

# 2

## Installing MySQL

Regardless of how you plan to use MySQL — whether to access data from within a data-driven application or to build databases that support data-driven applications — there will no doubt come a time when you want to install MySQL yourself. You might choose to do this in order to develop and test your own applications, or you might be the one in your organization responsible for implementing MySQL in a production environment. And it certainly wouldn't be unheard of if you found yourself having to do both.

Fortunately for everyone, MySQL is a relatively easy application to install, and it can be installed on a number of platforms. For example, you can install MySQL on a computer running FreeBSD or on an Apple computer running Mac OS X. The possibilities are numerous. This chapter, though, focuses only on how to install MySQL on computers running Linux or Windows, although much of what you learn can translate to other operating systems.

It should take you little preparation to install MySQL on Linux or Windows. To do so, you must make several preliminary decisions on exactly what you want to install, and then you must download the necessary files. From there, you can install MySQL on your selected platform, which is explained later in this chapter. Specifically, this chapter covers the following topics:

- ❑ What steps you must take before installing MySQL. This includes making preliminary decisions about the MySQL version, platform, distribution type, and edition.
- ❑ Step-by-step instructions on how to install MySQL on Linux and Windows. The instructions also include details about how to start the MySQL server in both environments.
- ❑ How to test your Linux and Windows installations and verify that the `mysql` administrative database has been properly initialized.

### Getting Started

Two of the features that make MySQL such a valuable relational database management system (RDBMS) are its ability to be implemented on multiple platforms and its different versions, editions, and distribution types. Before you can actually install MySQL, you must make several preliminary decisions about the installation, and then, based on these decisions, you must download the appropriate distribution files.

### ***Making the Preliminary Decisions***

From its inception, MySQL AB has aimed to provide an RDBMS that users can implement in different ways and on different platforms. As a result, whenever you're preparing to install MySQL, you must first decide which version to install, on which platform to install it, which distribution type to select, and which edition to use.

### **Selecting a Version of MySQL**

MySQL is a constantly evolving RDBMS. You can download the latest stable release of the product, or you can download a version that is in some stage of development. As a result, your choice of versions depends, in part, on weighing the need for stability against the need for features that might be available only in a version currently in development.

Each version of MySQL available for download is released in one of the following stages:

- ❑ **Alpha:** A version of MySQL in this stage has not been fully tested. In addition, major code changes could still occur, and major new features could be added.
- ❑ **Beta:** This is the next stage after alpha. At the beta stage, all code has been tested and no major new features will be added. A version of MySQL in the beta stage is considered fairly stable.
- ❑ **Gamma:** Once a version of MySQL has been in the beta stage for a while and the product appears to be stable and operating without significant problems, it is promoted to the gamma stage. At this point, only minor fixes are applied to the product.
- ❑ **Production:** This version of the product is considered very stable. It has been running at different sites with no significant problems. At this stage, only critical bug fixes are applied.

All releases, regardless of which stage they are in, are thoroughly tested to ensure that they're safe to use. Unless there are features that you specifically need to implement that are available only in a non-production release, your best bet is to go with the latest production release available to ensure that you have the most stable product.

### **Selecting a Platform to Run MySQL**

Once you determine which version of MySQL that you want to install, you must then decide on the platform on which to run MySQL. For many of you, the platform is decided by default (it's the only type of computer that you have to work on), and you're simply looking for the appropriate copy of MySQL to run on your existing computer. Some of you, however, can select the platform; in that case, you have a number of options from which to choose.

You can install MySQL on a variety of operating systems and computer architectures. The following list provides you with some of the options available for running MySQL:

- ❑ **Linux:** You can run MySQL on different versions of Linux running on multiple types of computer architectures, including x86, S/390, IA64, Alpha, PPC, and AMD64.
- ❑ **Windows:** You can run MySQL on different versions of Windows (including Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP, and Windows 2003) on x86 computers.
- ❑ **Solaris:** You can run MySQL on Solaris 8 and 9 on multiple types of computer architectures, including 32-bit SPARC, 64-bit SPARC, and 32-bit x86.

- ❑ **FreeBSD:** You can run MySQL on FreeBSD on multiple types of computer architectures, including x86 and LinuxThreads.
- ❑ **Mac OS X:** You can run MySQL on Mac OS X 10.2 and 10.3.

The platforms listed here do not represent a complete list but merely provide an overview of the options available to you. Keep in mind, however, that the exact platforms supported (in terms of operating systems and computer architectures) can vary from version to version of MySQL, so be sure to review the MySQL Web site ([www.mysql.com](http://www.mysql.com)) for platform-specific information about the specific version of MySQL that you want to install.

*You should note, however, that although supported platforms change, the core platforms remain the same.*

In general, you can install MySQL on most Unix-like operating systems. MySQL tends to run more efficiently on some operating systems than others. According to the MySQL product documentation (which you can view on the Web site), MySQL is developed primarily on the SuSE and Red Hat versions of Linux, FreeBSD, and Sun Solaris 8 and 9, and not surprisingly, these platforms have proven to be some of the best platforms on which to run Linux.

In the end, your choice of platforms might be somewhat limited because of the environment in which you work or because you simply do not want to mess with upgrading and changing platforms until you spend time working with MySQL. For the purposes of simply learning how to use MySQL and how to develop MySQL applications that connect to MySQL databases, the choice of platforms is not as critical at this time, although it could become a critical factor in the future. In any case, be sure always to check the most current platform-related information available on the MySQL Web site whenever you're preparing to implement MySQL.

## Selecting a MySQL Distribution Type

Once you determine which version of MySQL to install and on which platform to install it, you must select the distribution type to use for your installation. The *distribution type* refers to the format in which MySQL is available for download and installation. MySQL supports two distribution types:

- ❑ **Binaries:** Binaries are installation packages comprising precompiled files that allow you to install MySQL without having to work with or manipulate the source code. Binaries are provided for a number of operating systems (such as Linux, Solaris, and FreeBSD) and are packaged in compressed tar archive files. A *tar* file is a type of archive that stores files. For some platforms, the binary files are available in native format. For example, MySQL AB supplies RPM Package Manager (RPM) files for Linux installations, .dmg package installers for Mac OS X installations, and zipped files for Windows installations.
- ❑ **Source code:** As the name suggests, the source code distribution type refers to those files that contain the source code for MySQL. The code has not been compiled, so it is up to you to compile it prior to installation. The files that hold the source code come in three formats: compressed archive files for Unix-like machines, RPM files for Linux, and zipped files for Windows.

MySQL AB recommends that you use binary files for your installation, if they exist for your platform. They're generally easier to use than working directly with the source code, and they're an excellent way for beginning MySQL users to get acquainted with MySQL and the installation process. If an RPM binary file and a tar binary file are available for a Linux installation, you're usually better off using the RPM file because it makes MySQL installation much easier.

Some circumstances may arise in which you prefer using the source code. For example, you might want to configure MySQL with extra features or remove features that are inherent in MySQL. Or perhaps a MySQL binary isn't available for the platform you're using. Or maybe you simply prefer the control that compiling your own code provides. Whatever your reason, using a source code distribution is a viable alternative, but if using the source code isn't necessary, you should, at least at first, use binary files for your installation.

*A discussion of using source code to install MySQL is beyond the scope of this book. If you want to learn more about working directly with the source code, be sure to visit the MySQL Web site at [www.mysql.com](http://www.mysql.com).*

### Selecting an Edition of MySQL

In addition to determining the distribution type, you must determine which edition of MySQL you plan to install. MySQL AB provides binary files for the following editions of MySQL:

- ❑ **Standard:** This is the basic edition of MySQL and the edition that you're most likely to install. It includes all the fully tested functionality available in MySQL.
- ❑ **Max:** The Max edition includes all the features available in the Standard edition plus features that most users do not require or that have not been fully tested. In addition, the Max edition includes the Berkeley DB storage engine.
- ❑ **Debug:** The Debug binaries are compiled with additional debug data, which means that they can affect performance. These binaries are not intended for use in a production environment.

*Binary files for the Debug edition for Linux are not available as RPM files or zipped Windows files. In addition, if you use RPM files to install MySQL-Max on Linux, you must first install the Standard RPM and then install the Max RPM. Also, the zipped Windows file includes all editions in one file. After you install MySQL, you determine which edition to use.*

For most users, including beginning MySQL users, the Standard edition provides you with all the features that you need to learn about MySQL and to create applications that connect to MySQL.

### Downloading Distribution Files

Once you determine which version of MySQL to install, on which platform to install it, which distribution type to select, and which edition to use, you're ready to download the necessary files. To download the files, follow these steps:

1. Go to <http://dev.mysql.com/downloads>.
2. Click the link to the version of the MySQL database server and clients that you want to install.
3. From the list of files, locate the applicable platform, distribution type, and edition.
4. Click the [Pick a mirror](#) link for the file you want to download, and then select a mirror. (A *mirror* is a Web site that hosts the files that are available for download. MySQL downloads are normally available at numerous mirror sites.)
5. Download the file.

For most distribution types, you need to download only one file. For Linux RPM files, you must download a file for each component of MySQL that you want to install. At the very least, you should download the server RPM file and the client program's RPM file. Later in this chapter you learn more about installing MySQL from RPM files.

When you're downloading files from the MySQL Web site, notice that the files follow a specific naming convention. For the basic binary files, the files use the following format:

```
mysql-<edition>-<version>-<platform>.tar.gz
```

For example, the following filename identifies the distribution file for the Max edition of MySQL version 4.1.5-gamma:

```
mysql-max-4.1.5-gamma-pc-linux-i686.tar.gz
```

In this case, the platform is Linux running on an x86 computer. If you were going to download the RPM file for the same platform (as well as edition and version), the filename would be as follows:

```
MySQL-Max-4.1.5-0.i386.rpm
```

Although the RPM file indicates that this is the Max edition, because it is an RPM file, Max refers only to the Max upgrade for the server component of MySQL. If this were for the Standard version of the server, the name of the file would be as follows:

```
MySQL-server-4.1.5-0.i386.rpm
```

Windows installation files follow a slightly different convention than Linux installation files. If you want to use the Microsoft Windows Installer to install the full MySQL program, you would use a file similar to the following:

```
mysql-4.1.5-gamma-win.zip
```

The necessary setup file is compressed in a zip file and installs MySQL 4.1.5-gamma. If you want to install a more basic MySQL package, you can use the following file:

```
mysql-4.1.5-gamma-essential-win.zip
```

Notice that this is a Windows Installer (.msi) file and that it includes the word "essential" in the filename, indicating that this is the file to use to install the MySQL Essential package, a package that is similar to the regular version, except that it includes fewer features. In most cases, though, the Essential package provides all the necessary functionality you need to get started using MySQL, and it is the package that MySQL recommends for your basic installation. The examples shown in this book are based on the Essential package.

From the filenames, you should be able to determine all the information you need about which file you downloaded. This is handy in case you download multiple versions, editions, or distribution types and need to be able to easily distinguish among them. Once you've downloaded the files that you need, you're ready to install MySQL.

*MySQL recommends that, when you download a file, you verify the integrity of that file. They suggest that you use the MD5 checksums and GnuPG signatures to verify file integrity. For more information about how to verify files, go to [www.mysql.com](http://www.mysql.com).*

# Installing MySQL

As explained, you can install MySQL on a variety of platforms; however, it isn't possible to discuss the installation process for every one of these platforms. As a result, this section focuses on the following three scenarios:

- ❑ Using RPM files to install MySQL on Linux
- ❑ Using a tar file to install MySQL on Linux
- ❑ Using a zipped file to install MySQL on Windows

You have several reasons for choosing Linux and Windows. First, Linux, like MySQL, is an open-source application, which means that you can implement MySQL in a completely open-source environment and take advantage of the flexibility and cost savings that this provides. In addition, MySQL was developed primarily on Linux, which means that the Linux version of MySQL is the most tested and widely implemented version. In addition, Linux is similar to other Unix-like operating systems, so installation and configuration are similar among products.

To round out the choice of Linux, this section also describes how to install MySQL on Windows. The advantage of installing MySQL on Windows is that Windows has a wide user base, which has translated to an increasing number of users implementing MySQL on Windows. Knowing how to install MySQL on Windows is steadily becoming a more important consideration when implementing databases and data-driven applications.

For both Linux and Windows, you're running the Standard edition of MySQL on an x86 computer. Consequently, if you plan to follow the installation steps in this chapter, you must download the appropriate file or files for your specific platform. For example, if you're using RPM files to install MySQL on Linux, you should download the following two files, along with any additional files you want to include:

```
MySQL-server-<version>.i386.rpm  
MySQL-client-<version>.i386.rpm
```

The `<version>` placeholder refers to the version number and, if applicable, the release cycle, such as alpha, beta, or gamma. If you plan to use a tar file to install MySQL on Linux, you must download the following file:

```
mysql-standard-<version>-pc-linux-i686.tar.gz
```

Finally, for those of you who want to install MySQL on Windows, you must download the following file:

```
mysql-<version>-win.zip
```

Notice that, although the platforms (Linux and Windows on an x86 computer), the edition (Standard), and the distribution types (RPM, tar, and zipped files) are selected, only placeholders designate the version. This is because you should download the most current production version of MySQL. The installation process for various versions of 4.0 and 4.1 is consistent across versions.

*The exercises and examples in this book employ the latest version of MySQL 4.1. You may find that most of the code samples work in version 4.0 as well.*

If you're unable to implement MySQL on one of the two platforms described here, you can try to install MySQL from one of the binary files provided for other platforms. You should find the installation process on any Unix-like system to be similar to using tar files to install MySQL on Linux.

Once you download the necessary installation file or files, you can go to the appropriate section in this chapter for details about installing MySQL on Linux or Windows.

## Using RPM Files to Install MySQL on Linux

Of all the methods available for installing MySQL, using RPM files is by far the easiest. Basically, you need to copy the files to a directory on your computer and then unpack the files. The following steps explain how to copy and unpack the files:

1. Log on to Linux as the root user. If you're logged on as another user, type the following command at the command prompt and press Enter to execute the command:

```
su - root
```

The `su` command, which stands for *substitute user*, allows you to run Linux commands as the specified user.

2. The next step is to copy the RPM files to the `/tmp` directory or whichever directory you want to use. For the purposes of explaining the installation process, this chapter assumes that you're using the `/tmp` directory. If you're using another directory, simply replace `/tmp` with that directory path whenever the instructions reference `/tmp`. Before you actually copy the files, you might have to mount the drive where they are located, as could be the case with a CD-ROM drive. For example, if you plan to copy the files from a CD, insert the CD into the drive, type the following command, and then press Enter:

```
mount /mnt/cdrom
```

The `mount` command mounts the CD-ROM drive. You can then access the drive in the same way you would a directory.

3. Now you can copy the RPM file for MySQL server to the `/tmp` directory. For example, if the file is located in the Linux folder on the CD that you just mounted, you would use the following command:

```
cp /mnt/cdrom/linux/mysql-server-<version>.i386.rpm /tmp
```

The `cp` command allows you to copy files from one directory to another. Executing the command shown here copies the specified file to the `/tmp` directory.

*When you use this method to copy a file from a CD to a Linux directory, the file is saved exactly as the `cp` command specifies. For example, if the filename on the CD uses uppercase, as in MySQL, the capitalization is lost if the filename in the `cp` command is all lowercase. This is important when working in a Linux environment because Linux is case sensitive; you must use the exact case when working with Linux directories and files. As a result, you should stay aware of how you save files to a Linux directory. This book uses all lowercase here to make it easier to work with the files later.*

4. Your next step is to copy the RPM file for the MySQL client utilities to the `/tmp` directory:

```
cp /mnt/cdrom/linux/mysql-client-<version>.i386.rpm /tmp
```

As you can see, the only difference between this command and the previous command is that `server` replaces `client`.

5. After you copy the files to the `/tmp` directory, you might have to unmount your drive. Again, using the example of the CD-ROM drive, you would unmount the drive by using the following command:

```
umount /mnt/cdrom
```

Once you unmount the drive, you can remove the CD.

6. Now that you have copied the necessary RPM files to the `/tmp` directory, you're ready to install MySQL. The next step, then, is to change to the `/tmp` directory (or to the directory where you copied the files):

```
cd /tmp
```

The `cd` command, which stands for *change directory*, moves you to the directory that you specify, and your command prompt should reflect your directory change.

7. Before you actually run the installation, it's a good idea to make sure that the RPM files were copied to the directory correctly by entering the following command:

```
ls
```

The `ls` command allows you to view the contents of the active directory. When you run this command, you should see the two RPM files that you copied over.

8. Once you confirm the presence of the RPM files, you're ready to install the server and client utilities. Begin with the server. Enter the following command at the command prompt and press Enter:

```
rpm -i mysql-server-<version>.i386.rpm
```

The `rpm` utility allows you to work with RPM package files. The `-i` switch indicates that you want to install the specified package. When you execute this command, a number of events occur:

- The `mysql` user account and `mysql` group account are created in Linux.
- The MySQL files are extracted from the RPM package and copied to their appropriate directories.
- The `mysql` administrative database initializes.
- Ownership of MySQL-related directories and files is altered.
- The MySQL server starts.
- The MySQL server is set up to start automatically when you start up Linux.

You can get a glimpse of what's happening with the RPM extraction process in the command window as it is printing out. Figure 2-1 gives you an idea of what you might expect to see as you are installing MySQL. After the installation is complete, you're returned to the command prompt.



```

root@Server03:/tmp
File Edit View Terminal Go Help
[root@Server03 tmp]# rpm -i MySQL-server-4.1.3-0.i386.rpm
warning: MySQL-server-4.1.3-0.i386.rpm: V3 DSA signature: NOKEY, key ID 5072e1f5
040711 13:14:54 Warning: Asked for 196608 thread stack, but got 126976
040711 13:14:55 Warning: Asked for 196608 thread stack, but got 126976
PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h Server03 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
[root@Server03 tmp]#

```

Figure 2-1

9. Once the server is installed, you're ready to install the client utilities:

```
rpm -i mysql-client-<version>.i386.rpm
```

The command installs the client utilities on your system.

At this point, the MySQL server and client utilities are installed. You're ready to test your installation. (See the "Testing Your MySQL Installation" section later in the chapter.)

## Using a Tar File to Install MySQL on Linux

As you did with RPM files, you can use a tar file to install MySQL on Linux. The process, though, is not nearly as simple as it is with the RPM approach. Still, you might find that you're working on a version of Linux that doesn't support RPM installations or that you want to have more control over the installation process.

To use a tar file to install MySQL, you need to follow five specific steps:

1. Create a logon account and a group account.
2. Copy the tar files to your computer.
3. Unpack the distribution file.
4. Initialize the installation.
5. Start the MySQL server.

Each of these steps is described in detail in the text that follows. They're broken down into subsections so you're clear on the purpose of each task.

## Creating the User and Group Accounts

The first step that you need to take in preparing for a tar-based installation is to create the necessary Linux user and group accounts. For the purposes of this book, create a user account named `mysql` and a group account named `mysql`. You could use different names for the account; however, because the RPM

installation creates accounts with these names, this book uses the same names as well, just to be consistent and to avoid confusion later in the book. To create the two Linux accounts, follow these steps:

1. Log on to Linux as the root user. If you're logged on as another user, type the following command at the command prompt and press Enter to execute the command:

```
su - root
```

The `su` command, which stands for *substitute user*, allows you to run Linux commands as a substitute user, which in this case is the root user.

2. The next step is to create the group account:

```
groupadd mysql
```

The `groupadd` utility creates a group account that uses the specified name, which in this case is `mysql`.

3. Next, you must create the user account:

```
useradd -g mysql mysql
```

The `useradd` utility creates a user named `mysql` in a group named `mysql`. The `-g` switch indicates that the new user account should be added to the group account that follows the switch.

That's all there is to creating the necessary accounts. You can now move on to the next phase of the installation process.

## Copying the Tar File

The next step in the MySQL installation process is to copy the tar file to the `/usr/local` directory or whichever directory you want to use. If you downloaded the file from the MySQL Web site into a directory on your computer, you can copy the file from the location where you copied it to the target directory. For the purposes of explaining the installation process, this book assumes that you're using the `/usr/local` directory. If you're using another directory, simply replace `/usr/local` with that directory path whenever instructions reference `/usr/local`.

The following steps explain how to copy the tar file:

1. Before you actually copy the file, you might have to mount the drive where the file is located, as could be the case with a CD-ROM drive. For example, if you plan to copy the files from a CD, insert the CD into the drive, type the following command, and then press Enter:

```
mount /mnt/cdrom
```

The `mount` command mounts the CD-ROM drive. You can then access the drive in the same way you would a directory.

2. Now you can copy the tar file to the `/usr/local` directory. For example, if the file is located in the Linux folder on the CD that you just mounted, you would use the following command:

```
cp /mnt/cdrom/linux/mysql-standard-<version>-pc-linux-i686.tar.gz /usr/local
```

The `cp` command allows you to copy files from one directory to another. When you execute the command shown here, the specified file is copied to the `/usr/local` directory.

When you use this method to copy a file from a CD to a Linux directory, the file is saved exactly as it is specified in the `cp` command. For example, if the filename on the CD uses uppercase, as in MySQL, the capitalization is lost if the filename in the `cp` command is all lowercase. This is important when working in a Linux environment because Linux is case sensitive; you must use the exact case when working with Linux directories and files. As a result, you should stay aware of how you save files to a Linux directory. These instructions use all lowercase here to make it easier to work with the files later.

3. After copying the file to the `/usr/local` directory, you might have to unmount your drive. Again, using the example of the CD-ROM drive, you would unmount the drive by using the following command:

```
umount /mnt/cdrom
```

Once you've unmounted the drive, you can remove the CD.

Once the file has been copied, you're ready to unpack the tar file.

## Unpacking the Distribution File

Now that the necessary tar file is copied to the `/usr/local` directory, you're about ready to install MySQL. To do that, you must unpack the tar archive file. Before you unpack the file, you should change to the directory where the file is located and view the contents of that directory to be sure that the file was properly copied. Then you can extract the MySQL files, as shown in the following steps:

1. Execute the following command:

```
cd /usr/local
```

The `cd` command, which stands for *change directory*, moves you to the directory that you specify. Your command prompt should now reflect your directory change.

2. Before you actually run the installation, it's a good idea make sure that the file was copied over by entering the following command:

```
ls
```

The `ls` command allows you to view the contents of the active directory. When you run this command, you should see the tar file that you copied over.

3. Once you confirm the presence of the tar file, you're ready to install MySQL. Type the following command in the command prompt and press Enter:

```
tar xzvf mysql-standard-<version>-pc-linux-i686.tar.gz
```

The tar utility is an archiving program that allows you to store and extract files. The `x` switch indicates that files should be extracted from an archive. The `z` switch indicates the archive should be filtered through `gzip`, which is a file compression utility. The `v` switch indicates that files should be listed as they're being extracted, and the `f` switch specifies the use of an archive file. The name of the archive file is then specified. Figure 2-2 gives you an idea of what you might expect to see when you run the tar utility against this file.

After the extraction process completes, you're returned to the command prompt.

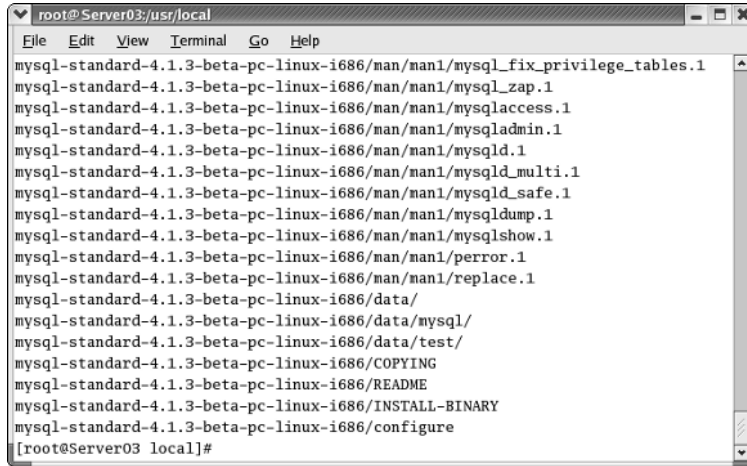


Figure 2-2

4. When you use the tar utility to extract a file, a directory is created in the current directory where you're located. The directory shares the same name as the version of the product that you're installing. For example, a directory named `/usr/local/mysql-standard-<version>-pc-linux-i686` should have been created when you ran the tar utility in the previous step. As you can see, referring to this directory by name can translate to a lot of typing (and most likely a lot of errors). The way around this is to assign an alias to that directory, as shown in the following step:

```
ln -s mysql-standard-<version>-pc-linux-i686 mysql
```

The `ln` command-line utility creates a link to the specified directory. The `-s` switch specifies that this is a symbolic link rather than a hard link. You specify the name of the link (`mysql`) after the actual directory name. Once you run this command, you can refer to the directory as `/usr/local/mysql`.

5. As one final check in this process, you can view the contents of the `/usr/local/` directory to see for yourself the directory that has been created.

```
ls -al
```

The `-al` switch indicates that the `ls` utility should list all directory content in a long format. You should see a directory named `mysql-standard-<version>-pc-linux-i686` and one named `mysql`. The `mysql` directory should contain a pointer (`->`) to the `mysql-standard-<version>-pc-linux-i686` directory.

Once you verify the contents of the `/usr/local` directory, you're ready to move on to the next step.

## Initializing the Installation

After extracting the files from the tar package, you must initialize the mysql database that is created as part of the installation process. The MySQL files that you just extracted include a script named `mysql_install_db`, which is included in the `/usr/local/mysql/scripts` directory. After you run the script, you must change ownership of a couple of directories to complete the MySQL installation.

The following steps allow you to initiate your MySQL installation:

1. First, change into the new directory that you created (`/usr/local/mysql`) using the following command:

```
cd mysql
```

Your command prompt should reflect that you're in the `mysql` directory

2. Next, you must run the `mysql_install_db` script by executing the following command:

```
scripts/mysql_install_db --user=mysql
```

The script works behind the scenes, and you don't need to do anything but wait. Once it completes, you're returned to the command prompt.

3. The next step is to change ownership of the current directory and its subdirectories to the root user:

```
chown -R root .
```

The `chown` utility allows you to change the ownership of a particular file or directory. The `-R` switch specifies a *recursive* change, which means that it affects any subdirectories and files in that directory. The period indicates that the change applies to the current directory (`mysql`).

4. Next, you must change the ownership of the data directory to the `mysql` user account that you created earlier:

```
chown -R mysql data
```

This time, the `chown` command references a specific directory rather than using a period. As a result, the change applies only to the data directory and the files and directories that it contains. (The data directory contains the actual database files.)

5. Finally, you must change the group ownership of the `mysql` directory to the `mysql` group:

```
chgrp -R mysql .
```

The `chgrp` utility, like the `chown` utility, changes the ownership of a file or directory, except that it applies to a group, not a user. In this case, it grants the `mysql` group ownership on the current directory (`mysql`).

This completes the MySQL installation process. Before you can start working with the MySQL server, however, you must start the server.

## Starting the MySQL Server

MySQL supports two methods for starting the MySQL server: starting the server manually and starting the server automatically. In either case, you must launch the `mysqld` program (or some version of that program) from a command prompt or through a script. The `mysqld` program is the MySQL server that that supports data storage and management and allows you to access the databases and their data.

### Starting the Server Manually

When you first install MySQL, you probably want to try it out. Before you can do that, you must start the server, which is actually a very straightforward process. To start MySQL server, take the following steps:

1. The following command launches the `mysqld` program in safe mode. If you're already in the `/usr/local/mysql` directory, you do not need to enter that part of the path. Type the following command and press Enter:

```
/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

Starting `mysqld` in safe mode adds safety features such as logging errors and restarting the server if necessary. Normally, using safe mode is the recommended way to start the MySQL server. The `--user=mysql` option instructs MySQL to run under the `mysql` user account. The ampersand (&) at the end of the command is specific to Linux and indicates that the `mysqld_safe` script should run as a process in the background to support the script's monitoring capabilities.

2. If, after you execute the `mysqld_safe` command, you're not returned to the command prompt right away, press Enter to display the prompt.

That's all there is to starting the server. There might be times when you want to stop the server after you've started it. To stop the MySQL server, execute the following command:

```
/usr/local/mysql/bin/mysqladmin shutdown
```

This command uses the `mysqladmin` utility, along with the `shutdown` option, to stop the server. Once the server is stopped, you can simply use the `mysqld_safe` command to start it back up again.

### Starting the Server Automatically

Although it's useful to know how to start and stop the MySQL server manually, you may find that, in most cases, you want the server to start automatically when you start up Linux. To do this, you must copy the `mysql.server` script to the proper directory, which, in this case, is `/etc/init.d`. The `mysql.server` script, located in `/usr/local/mysql/support-files`, provides the necessary commands to start MySQL automatically.

To set up the MySQL server to start automatically, you must take the following steps:

1. First, copy the `mysql.server` script to the appropriate directory. As you're copying the script, assign it the name `mysql`, as shown in the following command:

```
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql
```

The command uses the `cp` utility to copy the `mysql-server` script to the `/etc/init.d` directory. At the same time, the name `mysql` is assigned to the script.

2. Change the permissions for the `mysql` script:

```
chmod +x /etc/init.d/mysql
```

The command uses the `chmod` utility to change the permissions of the specified file. The `+x` switch indicates that execute permission will be added to the file.

3. You must next activate the script. In Linux, to activate a script, use the following command:

```
chkconfig --add mysql
```

The `chkconfig` utility allows you to maintain configuration information. By using the utility with the `--add mysql` option, you're ensuring that the `mysql` service has a start entry in every run level. (For more information about startup scripts and the `chkconfig` utility, see the Linux documentation.)

Once you set up the `mysql` service to run automatically and start the MySQL server, you're ready to test your installation. (See the "Testing Your MySQL Installation" section later in the chapter.)

## Installing MySQL on Windows

The process for installing MySQL on Windows is relatively simple. Basically, you run the installation program and then run the MySQL Server Instance Configuration Wizard. You should find that, for the most part, the MySQL installation process is the same from one Windows operating system to the next. In addition, MySQL should operate in a fairly similar manner on the different systems. If you're running MySQL on Windows NT, Windows 2000, Windows XP, or Windows Server 2003, you can run the MySQL server as a service. These four operating systems also support named pipe and TCP/IP connections. In contrast to this, Windows 95, Windows 98, and Windows Me do not allow you to run MySQL as a service, and they do not support named pipe connections, only TCP/IP connections.

*The process for installing MySQL on Windows XP and Windows Server 2003 is the same for both operating systems. Basic installation should be consistent on most Windows operating systems, although you might discover minor differences. Be sure to check out the MySQL product documentation for more information about installing MySQL on different versions of Windows.*

## Running the Installation Program

The installation process is essentially the same whether you install MySQL from a `.zip` file or a `.msi` file. The only difference is in how you begin the installation process. Once you start that process, the steps are the same. For the purposes of this book, the `.msi` file is used to install an Essential MySQL package. The steps that follow also apply to installing MySQL from a `.zip` file. The following steps describe how to install MySQL on Windows:

1. Log on to the Windows computer with an account that has administrative privileges.
2. Open Windows Explorer, and locate the `.zip` file or `.msi` file that you downloaded from the MySQL Web site. To open Windows Explorer, right-click the Start button, and click Explore.

If you're installing MySQL from a `.zip` file, you must extract the `Setup.exe` file from the `.zip` file. In Windows Server 2003 and Windows XP, you can extract the `Setup.exe` file simply by double-clicking the `.zip` file. A Windows Explorer window then opens and displays the `Setup.exe` file. From there, double-click the `Setup.exe` file to start the MySQL installation process. When the installation process begins, a MySQL Setup message box temporarily appears, indicating that the installation files are being extracted. Then a Windows Installer message box appears temporarily, indicating that your system is preparing to install MySQL.

*If you have an extraction utility such as WinZip installed on your system, you can use that to extract the files. If you're uncertain how to use the utility, check the utility's product documentation for details.*

If you're installing MySQL from a `.msi` file, double-click the file to start the MySQL installation process. When the installation process begins, a Windows Installer message box appears temporarily, indicating that your system is preparing to install MySQL.

Once any messages boxes have cleared, the first screen of the Setup Wizard appears, as shown in Figure 2-3.



Figure 2-3

3. Click Next. The Setup Type screen appears (shown in Figure 2-4) and provides you with several options that describe the types of setups that you can perform.

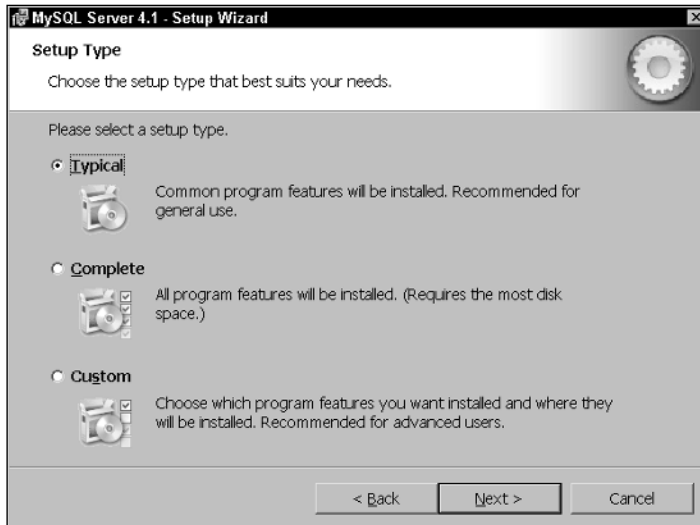


Figure 2-4

In most cases, you should choose the Typical installation. This normally provides you with all the functionality that you need at this time. As you become more familiar with MySQL, or if you're simply the type who likes to know exactly what you're installing, select the Custom



option and choose which components to install. The examples and exercises in this book use a Typical installation.

4. Click Next. The Ready to Install the Program screen appears (shown in Figure 2-5) and provides the details of your installation.

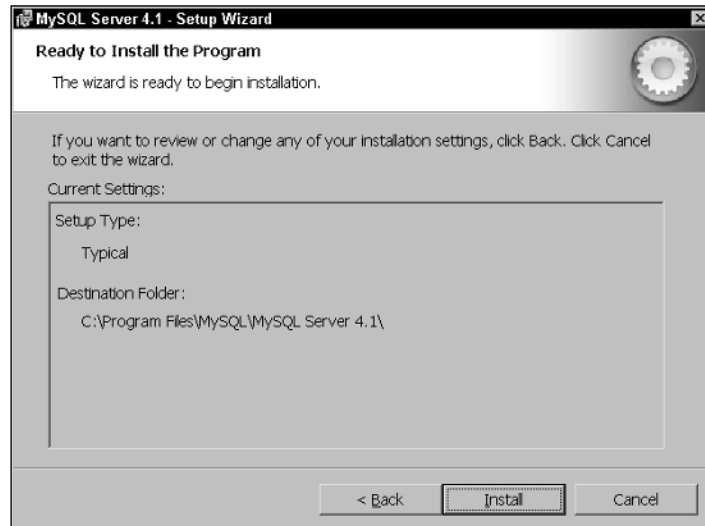


Figure 2-5

Be sure to review the information on the screen. If you want to change any of the settings, click the Back button to return to the appropriate screen.

5. Click Install to begin the actual installation. The Installing MySQL Server screen appears and provides you with the progress of your installation, as shown in Figure 2-6.

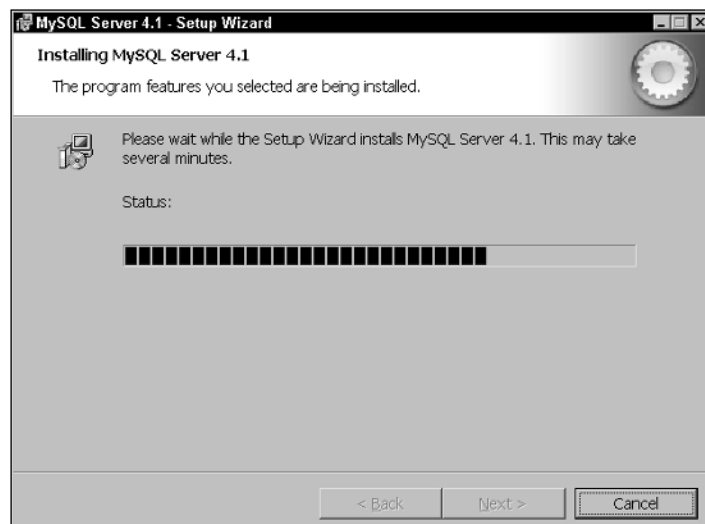


Figure 2-6

After the main part of the installation process is complete, the MySQL.com Sign-Up screen appears, as shown in Figure 2-7, providing options for creating a MySQL.com account.



Figure 2-7

For now, skip the sign-up process and continue with the installation; however, you can create an account at a later time.

6. Select the Skip Sign-Up option, and click Next. The Wizard Completed screen appears, as shown in Figure 2-8.



Figure 2-8

The Wizard Completed screen includes the option Configure the MySQL Server now. This option is selected by default. As a result, when you close the Setup Wizard, the MySQL Server Instance Configuration Wizard is launched automatically, allowing you to set up your system's initial MySQL configuration.

7. Ensure that the Configure the MySQL Server now option is selected, and click Finish.

The MySQL installation is now complete, and the first screen of the MySQL Server Instance Configuration screen appears.

## Configuring the MySQL Server

Once you complete the MySQL installation, you can run the MySQL Server Instance Configuration Wizard to set up your system. The wizard allows you to specify the type of configuration, install MySQL as a service, and set up initial security settings. If you selected the Configure the MySQL Server now option in the last screen of the MySQL Setup Wizard, the MySQL Server Configuration Wizard launches automatically. You can also launch the wizard from the Windows Start menu (as part of the MySQL program group).

Once you launch the MySQL Server Instance Configuration Wizard, you can follow these steps to set up the MySQL server:

1. Verify that the MySQL Server Instance Configuration Wizard has been launched. The first screen of the wizard is shown in Figure 2-9.



Figure 2-9

2. Click Next. The next screen of the wizard appears (shown in Figure 2-10), allowing you to choose a configuration type.

The MySQL Server Instance Configuration Wizard allows you to select the Detailed Configuration option or the Standard Configuration option. Detailed Configuration steps you through a process for identifying configuration settings that are specific to your environment. For example, you can specify whether you're working on a developer computer, a server computer, or a dedicated MySQL computer. You can also specify such configuration settings as the primary types

of databases that will be supported, the number of concurrent users, and the port setting. If you select the Standard Configuration option, the MySQL server is set up with a general-purpose configuration. For the purposes of this book, you use the standard configuration.

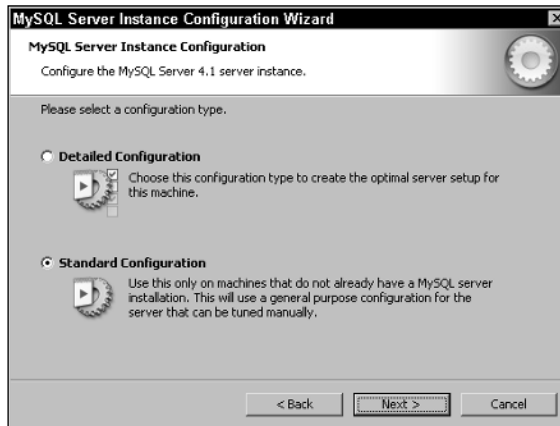


Figure 2-10

3. Ensure that the Standard Configuration option is selected; then click Next. The next screen of the wizard appears, as shown in Figure 2-11.



Figure 2-11

The screen allows you to specify whether to set up MySQL as a service. If you're running MySQL on Windows NT, Windows 2000, Windows XP, or Windows Server 2003, you can run the MySQL server as a service, which is the recommended method to use. By default, the Install As Windows Service option is selected, MySQL is selected in the Service Name drop-down list, and the Launch the MySQL Server automatically option is selected. By accepting the default options, MySQL runs automatically whenever you start your computer.

4. Ensure that the default settings are selected; then click Next. The next screen of the MySQL Server Instance Configuration Wizard appears (shown in Figure 2-12), allowing you to modify the initial security settings.



Figure 2-12

In this screen, you can modify the initial security settings by assigning a password to the root user account, the primary administrative user account automatically created in MySQL. If you select the Modify Security Settings option, you can also select the Root may only connect from localhost option, which is used to specify that the root user cannot connect to the MySQL server remotely. In addition, if you select the Modify Security Settings option, you can also select the Create An Anonymous Account option, which allows you to create an anonymous account on the MySQL server.

5. Ensure that the Modify Security Settings option is selected, and then enter a password in the New root password text box and in the Confirm text box. Also, ensure that the Root may only connect from the localhost option and that the Create An Anonymous Account option are *not* selected. Then click Next. The next screen of the wizard appears, as shown in Figure 2-13.



Figure 2-13

- Click Execute. The screen shown in Figure 2-13 provides you with a status report (by checking the option) as each task is completed. When the configuration process has been completed, all tasks should be checked. The screen should then appear as it does in Figure 2-14.



Figure 2-14

As you can see, the screen now indicates that a configuration file has been created, MySQL has been installed as a service, the service has been started, and the security settings have been applied.

- Click Finish. The wizard closes, and MySQL is ready to use.

## Editions of the MySQL Server

When you use the MySQL Server Instance Configuration Wizard to set up the initial configuration of the MySQL server, one of the tasks that it performs is to set up the MySQL server as a service. When the service is started, it accesses the `mysqld-nt` server file to run MySQL. The `mysqld-nt` file, however, is not the only file that you can use to run MySQL on Windows.

Earlier in the chapter, you learned that MySQL supports a Standard edition and a Max edition. (The Debug edition is rarely a consideration.) As you probably noticed when you downloaded the Windows distribution file, you were not given the option to select an edition. Windows distribution files, unlike other operating systems, include all editions in one file, so you do not have to make a decision about editions until after you install MySQL. Initially, that choice is made for you by running the MySQL Server Instance Configuration Wizard. There might come a time, though, when you want to choose a different edition of the server.

The Windows distribution of MySQL includes five editions of the MySQL server. These editions can be divided into the broader categories that you saw earlier — Standard and Max. The following table describes the five server editions available in a MySQL Windows installation.

Server edition	Description
mysqld	This is the basic compiled edition of the MySQL server. It supports all the basic features found in a production release of MySQL. This edition also contains full debugging support, which means that it might run more slowly and use more memory than other editions of the MySQL server.
mysqld-debug	The binaries are compiled with additional debug data and are not intended for use in a production environment.
mysqld-nt	This edition of the MySQL server is optimized to run on Windows NT, Windows 2000, Windows XP, and Windows Server 2003. In addition, the mysqld-nt server supports named pipes.
mysqld-max	The mysqld-max server is the Max edition of the MySQL server. As a result, it includes all features found in the mysqld server, plus additional features.
mysqld-max-nt	Like the mysqld-nt server, the mysqld-max-nt server is optimized to run on Windows NT, Windows 2000, Windows XP, and Windows Server 2003, and it supports named pipes. In addition, because the mysqld-max-nt server is the Max edition of the MySQL server, it includes all features found in the mysqld-nt server, plus additional features.

If you're running MySQL on Windows NT, Windows 2000, Windows XP, or Windows Server 2003, it's advisable that you run the mysqld-nt server, which is the edition used in the book (and the edition installed by default). If at some point you decide that you want the additional features available in the Max edition, you can switch over at that time.

## Testing Your MySQL Installation

Now that you have installed MySQL and have started it, you're ready to test your installation. This is a simple process that is intended to verify that the MySQL server is running and that the mysql administrative database has been properly initialized. For those of you who installed MySQL on Linux, see the "Verifying Your Linux Installation" section that follows. For those of you who installed MySQL on Windows, see the "Verifying Your Windows Installation" section later in the chapter.

*It's unfortunate that MySQL AB uses the same name for so many objects associated with MySQL. When you do a default installation, the term mysql describes the client utility, one of the default databases, the installation directory, the default user and group, and the name assigned to the startup script used to start MySQL automatically in Linux. Despite this, the text attempts to minimize any confusion surrounding its usage.*

### Verifying Your Linux Installation

After you install MySQL, you should verify that it has been properly installed. To do this, launch two of the MySQL client utilities. (You should still be logged on to Linux as the root user.) Follow these steps to verify that the MySQL server is working properly and that the administrative tables in the mysql database have been initialized:

## Chapter 2

---

1. Verify which version of MySQL is running. Do this simply to verify that MySQL was properly installed. If you used RPM packages to install MySQL, type the following command at the command prompt, then press Enter:

```
/usr/bin/mysqladmin version
```

If you used a tar file to install MySQL, type the following command at the command prompt; then press Enter:

```
/usr/local/mysql/bin/mysqladmin version
```

Information about the MySQL installation displays, including information about the version number and the edition of the MySQL server.

2. Next, verify whether the MySQL databases have been properly installed and configured. You achieve this by running the `mysql` command-line utility to access the databases. If you're using an RPM installation of MySQL, enter the following command:

```
/usr/bin/mysql
```

If you're using a tar installation of MySQL, enter the following command:

```
/usr/local/mysql/bin/mysql
```

You receive a message that welcomes you to the MySQL monitor and provides information about command termination, the MySQL connection, and getting help. In addition, the command prompt displays as follows:

```
mysql>
```

At the command prompt, simply type SQL commands as you would type commands in any Linux or Windows shell. When sending commands to MySQL, though, you must terminate your commands with a semi-colon.

3. Try out a couple of commands to verify that the databases have been set up properly. Enter the following command:

```
SHOW DATABASES;
```

You should receive the following results:

```
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+
2 rows in set (0.00 sec)
```

As you can see, two databases have been installed: `mysql` and `test`.

4. Next take a look at the tables contained in the `mysql` database. To do that, you must first switch to the database:

```
use mysql
```

After executing the command, you receive a message saying that the database has been changed. You can now run queries against the `mysql` database.



5. Enter the following command to display the tables in the mysql database:

```
SHOW TABLES;
```

You should receive results similar to the following:

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| tables_priv     |
| time_zone_      |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
15 rows in set (0.00 sec)
```

Depending on the version of MySQL that you're running, you may not see exactly these tables, but you should still see many of the same ones. You will learn more about these tables as the book progresses. For now, it's enough to know that the tables have been properly created.

6. The next step is to verify that the user table contains the proper host and user rows. Execute the following command:

```
SELECT host, user FROM user;
```

A `SELECT` statement retrieves data from the user table. Specifically, you want to retrieve information from the host column and the user column. You should receive results similar to the following:

```
+-----+-----+
| host      | user |
+-----+-----+
| <host>    |      |
| <host>    | root |
| localhost |      |
| localhost | root |
+-----+-----+
4 rows in set (0.00 sec)
```

The `<host>` placeholder refers to the name of the computer on which MySQL is installed. In other words, you see the name of your computer in place of `<host>`. The `localhost` entry also refers to the current computer. Notice that for one of the `<host>` rows and one of the `localhost` rows, the user is defined as `root`. In MySQL, the root user is the primary MySQL user account. When you first begin to use MySQL, you're using it under the root account. Later in the book, you learn how to modify the mysql tables, create users, assign passwords, and log into MySQL under a different account name.

7. Now you can exit the `mysql` utility. To do so, type and enter the following command:

```
exit
```

You can also use the following method to exit `mysql`:

```
quit
```

After you execute one of the commands, you're returned to the Linux command prompt.

If you were able to complete the preceding steps, you can assume that MySQL has been successfully installed and is operating properly. If you ran into any problems, such as not being able to launch any of the utilities or not being able to access the database, make sure that you typed in each command correctly and that you launched each utility from the correct directory. In addition, make certain that you completed the entire installation process before trying to test the installation, particularly if you performed a tar installation. However, once you have successfully verified the MySQL installation, you're ready to go. You can now start creating databases, adding data, and building applications that access the data.

## Verifying Your Windows Installation

To verify that you correctly installed MySQL in your Windows environment, you must open a Command Prompt window and enter commands at the command prompt. Follow these steps to verify that the MySQL server is working properly and that the administrative tables in the `mysql` database have been initialized:

1. Change to the directory where the MySQL utilities are located:

```
cd c:\program files\mysql\mysql server <version>\bin
```

The `<version>` placeholder refers to the version of MySQL that you just installed on your computer. The command prompt should indicate that you're in the `C:\Program Files\MySQL\MySQL Server <version>\bin` directory.

2. The first step to take is to check which version of MySQL is running. Do this simply to verify that MySQL was properly installed. Type the following command at the command prompt, and then press Enter:

```
mysqladmin -u root -p version
```

After you press Enter, you are prompted for a password. Type the password that you entered in the MySQL Server Instance Configuration Wizard when you configured your MySQL installation, and then press Enter. Information about the MySQL installation displays, including information about the version number and the edition of the MySQL server.

3. Next, verify whether the MySQL databases have been properly installed and configured. Achieve this by running the `mysql` command-line utility to access the databases. Enter the following command:

```
mysql -u root -p
```

You are again prompted for a password. Enter the password and press Enter. You receive a message that welcomes you to the MySQL monitor and that provides information about command termination, the MySQL connection, and getting help. In addition, the command prompt displays as follows:

```
mysql>
```

Whenever you use the `mysql` client utility, the command prompt displays in this manner, unless you configure it to display other prompts. At the command prompt, you simply type in SQL commands as you would type in commands at the Windows command shell. When sending commands to MySQL, though, you must terminate your commands with a semi-colon.

4. Try out a couple of commands to verify that the databases have been set up properly. Enter the following command:

```
SHOW DATABASES;
```

You should receive the following results:

```
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.31 sec)
```

5. Next, take a look at the tables contained in the `mysql` database. To do that, you must first switch to the database:

```
use mysql
```

You receive a message saying that the database has been changed. You can now run queries against the `mysql` database.

6. Enter the following command to display the tables in the `mysql` database:

```
SHOW TABLES;
```

You should receive results similar to the following:

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| tables_priv     |
| time_zone_      |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
15 rows in set (0.01 sec)
```

Depending on the version of MySQL that you're running, you may not see exactly these tables, but you should still see many of the same ones. You will learn more about these tables as the book progresses. For now, it's enough to know that you created the tables properly.

7. The next step is to verify that the user table contains the proper host and user rows. Execute the following command:

```
SELECT host, user FROM user;
```

A `SELECT` statement retrieves data from the user table. Specifically, you want to retrieve information from the host column and the user column. You should receive results similar to the following:

```
+-----+-----+
| host      | user  |
+-----+-----+
| %         | root  |
| localhost | root  |
+-----+-----+
4 rows in set (0.00 sec)
```

The `%` wildcard indicates that any host has access to MySQL. The `localhost` entry refers to the current computer. Notice that the `%` rows and the `localhost` row each define the user as `root`. In MySQL, the root user is the primary MySQL user account. When you first begin to use MySQL, you're using it under the root account. Later in the book, you learn how to modify the `mysql` tables, create users, assign passwords, and log in to MySQL under a different account name.

8. Now you can exit the `mysql` utility. To do so, type and enter the following command:

```
exit
```

You can also use the following method to exit `mysql`:

```
quit
```

After you execute one of these commands, you're returned to the Linux command prompt.

If you were able to complete the preceding steps, you can assume that MySQL has been successfully installed and is operating properly. If you ran into any problems, such as not being able to launch any of the utilities or not being able to access the database, be sure that you typed each command correctly and that you launched each utility from the correct directory. In addition, make certain that MySQL has been installed as a service and is started. (You can use the Services utility in Windows to verify if MySQL is running properly.) Once you have successfully verified the MySQL installation, you can start creating databases, adding data, and building applications that access the data.

## Summary

As you have seen in this chapter, installing MySQL is a very straightforward process. It is, of course, easier to install MySQL using RPM files on Linux or a zipped file on Windows than it is to use a tar file to install the product. But once you know how to use a tar file to install MySQL on Linux, you should be able to install it on any Unix-like operating system. This chapter, then, has provided you with the background information and steps necessary to perform the following tasks:

- Select which version of MySQL to install
- Choose a platform on which to run MySQL
- Choose a distribution type
- Select an edition of MySQL
- Download the appropriate distribution files
- Install MySQL on Linux (using RPM and tar files) or install MySQL on Windows (using a zipped file)
- Start the MySQL server and, if installing on Windows, set up the server as a service
- Test your MySQL installation on Linux or Windows

Once you've installed MySQL and tested the installation, you can create databases and tables; insert, modify, and retrieve data; modify the configuration settings; and create data-driven applications that connect to the databases. As you progress through the book, you will learn how to do each of these tasks. Before you learn how to perform any of them, you should have a strong foundation in MySQL itself—how it is set up on your computer, what client tools are provided, and how to use those tools, which are the subjects of Chapter 3.

## Exercises

The following exercises provide you with the opportunity to test yourself on some of what you learned in this chapter. To view the answers to these questions, see Appendix A.

1. You are preparing to install the Max edition of MySQL 4.0.20 on an x86 computer that is running Linux. You plan to install `mysql` from a tar archive file. What file or files should you download from the MySQL Web site?
2. You are installing the Standard edition of MySQL on an x86 computer that is running Linux. You plan to use RPM files to install the server and client tools for MySQL 5.0.0-0. You copy the necessary files to the `/tmp` directory on your computer and then change to that directory. Which command should you use to install the MySQL server on your computer?
3. You used a tar file to install the Standard edition of MySQL on an x86 computer that is running Linux. You performed the installation in the `/usr/local` directory. You now want to start the server in safe mode in the context of the `mysql` user. Which command should you use?
4. You installed MySQL on a Windows XP computer. You now want to set up the `mysqld-nt` server as a service on your system. You then want to start the service. In addition, you want to create a configuration file and assign a password to the MySQL root user account. What is the easiest way to accomplish these tasks?