql−standard−4.0.9−gamma−pc−linux−i686 mysql
```

4. Change into this directory, and take a look at how the files are arranged. You should see something like Figure 3. (Take a look at the sidebar entitled “Up a Tree” for more information on what each directory contains.)



```
[root@medusa local]# cd mysql
[root@medusa mysql]# ls -l
total 40
drwxr-xr-x 2 root 501 4096 Jan 31 18:16 bin
drwxr-xr-x 3 root 501 4096 Jan 31 18:15 include
drwxr-xr-x 2 root 501 4096 Jan 31 18:15 info
drwxr-xr-x 3 root 501 4096 Jan 31 18:15 lib
drwxr-xr-x 2 root 501 4096 Jan 31 18:16 libexec
drwxr-xr-x 3 root 501 4096 Jan 31 18:16 man
drwxr-xr-x 6 root 501 4096 Jan 31 18:16 mysql-test
drwxr-xr-x 3 root 501 4096 Jan 31 18:16 share
drwxr-xr-x 7 root 501 4096 Jan 31 18:16 sql-bench
drwxr-xr-x 4 501  501 4096 Feb  1 12:09 var
[root@medusa mysql]# 
```

FIGURE 3 *The directory structure obtained on unpackaging of a MySQL binary tarball on Linux*

Up a Tree

If you have the time (and the inclination), you might find it instructive to explore the MySQL directory structure to help you better understand where the important files are located.

For a binary distribution, the directory structure for a typical MySQL installation looks like this:

```
<mysql-install-root>
|-- bin           [client and server binaries]
|-- data          [databases and error log]
|-- include       [header files]
|-- lib           [compiled libraries]
|-- man           [manual pages]
|-- mysql-test    [test suite]
|-- share         [error messages in different languages]
|-- scripts       [startup, shutdown and initialization
scripts]
|-- sql-bench     [queries and data files for benchmark
tests]
|-- support-files [sample configuration files]
|-- tests         [test cases]
```

For a source distribution, the directory structure for a typical MySQL installation looks like this:

```
<mysql-install-root>
```

```
|-- bin        [client binaries]
|-- libexec    [server binaries]
|-- var        [databases and error log]
|-- lib        [compiled libraries]
|-- include    [header files]
|-- info       [info pages]
|-- man        [manual pages]
|-- mysql-test [test suite]
|-- share      [error messages in different languages]
|-- sql-bench  [queries and data files for benchmark tests]
```

Take a look at the documentation that ships with the MySQL distribution for a more detailed discussion of this directory structure.

5. The MySQL database server can run as either the system root user or any other user on the system. From the security point of view, it’s considered a bad idea to run the MySQL database server as root ; hence, it becomes necessary for you to create a special mysql user and group for this purpose.

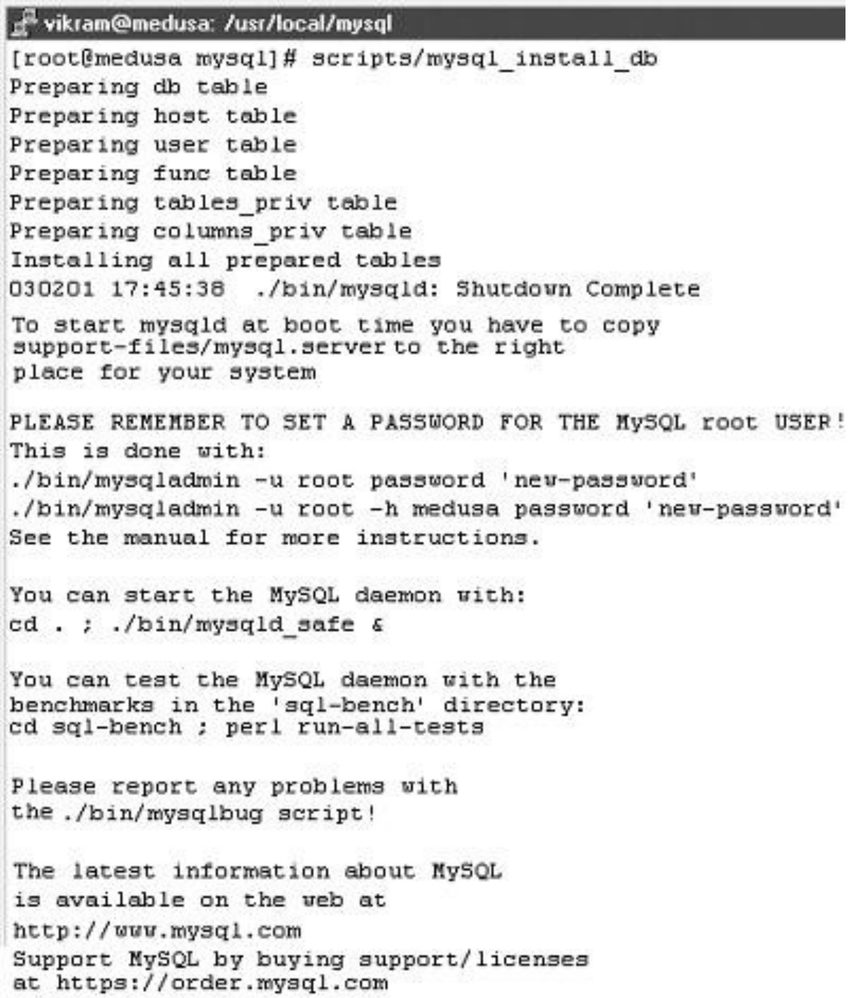
You can accomplish this using the `groupadd` and `useradd` commands:

```
[root@host]# groupadd mysql
[root@host]# useradd –g mysql mysql
```

6. Run the initialization script, `mysql_install_db`, that ships with the program:

```
[root@host]#
/usr/local/mysql/scripts/mysql_install_db
```

Figure 4 demonstrates what you should see when you do this.



vikram@medusa: /usr/local/mysql

```
[root@medusa mysql]# scripts/mysql_install_db
Preparing db table
Preparing host table
Preparing user table
Preparing func table
Preparing tables_priv table
Preparing columns_priv table
Installing all prepared tables
030201 17:45:38  ./bin/mysqld: Shutdown Complete

To start mysqld at boot time you have to copy
support-files/mysql.server to the right
place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER!
This is done with:
./bin/mysqladmin -u root password 'new-password'
./bin/mysqladmin -u root -h medusa password 'new-password'
See the manual for more instructions.

You can start the MySQL daemon with:
cd . ; ./bin/mysqld_safe &

You can test the MySQL daemon with the
benchmarks in the 'sql-bench' directory:
cd sql-bench ; perl run-all-tests

Please report any problems with
the ./bin/mysqlbug script!

The latest information about MySQL
is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses
at https://order.mysql.com
```

FIGURE 4 *The output of running the MySQL initialization script*

As you can see from the output in the figure, this initialization script prepares and installs the various MySQL base tables and also sets up default access permissions for MySQL.

7. Alter the ownership of the MySQL binaries so that they are owned by root:

```
[root@host]# chown −R root /usr/local/mysql
```

8. Now ensure that the newly−minted `mysql` user has read/write access to the MySQL data directories:

```
[root@host]# chown −R mysql /usr/local/mysql/data
[root@host]# chgrp −R mysql /usr/local/mysql
```

9. Start the MySQL server by manually running the `mysqld` daemon:

```
[root@host]# /usr/local/mysql/bin/mysqld_safe
–user=mysql &
```

MySQL should start up normally, reading the base tables created in `/usr/local/mysql/data`.

Once installation has been successfully completed, you can skip to the section titled “Testing MySQL,” later in this chapter, to verify that your server is functioning properly.

If you’re planning to install MySQL from a source distribution, you’ll need to untar the source tree and go through the traditional `configure−make−make install` cycle to get MySQL up and running. This is a fairly time−consuming and complex process, and it’s one that shouldn’t really be attempted by novice users; however, if you’re determined to do it, here’s how:

1. Ensure that you’re logged in as root:

```
[user@host]# su − root
```

2. Switch to the directory containing the source tarball, and extract the files within it. (Note that you will need approximately 80 MB of free space for the source tree.)

```
[root@host]# cd /tmp
[root@host]# tar −xzvf mysql−4.0.9−gamma.tar.gz
```

Remember to replace the file name in italics with the file name of your source tarball.

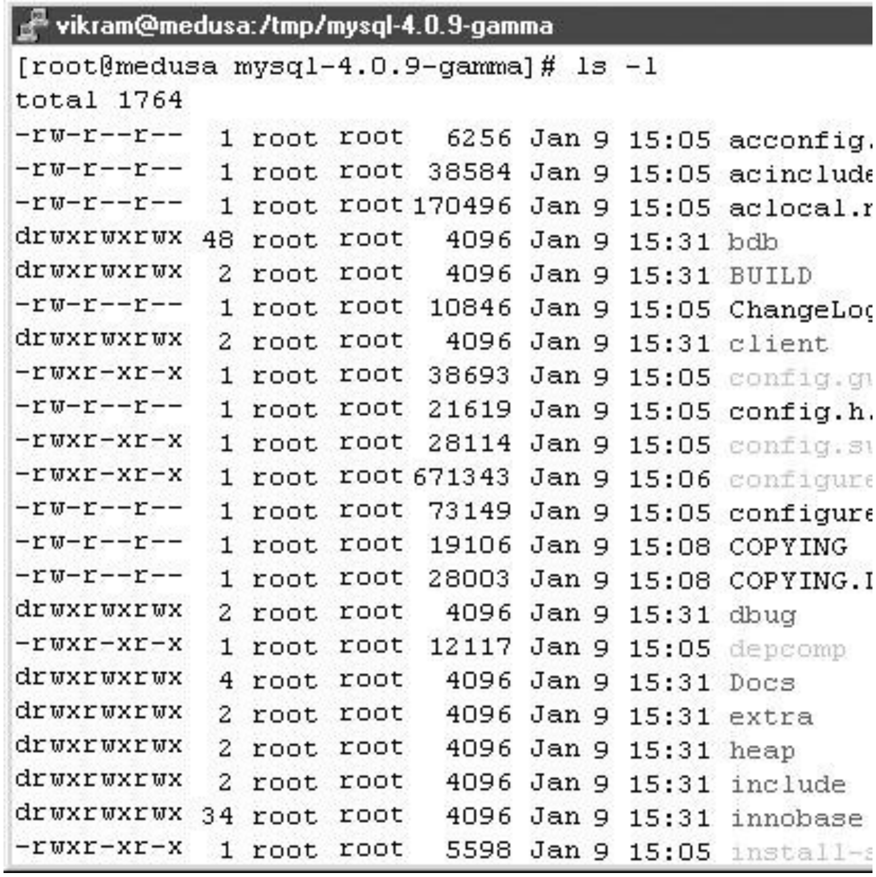
3. Move into the directory containing the source code,

```
[root@host]# cd mysql−4.0.9−gamma
```

and take a look at the contents with `ls`:

```
[root@host]# ls −l
```

You should see something like Figure 5.



```
vikram@medusa:/tmp/mysql-4.0.9-gamma
[root@medusa mysql-4.0.9-gamma]# ls -l
total 1764
-rw-r--r--  1 root root   6256 Jan 9 15:05 acconfig.
-rw-r--r--  1 root root  38584 Jan 9 15:05 acinclude
-rw-r--r--  1 root root 170496 Jan 9 15:05 aclocal.
drwxrwxrwx 48 root root   4096 Jan 9 15:31 bdb
drwxrwxrwx  2 root root   4096 Jan 9 15:31 BUILD
-rw-r--r--  1 root root  10846 Jan 9 15:05 ChangeLo
drwxrwxrwx  2 root root   4096 Jan 9 15:31 client
-rwxr-xr-x  1 root root  38693 Jan 9 15:05 config.g
-rw-r--r--  1 root root  21619 Jan 9 15:05 config.h.
-rwxr-xr-x  1 root root  28114 Jan 9 15:05 config.s
-rwxr-xr-x  1 root root 671343 Jan 9 15:06 configur
-rw-r--r--  1 root root  73149 Jan 9 15:05 configur
-rw-r--r--  1 root root  19106 Jan 9 15:08 COPYING
-rw-r--r--  1 root root  28003 Jan 9 15:08 COPYING.
drwxrwxrwx  2 root root   4096 Jan 9 15:31 dbug
-rwxr-xr-x  1 root root  12117 Jan 9 15:05 depcomp
drwxrwxrwx  4 root root   4096 Jan 9 15:31 Docs
drwxrwxrwx  2 root root   4096 Jan 9 15:31 extra
drwxrwxrwx  2 root root   4096 Jan 9 15:31 heap
drwxrwxrwx  2 root root   4096 Jan 9 15:31 include
drwxrwxrwx 34 root root   4096 Jan 9 15:31 innobase
-rwxr-xr-x  1 root root   5598 Jan 9 15:05 install-
```

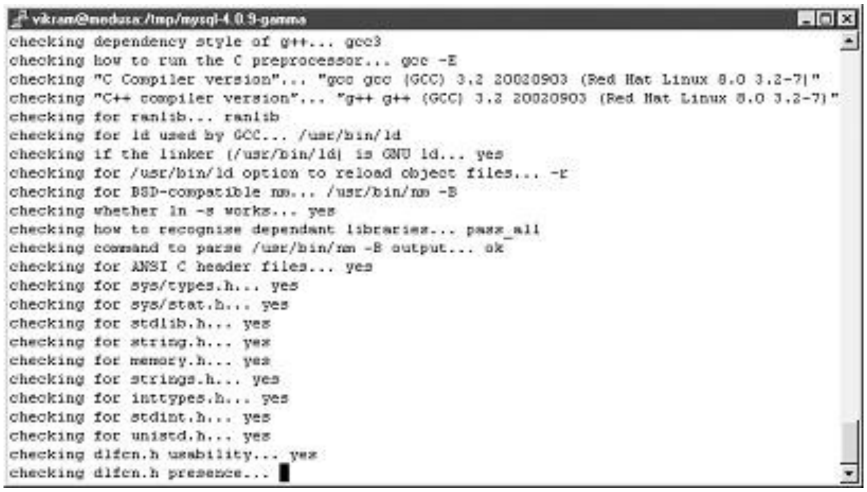
FIGURE 5 *The directory structure obtained on unpackaging of a MySQL source tarball on Linux*

Take a look at the sidebar entitled "Up a Tree" for more information on what each directory contains.

4. Now, set variables for the compile process via the included configure script. (Note the use of the `--prefix` argument to configure, which sets the default installation path for the compiled binaries.)

```
[root@host]# ./configure --prefix=/usr/local/mysql
```

You should see a few screens of output (Figure 6 has a sample) as configure configures and sets up the variables needed for the compilation process.



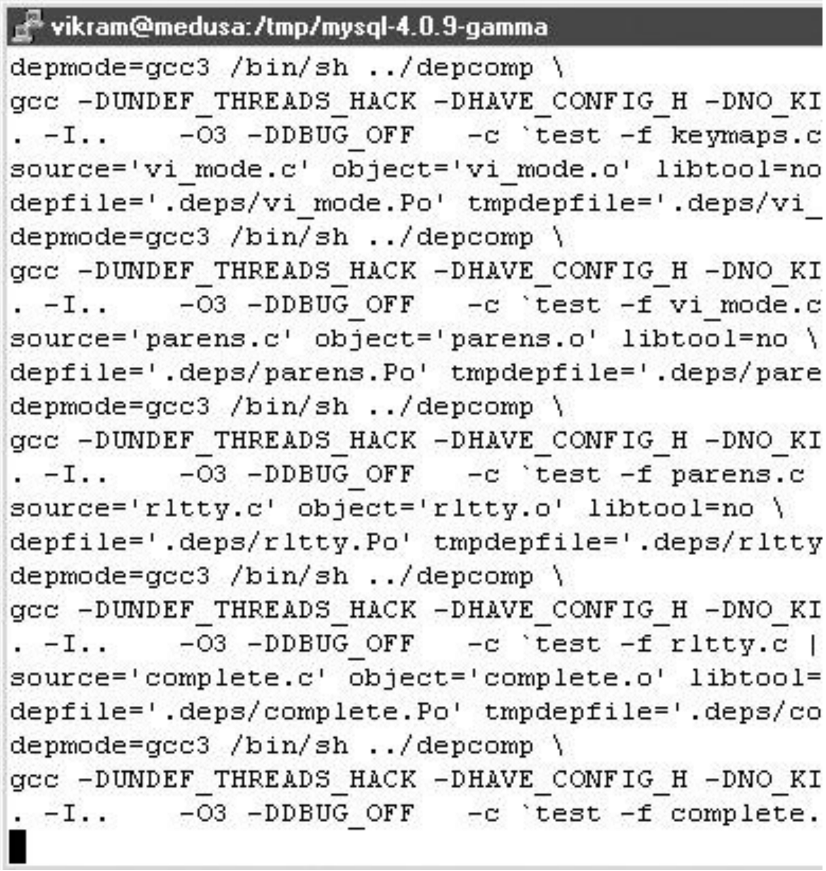
```
checking dependency style of g++... gcc3
checking how to run the C preprocessor... gcc -E
checking "C Compiler version"... "gcc gcc (GCC) 3.2 20020903 (Red Hat Linux 8.0 3.2-7|"
checking "C++ compiler version"... "g++ g++ (GCC) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)"
checking for ranlib... ranlib
checking for ld used by GCC... /usr/bin/ld
checking if the linker (/usr/bin/ld| is GNU ld... yes
checking for /usr/bin/ld option to reload object files... -r
checking for BSD-compatible nm... /usr/bin/nm -B
checking whether ln -s works... yes
checking how to recognise dependant libraries... pass_all
checking command to parse /usr/bin/nm -B output... ok
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking dlfcn.h usability... yes
checking dlfcn.h presence... █
```

FIGURE 6 *Configuring the MySQL source tree on Linux*

5. Now compile the program using `make`:

```
[root@host]# make
```

Watch as your screen fills up with all manner of strange symbols and characters (see Figure 7).



```
depmode=gcc3 /bin/sh ../depcomp \
gcc -DUNDEF_THREADS_HACK -DHAVE_CONFIG_H -DNO_KI
. -I..    -O3 -DDBUG_OFF   -c `test -f keymaps.c
source='vi_mode.c' object='vi_mode.o' libtool=no
depfile='.deps/vi_mode.Po' tmpdepfile='.deps/vi_
depmode=gcc3 /bin/sh ../depcomp \
gcc -DUNDEF_THREADS_HACK -DHAVE_CONFIG_H -DNO_KI
. -I..    -O3 -DDBUG_OFF   -c `test -f vi_mode.c
source='parens.c' object='parens.o' libtool=no \
depfile='.deps/parens.Po' tmpdepfile='.deps/pare
depmode=gcc3 /bin/sh ../depcomp \
gcc -DUNDEF_THREADS_HACK -DHAVE_CONFIG_H -DNO_KI
. -I..    -O3 -DDBUG_OFF   -c `test -f parens.c
source='rltty.c' object='rltty.o' libtool=no \
depfile='.deps/rltty.Po' tmpdepfile='.deps/rltty
depmode=gcc3 /bin/sh ../depcomp \
gcc -DUNDEF_THREADS_HACK -DHAVE_CONFIG_H -DNO_KI
. -I..    -O3 -DDBUG_OFF   -c `test -f rltty.c |
source='complete.c' object='complete.o' libtool=
depfile='.deps/complete.Po' tmpdepfile='.deps/co
depmode=gcc3 /bin/sh ../depcomp \
gcc -DUNDEF_THREADS_HACK -DHAVE_CONFIG_H -DNO_KI
. -I..    -O3 -DDBUG_OFF   -c `test -f complete.
█
```

FIGURE 7 *Building MySQL on Linux*

The compilation process takes a fair amount of time (refer to the sidebar titled “Watching the

Clock” for my empirical observations on how long you’ll be waiting), so this is a good time to get yourself a cup of coffee or check your mail.

Now that you’re all done, you can test to ensure that everything is working properly.

6. Run the following command:

```
[root@host]# make tests
```

Handcrafting Your Build

You can pass configure a number of command−line options that affect the build process. Here’s a list of the more interesting ones:

- ***--prefix*** Sets the prefix for installation paths
- ***--without-server*** Disables compilation of the server, and compiles only the MySQL client programs and libraries
- ***--localstatedir*** Sets the location in which the MySQL databases will be stored
- ***--with-charset*** Sets a default character set
- ***--with-debug*** Turns on extended debugging
- ***--with-raid*** Enables RAID support
- ***--with-embedded-server*** Builds the libmysqld embedded server library
- ***--without-query-cache*** Disables the query cache
- ***--without-debug*** Disables debugging routines
- ***--with-openssl*** Includes OpenSSL support

Use the configure `--help` command to get a complete list of options.

Watching the Clock

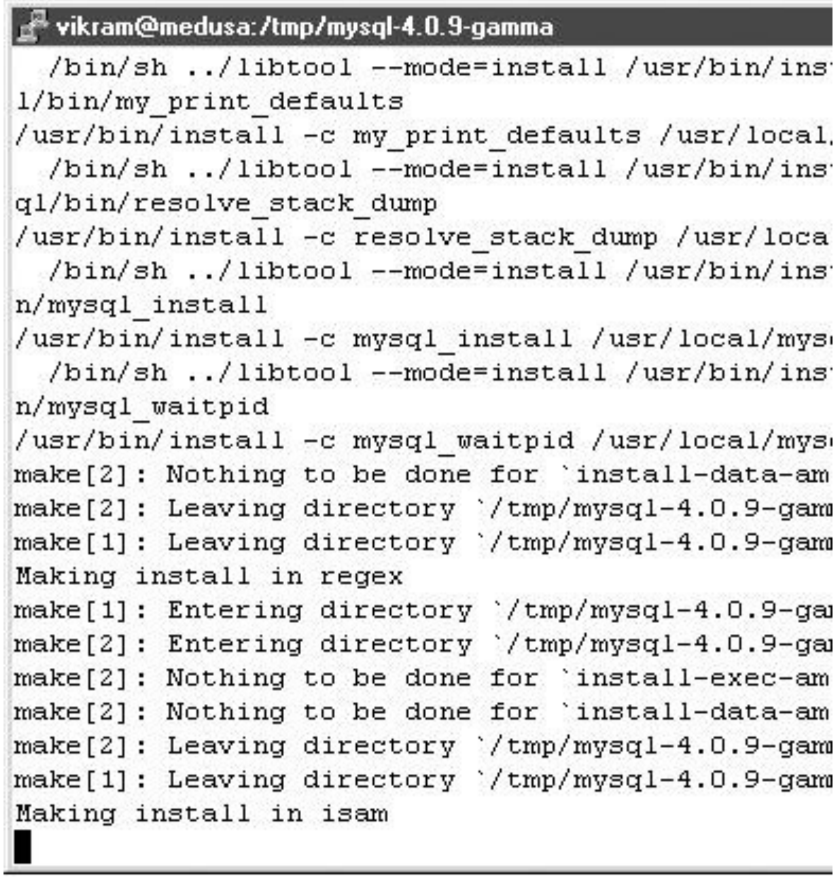
Compiling MySQL is a fairly time−consuming process, and you should be prepared to spend anywhere between 15 to 60 minutes on the task. The following table contains some empirical observations on the time taken to compile the program on various hardware configurations:

System Configuration	***Time Taken to Compile MySQL***
Pentium−II@350 MhZ, 64 MB RAM	45 minutes
Pentium−III@700MhZ, 256 MB RAM	30 minutes
AMD Athlon MP 1500+, SuSE Linux 7.3	8 minutes
AMD Opteron@2x1.6 GHz, UnitedLinux 1.0	7 minutes
Apple PowerMac G4@2x1.2 GHz, Mac OS X 10.2.4	14 minutes
Compaq AlphaServer DS20@500 MHz, SuSe Linux 7.0	17 minutes
HP 9000/800/A500−7X, HP−UX 11.11	14 minutes
IBM RS/6000, AIX 4.3.3	35 minutes
Intel Itanium2@900 MHz, Red Hat AS 2.1	14 minutes
MIPS R5000@500 MHz, SGI IRIX 6.5	2 hours 30 minutes

7. Install the MySQL binaries to their new home in `/usr/local/mysql`:

```
[root@host]# make install
```

Figure 8 demonstrates what your screen should look like during the installation process.



```
  /bin/sh ../libtool --mode=install /usr/bin/ins
l/bin/my_print_defaults
/usr/bin/install -c my_print_defaults /usr/local
  /bin/sh ../libtool --mode=install /usr/bin/ins
ql/bin/resolve_stack_dump
/usr/bin/install -c resolve_stack_dump /usr/loca
  /bin/sh ../libtool --mode=install /usr/bin/ins
n/mysql_install
/usr/bin/install -c mysql_install /usr/local/mys
  /bin/sh ../libtool --mode=install /usr/bin/ins
n/mysql_waitpid
/usr/bin/install -c mysql_waitpid /usr/local/mys
make[2]: Nothing to be done for `install-data-am
make[2]: Leaving directory `/tmp/mysql-4.0.9-gam
make[1]: Leaving directory `/tmp/mysql-4.0.9-gam
Making install in regex
make[1]: Entering directory `/tmp/mysql-4.0.9-ga
make[2]: Entering directory `/tmp/mysql-4.0.9-ga
make[2]: Nothing to be done for `install-exec-am
make[2]: Nothing to be done for `install-data-am
make[2]: Leaving directory `/tmp/mysql-4.0.9-gam
make[1]: Leaving directory `/tmp/mysql-4.0.9-gam
Making install in isam
```

***FIGURE 8** Installing compiled MySQL binaries on Linux*

8. Create the special mysql user and group with the `groupadd` and `useradd` commands:

```
[root@host]# groupadd mysql
[root@host]# useradd −g mysql mysql
```

9. Run the initialization script, `mysql_install_db`, which ships with the program, to prepare MySQL for operation:

```
[root@host]#
/usr/local/mysql/scripts/mysql_install_db
```

10. Alter the ownership of the MySQL binaries so that they are owned by root:

```
[root@host]# chown −R root /usr/local/mysql
```

Now ensure that the newly minted mysql user has read/write access to the MySQL data directories:

```
[root@host]# chown −R mysql /usr/local/mysql/var
[root@host]# chgrp −R mysql /usr/local/mysql
```

11. Start the MySQL server by manually running the mysqld daemon:

```
[root@host]# /usr/local/mysql/bin/mysqld_safe
−−user=mysql &
```

MySQL should start up normally, reading the base tables created in `/usr/local/mysql/var`.

At this point, you can proceed to the section titled “Testing MySQL” to verify that everything is working as it should.

Version Control

In case you’re wondering, all the binaries used when developing this book have been built on Linux using the following software versions:

- mysqld 4.0.15−standard
- mysqld−4.1−alpha
- rpm 4.1
- gcc 3.2
- tar 1.13.25
- gunzip 1.3.3
- unzip 5.50
- make 3.79.1
- autoconf 2.53
- automake 1.6.3

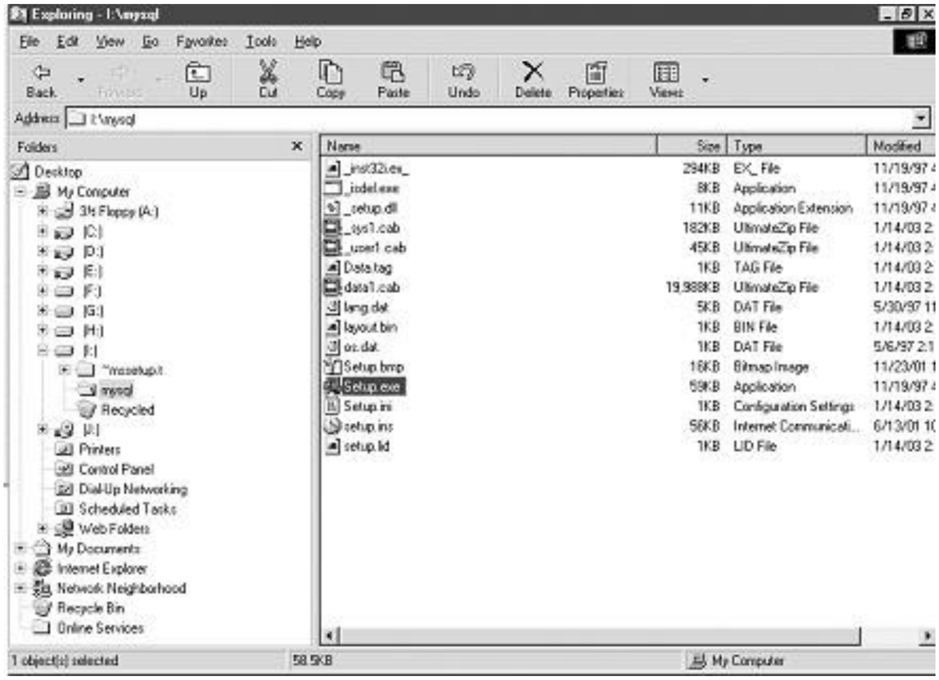
MySQL is available in both source and binary form for Windows 95/98/Me/2000/ XP/NT. Most often, you will want to use the binary distribution, which comes with an automated installer and allows you to get MySQL up and running on your Windows

system in just a few minutes. However, if you’re the type who likes rolling your own, MySQL AB also makes MySQL source code available for Windows. In this section, I’ll be exploring the installation of both source and binary distributions on Windows 98 and Windows NT.

Installing MySQL from a Binary Distribution

Installing a binary distribution of MySQL on Windows is a fairly simple process−−all you need to do is point and click your way through the installer provided with the distribution. Here’s how:

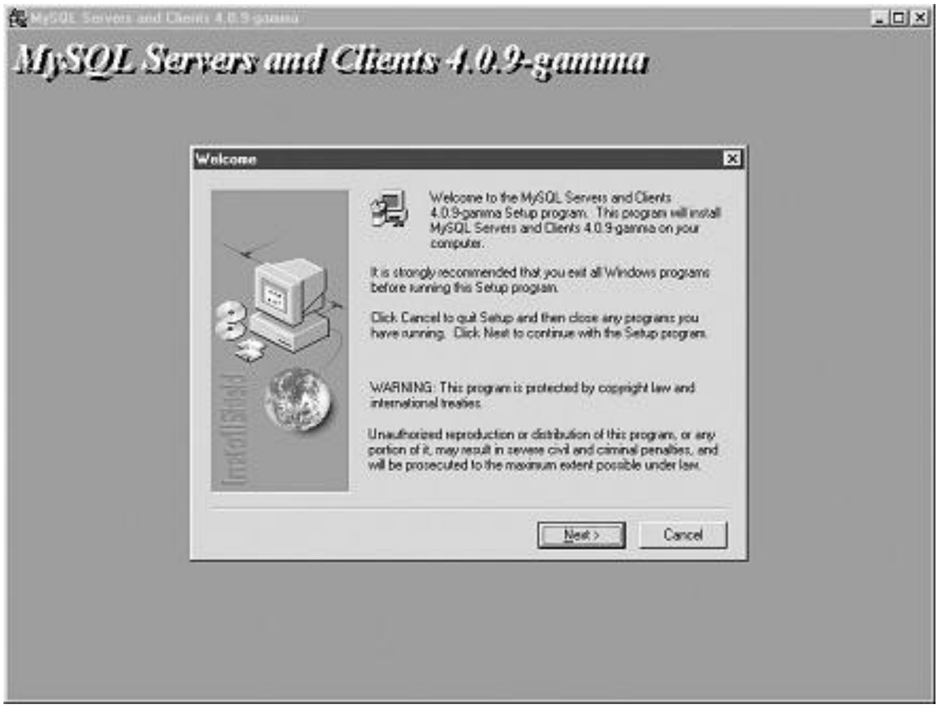
1. Log in as an administrator (if you're using Windows NT or Windows 2000) and unzip the distribution archive to a temporary directory on your system. After extraction, your directory should look something like Figure 9.



Name	Size	Type	Modified
inst32i.ex	294KB	EX_ File	11/19/97 4
_isdel.exe	8KB	Application	11/19/97 4
_setup.dll	11KB	Application Extension	11/19/97 4
_sys1.cab	182KB	UltimateZip File	1/14/03 2
_user1.cab	45KB	UltimateZip File	1/14/03 2
Data.tag	1KB	TAG File	1/14/03 2
data1.cab	19,988KB	UltimateZip File	1/14/03 2
lang.dat	5KB	DAT File	5/30/97 11
layout.bin	1KB	BIN File	1/14/03 2
os.dat	1KB	DAT File	5/6/97 2:1
Setup.bmp	16KB	Bitmap Image	11/23/01 1
Setup.exe	59KB	Application	11/19/97 4
Setup.ini	1KB	Configuration Settings	1/14/03 2
setup.ins	56KB	Internet Communicati...	6/13/01 10
setup.lid	1KB	LID File	1/14/03 2

FIGURE 9 *The directory structure created on unpackaging a MySQL binary distribution for Windows*

2. Double−click the setup.exe file to begin the installation process. You should see a welcome screen (Figure 10). Click Next.


MySQL Servers and Clients 4.0.9-gamma
MySQL Servers and Clients 4.0.9-gamma
Welcome
InstallShield
Next >
Cancel

Welcome to the MySQL Servers and Clients 4.0.9-gamma Setup program. This program will install MySQL Servers and Clients 4.0.9-gamma on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

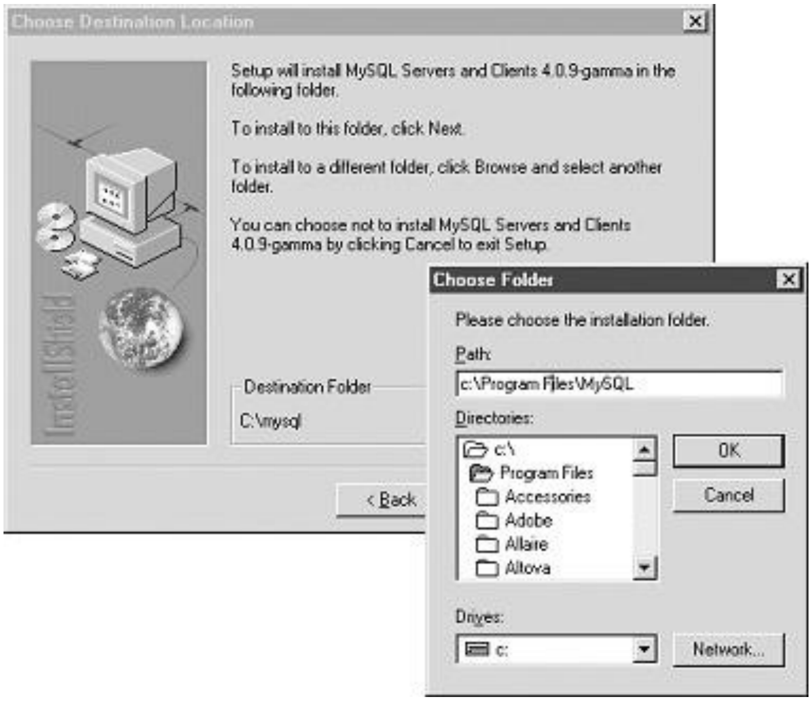
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

FIGURE 10 *MySQL installation on Windows*

3. Click Browse and select the directory in which MySQL is to be installed−−in this example, select `c:\program files\mysql` (Figure 11). Click Next.



Choose Destination Location

Setup will install MySQL Servers and Clients 4.0.9-gamma in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

You can choose not to install MySQL Servers and Clients 4.0.9-gamma by clicking Cancel to exit Setup.

Destination Folder: C:\mysql

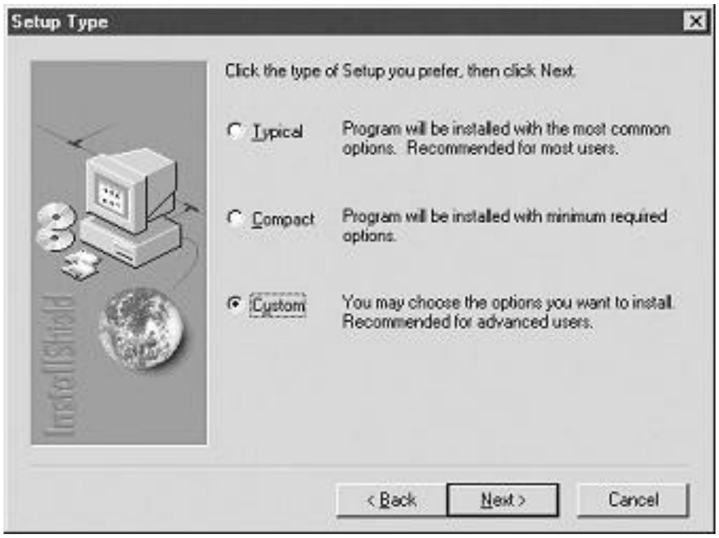
Choose Folder

Please choose the installation folder.

Path: c:\Program Files\MySQL

FIGURE 11 *Selecting installation directory on Windows*

4. Select the type of installation required (Figure 12). Click Next.



Setup Type

Click the type of Setup you prefer, then click Next.

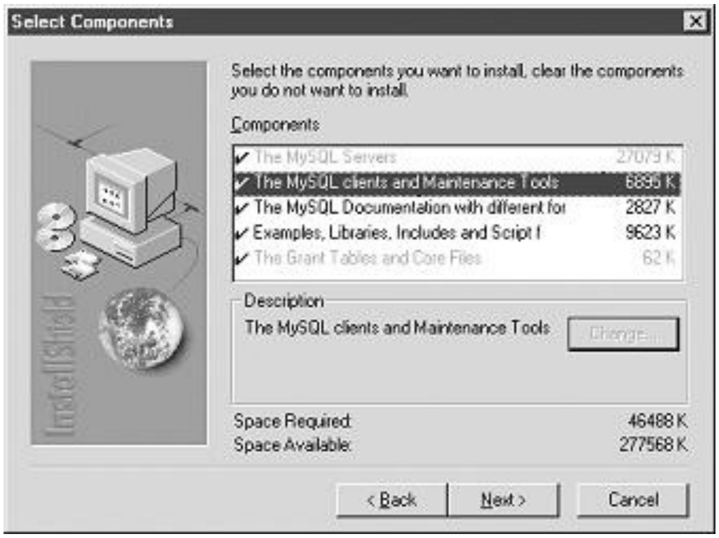
○ Typical — Program will be installed with the most common options. Recommended for most users.

○ Compact — Program will be installed with minimum required options.

● Custom — You may choose the options you want to install. Recommended for advanced users.

FIGURE 12 *Selecting installation type*

Most often, a Typical Installation will do; however, if you’re the kind who likes tweaking default settings, select the Custom Installation option, click Next, and then decide which components of the package should be installed (Figure 13).



Select Components

Select the components you want to install, clear the components you do not want to install.

Components

- ✓ The MySQL Servers 27079 K
- ✓ The MySQL clients and Maintenance Tools 6895 K
- ✓ The MySQL Documentation with different for 2827 K
- ✓ Examples, Libraries, Includes and Script f 9623 K
- ✓ The Grant Tables and Core Files 62 K

Description

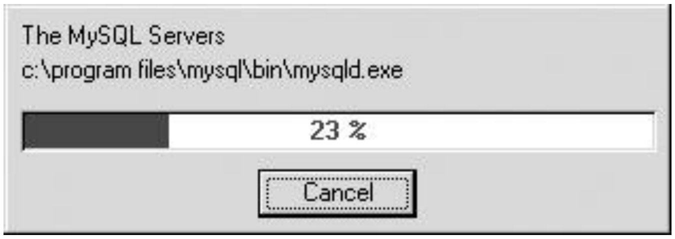
The MySQL clients and Maintenance Tools

Space Required: 46488 K

Space Available: 277568 K

FIGURE 13 *Selecting components for a custom installation on Windows*

5. Click Next, and MySQL should now begin installing to your system (Figure 14).

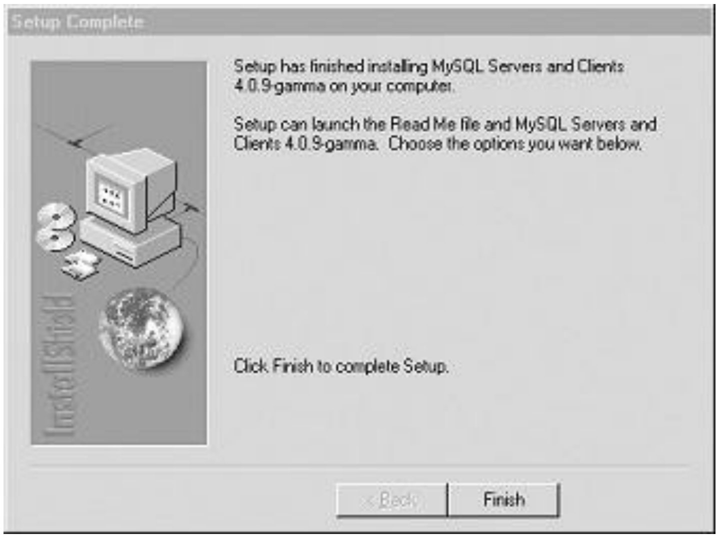


The MySQL Servers
c:\program files\mysql\bin\mysqld.exe

23 %

FIGURE 14 *Installation in progress on Windows*

6. After installation is complete, you should see a screen like Figure 15. Click Finish.



Setup Complete

Setup has finished installing MySQL Servers and Clients 4.0.9-gamma on your computer.

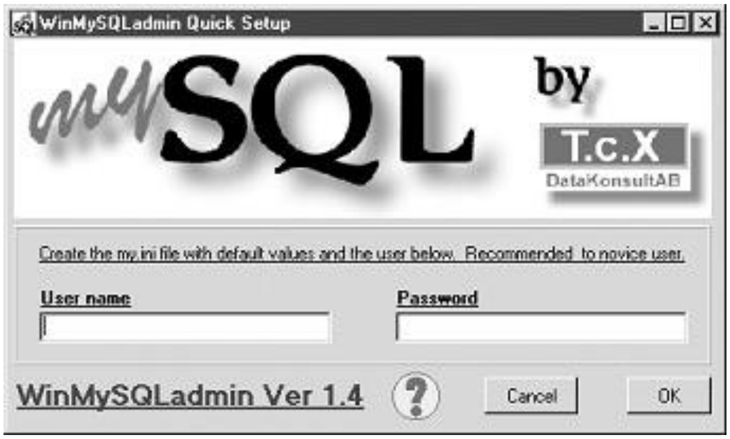
Setup can launch the Read Me file and MySQL Servers and Clients 4.0.9-gamma. Choose the options you want below.

Click Finish to complete Setup.

FIGURE 15 *Installation successfully completed on Windows*

You should now be able to start the MySQL server by diving into the `bin\` subdirectory of your MySQL installation and launching the WinMySQLadmin tool (`winmysqladmin.exe`). This tool provides a graphical user interface (GUI) to MySQL configuration, and it is by far the simplest way to configure MySQL on Windows systems.

The first time you start WinMySQLadmin, you will be asked for the name and password of the user which the server should run as (Figure 16).



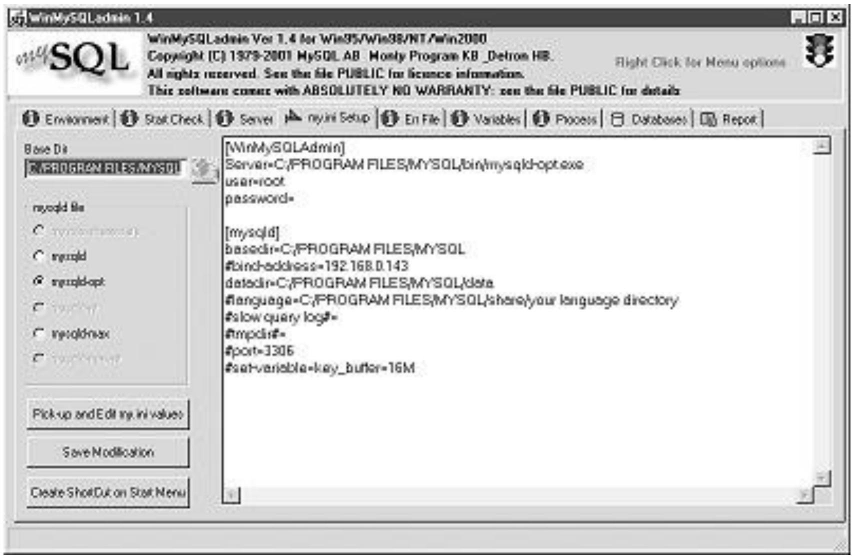
Create the my.ini file with default values and the user below. Recommended to novice user.

User name

Password

FIGURE 16 *Configuring the MySQL user on Windows via WinMySQLadmin*

After you have entered this information, WinMySQLadmin will automatically create the MySQL configuration file (named `my.ini`) and populate it with appropriate values for your system. You can edit these values at any time through the “my.ini Setup” tab of the main WinMySQLadmin application window (see Figure 17).



WinMySQLadmin Ver 1.4 for Win95/Win98/NT/win2000
Copyright (C) 1979-2001 MySQL AB Monty Program KB _Detron HB.
All rights reserved. See the file PUBLIC for licence information.
This software comes with ABSOLUTELY NO WARRANTY: see the file PUBLIC for details

Base Dir: C:/PROGRAM FILES/MYSQL

● mysqld-opt

```
[WinMySQLAdmin]
Server=C:/PROGRAM FILES/MYSQL/bin/mysqld-opt.exe
user=root
password=

[mysqld]
basedir=C:/PROGRAM FILES/MYSQL
#bind-address=192.168.0.143
datadir=C:/PROGRAM FILES/MYSQL/data
#language=C:/PROGRAM FILES/MYSQL/share/your language directory
#slow query log#=
#tmpdir#=
#port=3306
#set-variable=key_buffer=16M
```

FIGURE 17 *Editing MySQL configuration on Windows via WinMySQLadmin*

NOTE *You can also start the MySQL server by directly launching the* `mysqld.exe` *or* `mysqld-nt.exe` *binary from the* `bin\` *subdirectory of your MySQL installation.*

Once the server has started,WinMySQLadmin will minimize to a green icon in your Windows taskbar notification area. You can now proceed to test the server as described in the section “Testing MySQL” to ensure that everything is working as it should.

Note that you can bring the WinMySQLadmin application back to the foreground at any time by right−clicking the taskbar icon and choosing “Show Me” from the pop−up menu (see Figure 18).

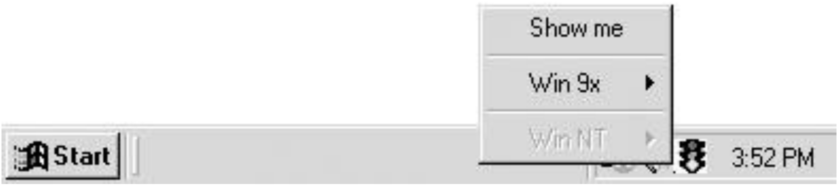
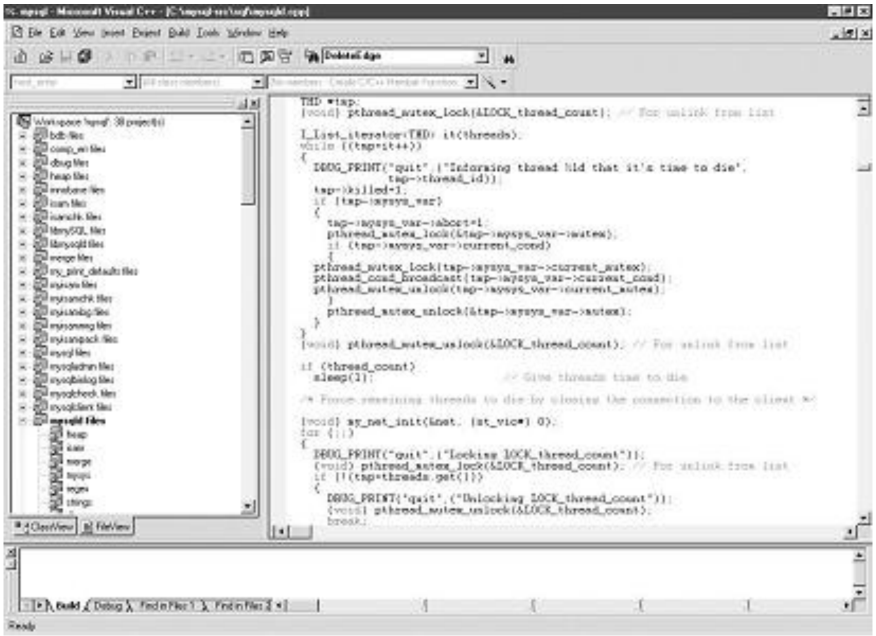


FIGURE 18 *Using the WinMySQLadmin system tray icon*

While compiling MySQL for Windows from the source archive is not something that’s generally recommended−−it’s far safer, not to mention easier, to use the provided binaries−−it’s certainly doable, assuming you have a copy of the Visual C++ 6.0 compiler (with Service Pack 5 and the preprocessor package). Here’s how it’s done:

1. Unzip the source archive to a working directory on your system.

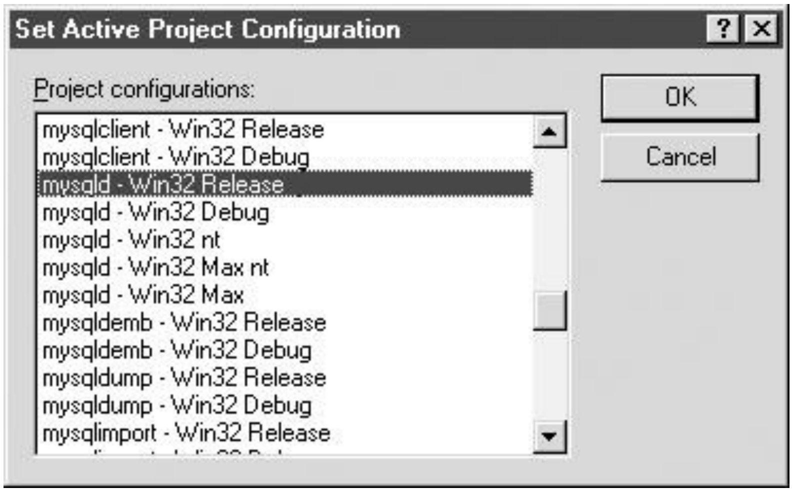
2. Launch the Visual C++ compiler, and open the mysql.dsw workspace from the working directory. You should see a window like Figure 19.



```
  THD *tmp;
  (void) pthread_mutex_lock(&LOCK_thread_count); // For unlink from list

  I_List_iterator<THD> it(threads);
  while ((tmp=it++))
  {
    DBUG_PRINT("quit",("Informing thread %ld that it's time to die",
                       tmp->thread_id));
    tmp->killed=1;
    if (tmp->mysys_var)
    {
      tmp->mysys_var->abort=1;
      pthread_mutex_lock(&tmp->mysys_var->mutex);
      if (tmp->mysys_var->current_cond)
      {
    pthread_mutex_lock(tmp->mysys_var->current_mutex);
    pthread_cond_broadcast(tmp->mysys_var->current_cond);
    pthread_mutex_unlock(tmp->mysys_var->current_mutex);
      }
      pthread_mutex_unlock(&tmp->mysys_var->mutex);
    }
  }
  (void) pthread_mutex_unlock(&LOCK_thread_count); // For unlink from list

  if (thread_count)
    sleep(1);                    // Give threads time to die

  /* Force remaining threads to die by closing the connection to the client */

  (void) my_net_init(&net, (st_vio*) 0);
  for (;;)
  {
    DBUG_PRINT("quit",("Locking LOCK_thread_count"));
    (void) pthread_mutex_lock(&LOCK_thread_count); // For unlink from list
    if (!(tmp=threads.get()))
    {
      DBUG_PRINT("quit",("Unlocking LOCK_thread_count"));
      (void) pthread_mutex_unlock(&LOCK_thread_count);
      break;
```

FIGURE 19 *The MySQL workspace in Visual C++*

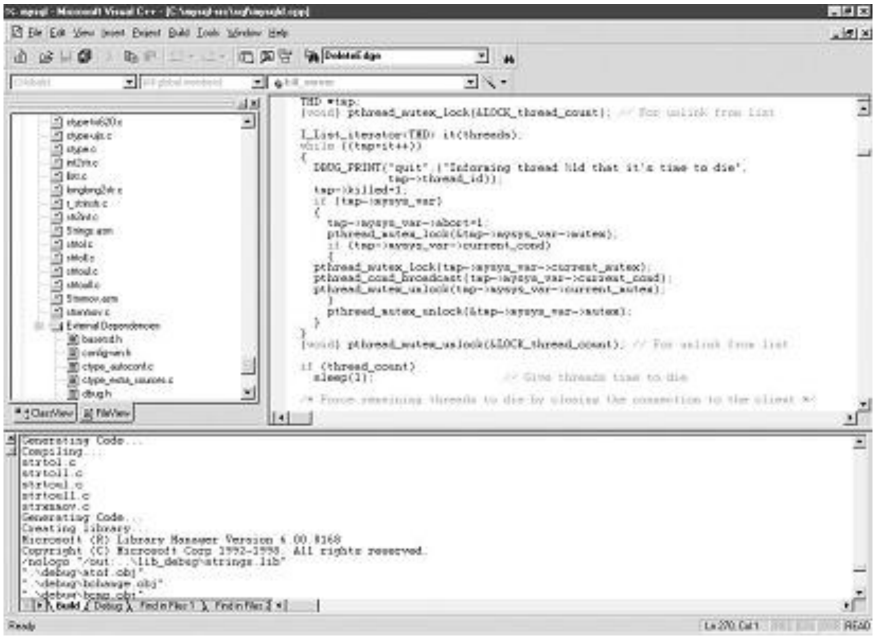
3. Choose Build | Set Active Configuration to obtain a list of available configurations. Select mysqld – Win32 Release (see Figure 20). Click OK.



mysqlclient - Win32 Release
mysqlclient - Win32 Debug
mysqld - Win32 Release
mysqld - Win32 Debug
mysqld - Win32 nt
mysqld - Win32 Max nt
mysqld - Win32 Max
mysqldemb - Win32 Release
mysqldemb - Win32 Debug
mysqldump - Win32 Release
mysqldump - Win32 Debug
mysqlimport - Win32 Release

FIGURE 20 *Selecting which version of MySQL to build in Visual C++*

4. Begin compiling by pressing the F7 key. The various MySQL binaries will be compiled−−expect the process to take from 20 minutes to an hour, depending on the capabilities of your machine. During the compilation process, the Visual C++ compiler window will display a series of messages, such as those shown in Figure 21.



```
THD *tmp;
(void) pthread_mutex_lock(&LOCK_thread_count); // For unlink from list

I_List_iterator<THD> it(threads);
while ((tmp=it++))
{
  DBUG_PRINT("quit",("Informing thread %ld that it's time to die",
                     tmp->thread_id));
  tmp->killed=1;
  if (tmp->mysys_var)
  {
    tmp->mysys_var->abort=1;
    pthread_mutex_lock(&tmp->mysys_var->mutex);
    if (tmp->mysys_var->current_cond)
    {
  pthread_mutex_lock(tmp->mysys_var->current_mutex);
  pthread_cond_broadcast(tmp->mysys_var->current_cond);
  pthread_mutex_unlock(tmp->mysys_var->current_mutex);
    }
    pthread_mutex_unlock(&tmp->mysys_var->mutex);
  }
}
(void) pthread_mutex_unlock(&LOCK_thread_count); // For unlink from list

if (thread_count)
  sleep(1);                       // Give threads time to die

/* Force remaining threads to die by closing the connection to the client */
```

```
Generating Code...
Compiling...
strtol.c
strtoll.c
strtoul.c
strtoull.c
strxmov.c
Generating Code...
Creating library...
Microsoft (R) Library Manager Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
/nologo "/out:..\lib_debug\strings.lib"
".\debug\atof.obj"
".\debug\bchange.obj"
".\debug\bcmp.obj"
```

FIGURE 21 *Compiling MySQL for Windows in Visual C++*

5. After compilation is complete, create a separate installation directory to house the compiled binaries−−for example, `c:\program files\mysql`.

6. Create a `bin\` subdirectory under this directory, and move the compiled libraries and executables into this directory. While you're at it, also move the `data\`, `share\`, `docs\`, and `support−files\` directories from the working directory into this directory.

7. You should now be able to start the MySQL server by diving into the `bin\` subdirectory of your MySQL installation and launching the MySQL server daemon directly

(`mysqld.exe` or `mysqld-nt.exe`).

Once the server has started, proceed to test it as per the instructions in "Testing MySQL," next.

After MySQL has been successfully installed, the base tables have been initialized, and the server has been started, you can verify that all is working as it should via some simple tests.

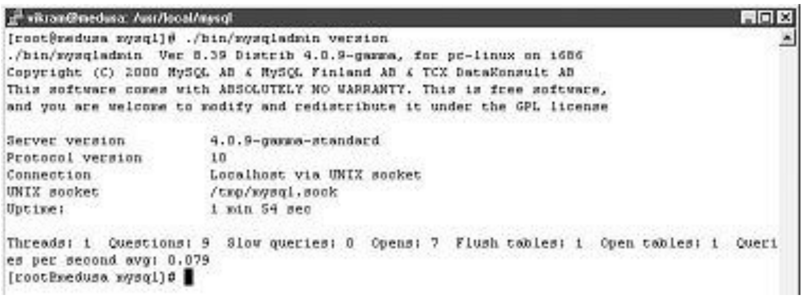
Note that all these commands should be run from your UNIX or Windows command prompt. I am assuming here that you are running them from your MySQL installation directory (as per the examples in the section "Installing and Configuring MySQL," this will be either `/usr/local/mysql` in UNIX or `c:\program files\mysql` in Windows).

Use the mysqladmin Utility to Obtain Server Status

The mysqladmin utility is usually located in the bin subdirectory of your MySQL installation. You can execute it by changing to that directory and executing the following command:

```
[root@host]# mysqladmin version
```

You should see something resembling the output shown in Figure 22.



```
[root@medusa mysql]# ./bin/mysqladmin version
./bin/mysqladmin  Ver 8.39 Distrib 4.0.9-gamma, for pc-linux on i686
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version          4.0.9-gamma-standard
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /tmp/mysql.sock
Uptime:                 1 min 54 sec

Threads: 1  Questions: 9  Slow queries: 0  Opens: 7  Flush tables: 1  Open tables: 1  Queri
es per second avg: 0.079
[root@medusa mysql]# 
```

__FIGURE 22__ The output of a call to mysqladmin

Connect to the Server Using the MySQL Client, and Execute Simple SQL Commands

The MySQL client that ships with the MySQL distribution is named, funnily enough, mysql. Fire it up from your command prompt by switching to the bin directory of your MySQL installation and typing

```
[root@host]# mysql
```

You should be rewarded with a `mysql>` prompt.

At this point, you are connected to the MySQL server and can begin executing SQL commands or queries. Here are a few examples, with their output:

```
mysql> SHOW DATABASES;
+----------+
| Database |
+----------+
| mysql    |
| test     |
+----------+
2 rows in set (0.13 sec)

mysql> USE mysql;

Reading table information for completion of table
and column names
You can turn off this feature to get a quicker
startup with −A
Database changed

mysql> SHOW TABLES;
+-----------------+
| Tables_in_mysql |
+-----------------+
| columns_priv    |
| db              |
| func            |
| host            |
| tables_priv     |
| user            |
+-----------------+
6 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM user;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)
```

Once testing is complete, you should perform two more tasks to complete your MySQL installation:

Alter the MySQL root Password

When MySQL is first installed, access to the database server is restricted to the MySQL administrator, aka root. By default, this user is initialized with a null password, which is generally considered a Bad Thing. You should therefore rectify this as soon as possible by setting a password for this user via the included mysqladmin utility, using the following syntax in UNIX

```
[root@host]# /usr/local/mysql/bin/mysqladmin −u
root password ' <new−password>'
C:\> c:\program files\mysql\bin\mysqladmin −u root
password ' <new−password>'
```

This password change goes into effect immediately, with no requirement to restart the server or flush the privilege table.

NOTE *The MySQL root user is not the same as the system root user on UNIX.*

Configure MySQL to Start Automatically When the System Boots up
On UNIX, MySQL comes with a startup/shutdown script, which is the recommended way of starting and stopping the MySQL database server. This script, named mysql.server, is available in the support−files subdirectory of your MySQL installation, and it can be invoked as follows:

```
[root@host]#
/usr/local/mysql/support−files/mysql.server start
[root@host]#
/usr/local/mysql/support−files/mysql.server stop
```

To have MySQL start automatically at boot time, you simply need to copy this script to the `/etc/init.d/*` directory hierarchy of your system, and then you can invoke it with appropriate parameters from your system’s bootup and shutdown scripts.

To start MySQL automatically on Windows, you can simply add a link to the mysqld server binary to your Startup group. For more information, please refer to Chapter 13.

With a Little Help from My Friends…

In case you have problems starting the MySQL server, you can obtain fairly detailed information on what went wrong by looking at the MySQL error log. Most often, this log can be found in the var subdirectory of your MySQL installation, and it is named hostname.err. Other common problems, such as a forgotten superuser password or incorrect path settings, can also be discovered and resolved via a close study of this error log. You can also visit the following resources for advice on how to resolve problems you may encounter during the installation process:

- ***The MySQL manual*** http://www.mysql.com/documentation
- ***The MySQL mailing lists*** http://lists.mysql.com
- ***Google*** http://www.google.com
- ***Google Groups*** http://groups.google.com

If you’re reporting a problem or a bug, remember to use the supplied mysqlbug script to gather necessary system information and include it in your report.

Summary

As a popular open−source application, MySQL is available for a wide variety of platforms and architectures, in both binary and source form. This chapter explained the distinction among the different versions of MySQL, together with recommendations on the most appropriate version for your requirements; it also demonstrated the process of installing MySQL on the two most common platforms, Linux and Windows. It provided installation and configuration instructions for both binary and source distributions and also provided pointers to online resources for other platforms and for detailed troubleshooting advice and assistance.

Remember: this is chapter three of *MySQL: The Complete Reference*, by Vikram Vaswani (McGraw−Hill/Osborne, ISBN 0−07−222477−0, 2004). Vikram is the founder of Melonfire, and has had numerous articles featured on Dev Shed.
Buy this book now.