

After reading this chapter and completing the exercises, you will be able to:

- Identify the considerations that an administrator must take into account prior to installation
- Download and install the binary distribution of MySQL for the most common operating systems
- Configure a MySQL database for both Windows- and UNIX-based platforms
- Secure the installation and configuration of MySQL server



Security In Your World

Tony, the system administrator for Haphazard, Inc. is about to install MySQL for the first time. The company has experienced a bit of success, so a larger database storage plan became necessary. Their current implementation of Microsoft Access no longer meets the needs of their growing clientele and so Tony has been asked to find an inexpensive solution that would offer a way for their external customers to access their inventory through the Web. Tony researches the current available database management systems and decides that MySQL is the best fit for the organization. He presents his proposal to the senior managers, explaining that MySQL offers a free, scalable DBMS that seems simple to install and maintain.

Pleased with his proposal, Haphazard managers promote Tony to the database administrator role and send him to training to prepare him for the administration of MySQL. Unfortunately for Tony, and as often is the case, the training focused primarily on planning and building a database schematic with little emphasis on security.

Equipped with his new knowledge and focused on data administration, Tony migrates the company's Access database with MySQL and begins providing access to the company's external users. After only a few months, the company begins to reap the benefits of the new system and the project seems successful. The clients are pleased with their new real-time access, and the managers are pleased with the speed with which orders are being delivered.

Three months after the implementation, Tony is sitting proudly at his desk when his phones all begin ringing simultaneously. It seemed that several customers were experiencing issues connecting to the database. Unable to access the data himself, Tony called MySQL support for help. To his dismay, the database appeared to be infected by a virus, which had corrupted several major tables within the database.

After many hours and several unsuccessful attempts at recovering the data using antivirus programs, MySQL embedded recovery tools, and the most recent backups, Tony had no other choice but to call a third party to help recover the database. In the end, the company lost money due to the database outage, the technical support fees, and the third-party application. The customers lost trust in the company and its ability to maintain secure and private data and the managers lost faith in Tony as a database administrator. All of this could have been avoided if only Tony had taken a few small security precautions during the software installation process.

It is not uncommon for database administrators to focus on the design of the database and its administration, leaving few resources for database security. This chapter will cover the information needed to download, install, and configure MySQL, as well as learn the basic security precautions for the initial setup.

Preinstallation Preparation

MySQL is a reliable and robust open-source application licensed under the General Public License (GPL). It is the most popular open-source database application used today. MySQL offers many features to its users, and can be customized to fit almost any business or personal environment. MySQL can be found servicing major Internet Web sites as well as popular enterprise corporations.

Flexibility brings choices, and there are a few considerations that an administrator must take into account prior to installation. This section contains the necessary preinstallation considerations needed to ensure the best fit for an environment. Topics covered in this area include distribution formats, MySQL versions, supporting platforms, and technical support lines.

3

Choosing a Distribution Format

Due to MySQL's open-source nature, there are different ways MySQL can be installed. These options vary from simple prepackaged executions to individualized custom downloads. The two main installation formats in which you can choose to install MySQL are source code installation and binary code installation.

Source code is a group of statements or functions that are written using a specific programming language and then combined to create a specific type of application or utility. Source code usually refers to the main code that makes up an application. A **programming language** is a type of synthetic language developed with specific syntax and semantic rules that allow individuals to create statements or functions to interface and control the behavior or functionality of a machine. **Source code installations** of MySQL allow users to download the actual MySQL source code, change it, customize it, and compile it into a binary file for installation execution.

A **binary file** is a file that contains code that can be read by machines and run as an executable file. **Binary code installations** of MySQL are already packaged as binary files and are ready to be installed without the need for compiling the code to enable it to be run as an executable file on a particular machine. **Compiling** refers to the act of converting source code that is written in one language into a different programming language or machine language.

MySQL is written using C and C++ programming languages. There are several different types of programming languages, and each is developed to interface with a particular type of system or device. Compiling source code results in a binary file. Compiling MySQL can be time consuming, taking up to 60 minutes to compile.

There are advantages and disadvantages to both source code and binary distribution formats. Although most environments choose to install the binary distribution formats, it is important to know what each distribution offers prior to installation. Source code distribution formats require more steps to complete than binary file installations. Although they allow for customization within an environment, expertise is necessary to achieve this goal. In this case, software programmers would be used to create a slightly different, more compatible version of MySQL for installation within their own environment. For example, an environment that requires a small lightweight database application may choose to install the source code distribution format and remove all of the features that exist within the default packaging of a specific version of MySQL. This would be done prior to compiling and can greatly increase a database network's efficiency.

Another reason for using the source distribution format would be for strict environments where specific compilers must be used; however, this is a unique scenario. Despite its customization, installing a source code distribution format can be quite cumbersome because it requires additional steps and longer compiling times. Source distribution installations also require detailed documentation that describes any customization that has taken place. This is to ensure that any updates to the software are handled appropriately, given the original changes that were made.

Binary distribution formats are easy to install and require less time and expertise. Since they are already executable files, customization cannot take place and compilers cannot be changed. As you will see in this chapter, binary distribution installations are as simple as downloading and installing a small software application found on the Internet.

Prior to making a decision about using source code distribution installations, all alternative choices should be thoroughly considered. MySQL offers a number of different versions from which to choose, and there are also third-party vendors that distribute their own versions of MySQL with customized features to meet the needs of specific environments and operating systems. Only around 5% of the MySQL installations are of source code distribution formats, which should offer a clue as to how many different versions and flavors are available.



Although third-party distributions of MySQL can appear to be a great fit for a specific environment, the templates are often out of date and do not include the updates, fixes, and features that are available with the most current Sun distribution.

Choosing a Version of MySQL

As mentioned in the preceding section, MySQL comes in many different version releases and includes third-party vendor developed binary files to offer custom templates of MySQL installations. These releases are systematic and predictable. Identifying the most stable MySQL available is important to the success of an implementation of the database application.

Sun Microsystems releases new versions of MySQL subsequent to one another while they are at different levels or stages of maturity. New versions of the database are released in their testing and debugging phase, while older, more stable versions are still available. This allows organizations to test upcoming versions in a controlled environment prior to their true production release. It is important to review the phased releases of the database application prior to exploring the available versions of MySQL to ensure that the most stable versions are explored and installed. Originally, there were four main phases of maturity for a release of MySQL: General Availability (GA), Release Candidate (RC), beta, and alpha.

- **General Availability**—General Availability-labeled software is ready for production. In this stage of maturity, it is a stable application and there are very few known bugs.
- **Release Candidate**—Research Candidate-phase software is considered somewhat stable, though it might have serious bugs that still exist, but do not affect everyone. Normally these bugs only affect unique and rare environments.
- **Beta**—In the beta phase of the application, there are known issues that exist with the software. This software should be used for testing and experimental purposes only. These versions should not be downloaded for use in a working environment.

- *Alpha*— The alpha phase is where new features are added. This is the very early stage of testing. Like the beta stage, software in this phase should not be downloaded for use in a working environment.

Around the time that the MySQL 5.1 production phase was released in 2008, a new approach to the phase and release cycle was introduced, called the milestone model. This model was implemented to improve the efficiency of the version phase and iteration process as well as allow for smaller, more frequent, and controllable releases of the MySQL database software. Unlike the previous phased approach, the new milestone approach focuses on smaller iterations of updates and changes. In the prior phased approach, where applications are released in a beta form with many changes to the application being released, the milestone model does not release MySQL in any stage lower than the beta stage. The version numbers use a more specific naming scheme, and a suffix has been added to the version number to indicate the phase of a particular release. Let's examine the release version MySQL 5.5.0-m2:

- The first number of a version indicates the main file format for a particular application. In this case, we have a five. All MySQL applications with the major version number 5 maintain the same file format.
- The second number is the release level and indicates the number of major changes or updates that have been applied to a particular application. For the preceding example, there have been five major changes or added features to this release.
- The third number indicates the number of smaller changes that have been tested and released for that particular application. In our example, there have been no minor changes.

As you can see from the example, the suffix that has been added to the version number is *m2*. This suffix indicates how stable the application is, similar to the old GA, RC, beta, and alpha system. There are three main categories for which the suffixes apply. Table 3-1 defines these suffix types.

Suffix	Meaning	Example
m1, m2, m3, etc.	A suffix of m1, m2, m3, etc., indicates the number of milestones that have been achieved thus far This stage is similar to the beta stage described earlier.	MySQL 5.5.0-m2 For this example, two milestones have been achieved; a milestone is a small focused change or feature added to an application that has been tested rigorously
rc	A suffix of rc indicates that an application is a release candidate	MySQL 5.5.0-RC For this example, the application is considered to be somewhat stable, although it may have serious bugs that still exist but do not affect everyone; normally these bugs only affect unique and rare environments
None	If no suffix is provided at the end of a version of a MySQL application, then the application is considered to be of General Availability	MySQL 5.5.0 For this example, the application is ready for production; in this stage of maturity, there are very few known bugs and it is a stable application

Table 3-1 Suffix types

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Another goal of the milestone model was to provide releases that are more predictable than in the past. With the milestone model, RC releases occur every 3–6 months, while GA releases occur every 12–18 months. Refer to Table 3-2 for a complete history of MySQL releases and added features. More details on the MySQL lifecycle policy can be found at <http://www.mysql.com/about/legal/lifecycle>.

Year	Version	Notes
1994	Original development	
1996	Public release for Solaris	
1998	Windows version released for Windows 95 and NT	
2000	Version 3.23 beta release	BDB and InnoDB engines
2001	GA production release of version 3.23	
2002	Beta version 4.0 release	
2003	GA production release version 4.0 three months after Beta release version 4.01	Version
2004	Beta and production versions of 4.1 are released	Subqueries R-trees, B-trees, prepared statements
2005	Beta and production versions of 5.0 are released	Cursors, stored procedures, triggers, views, XA transaction Federated Storage Engine included as a default engine
2006	Beta release version 5.1	Partitioning, row-based replications, plug-in storage engine API, server log tables, and event scheduler
2007	6.0 Alpha version released	Falcon and Maria storage engines and online backup
2008	Sun Microsystems acquired MySQLAB version 5.1 production, version production release 5.4-m1	Version 6.0 withdrawn Version 5.4 upgraded features: improve scalability on multicore CPUs, I/O subsystem changes, enhanced Solaris support, diagnostic, and monitoring capabilities
2009	Version 6.0 cancelled Version 5.5.0-m2	Semisynchronous replication, key caching for index, two new users defined, enhanced XML functionality

Table 3-2 Timeline of MySQL releases

New versions of MySQL are released every 3–6 months, and third-party vendors supply their own editions or templates of the database application. In 2007, MySQL 6.0 was released in the alpha phase, yet only a year later the milestone model was introduced. To maintain consistency with the (then new) milestone model, MySQL 6.0 was withdrawn due to its advanced maturity. When choosing a version of MySQL to install for the first time, it is recommended that you use the most stable version, which is the current Generally Available version. This version can be identified with no suffix added to the milestone model versions.

Supported Platforms

Users are encouraged to download, modify, and use MySQL free of charge. It is for this reason that MySQL has obtained widespread popularity. MySQL works on several different operating system platforms. These platforms include AIX, BSDi, eComStation, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UNIXWare, Sanos, and Tru64. You can find more details, as well as a complete list of the platforms supported, on MySQL's site.

Platform information for the Community Edition of MySQL can be found at <http://dev.mysql.com/doc/refman/5.5/en/which-os.html>. You can also find platform information for the Enterprise Edition of MySQL at <http://www.mysql.com/support/supportedplatforms.html>.



As with operating systems, MySQL is available as a 32-bit and 64-bit system. The suggested install is 64-bit. Ideally, MySQL would be installed as a 64-bit software application residing on a 64-bit operating system.

Locating Help

There are a great number of places to locate support for MySQL. One option for obtaining help is through a Sun Microsystems paid service called MySQLAB. This service will connect users and administrators directly to MySQL developers to answer questions, receive consultations, or obtain training. There are also several third-party companies that exist to provide this type of help as well; however, most of the resources that are available for MySQL can be found free online. Below is a list of the help that can be obtained through online services:

- *MySQL manual*—This resource provides a comprehensive online manual for all of the current versions and editions of MySQL. These manuals provide great resources for MySQL, and can be found at <http://dev.mysql.com/doc>.
- *MySQL forums*—Forums are a great way to connect online with other users and MySQL professionals that have a wealth of knowledge to share. There are several great forums available throughout the Internet. One suggested group of forums can be found on MySQL's site at <http://forums.mysql.com>.
- *Mailing lists*—Subscribing to mailing lists is a great way to receive updates on all different types of technology. Through the MySQL mailing list, a user can ask questions and receive information on the most current updates and alerts at <http://lists.mysql.com>.
- *Bloggers*—Blogs have become a fantastic resource, especially in the technology fields. Many people who work with MySQL or have experience with MySQL blog about their experiences and share their knowledge. These blogs can be found at <http://planet.mysql.com/>.
- *Twitter*—Twitter has become a popular way to keep people informed on a minute-by-minute basis. Almost every organization has a Twitter page. It provides a quick and easy way to reach customers by sending quick updates and reminders; organizations can reach out to their customers directly via e-mail and cell phone. MySQL uses Twitter as a way for users to ask questions and for MySQL professionals to send out

alerts relevant to upgrades, bugs, and other software-related support, and can be found at <http://twitter.com/MySQL>.

- **Bug alerts**—MySQL, like many technical vendors, provides a Web site at <http://bugs.mysql.com> that is dedicated to obtaining information on the most recently discovered problems for all of its applications. This site can be used to report a problem or use the sophisticated search utility to find an already-known bug for a system.

Downloading MySQL

The official download site for MySQL is <http://dev.mysql.com/downloads>. This Web site contains all available versions, editions, and distribution formats of MySQL. When it comes to MySQL, you have two separate options for download: Community and Enterprise.

The Community Edition of MySQL is the most popular edition available as open source and is referred to throughout this chapter. The Enterprise Edition of MySQL provides additional assistance for monitoring and analyzing the performance of your database server.

Although it is not necessary to purchase the Enterprise Edition of MySQL server, it is recommended for large organizations. Providing tools such as MySQL Enterprise Monitor and MySQL Query Analyzer, an administrator is likely to find that the cost of this edition will override the short- and long-term benefits for larger enterprise organizations.

The Enterprise Edition also offers support, training, and consultation directly from MySQL developers. For large organizations, this in itself may be worth the cost. MySQL downloads are often combined and packaged as platform-dependent formats such as .tar, .rpm, .zip, and gz. Therefore, additional applications are often necessary to extract the files from within these packages. For example, for Windows, a .zip-compressed package is downloaded that contains the files to run the Windows platform compatible with MySQL.



To extract these files after download, the compatible decompression and unpacking tool named WinZip is needed.

NOTE

Once MySQL is downloaded and before it is installed, steps can be taken to verify the integrity of the files included within the package. The verification checks listed below are used to ensure that a package has not been tampered with and is free from Trojans:

- **MD5 Checksum**—Each package that is downloaded from MySQL is assigned an MD5 checksum, or a hexadecimal code. This code appears on the downloaded pages of MySQL, as well as within the installed package itself. The code retrieved from the downloaded pages can be compared with the code retrieved from within the package to determine a match. If the two codes match, then the application is verified. If the two codes are different, then the file should be discarded and a new file should be downloaded. How the code is retrieved from within the package is dependent on the operating system that is being used. For example, in most UNIX-based operating systems, the program `md5sum` is used to extract the MD5 code from the package, while Windows uses a variant of the `md5sum` called the Microsoft File Checksum Integrity Verifier to extract the code.

- **GnuPG**—MySQL uses Gnu Privacy Guard to add digital signatures from which authenticity can be determined. A **digital signature** is code that uses cryptography to verify the authenticity of a source of information. Therefore, MySQL users can obtain a key to the encrypted digital signature in order to verify that the package came from MySQL, the trusted source. Digital signatures and cryptography are discussed in later chapters.

Verification of your downloaded package takes very little time, and can help to maintain the confidentiality, integrity, and availability of a database application. You can find more information regarding MySQL's package verification at <http://dev.mysql.com>.

3

Installation

Once a user has downloaded and verified a copy of MySQL in the distribution format edition and version that best fits the needs of the environment, installation can begin. This section will include the steps for installing MySQL on both Windows- and UNIX-based machines. The instructional steps provided in this section are intended to provide steps for installing the binary distribution formation of the Community Edition of MySQL 5.5.

Installing on Windows

The Windows Operating System is a popular platform on which MySQL is installed. This is partly due to its availability for Windows. MySQL is available for virtually all Windows Operating Systems active today, and both 32-bit and 64-bit versions are supported. The following section will provide information and instructions on binary installations of MySQL on Windows.



NT-Based systems (e.g., NT, 2000, XP, 7, Server 2003, and Server 2008) can run MySQL as a service, allowing it to start and stop with a system boot up and shut down.

Installing MySQL on Windows Using an Installer Package MySQL offers three different binary packages for the Windows operating system. This makes it easy for users by automatically installing and configuring MySQL so that once installed, the application can be used immediately. There are two automatic installer packages: the Essentials Package and the Complete Package, and one Noinstall Archive. The installer packages make for easy application since they include installation wizards that automatically load the server binary configuration wizard. The configuration wizard automatically creates the option files and initial user's accounts.

Here is a brief description of these packages:

- **The Essentials Package**—As its name implies, this package includes only the essential or necessary files for binary installation. A configuration wizard is provided and optional features are not included. This package is ideal for almost any environment because it allows for a lighter installation of MySQL.
- **The Complete Package**—As its name also implies, this package provides the complete installation, which includes both necessary and optional files and features. This package should only be used if an optional component is necessary, which in most cases it is not.

- *The Noinstall Archive*—Unlike the other two packages, this package does not include the automatic installation or configuration wizard. This package contains the same components as the Complete Package, yet it requires manual installation and configuration, making this package the most difficult to apply.

It is necessary to run these installers from within an administrative account on the operating system to avoid any issues during the installation. In Windows Vista and Windows 7, a port within the Windows firewall must be added to be able to configure the server after the installation. Use the following steps to add a MySQL port in Windows Vista and Windows 7.



The instructions provided in the following section are intended for default installations of Windows Vista and Windows 7, and are best suited for environments where the database administrator maintains administrative security rights on the system. Adjustments to these instructions may be necessary and collaboration with the network security team for your organization may be necessary in environments that are restrictive or atypical.

Adding a MySQL Port in Windows Vista

1. Click **Control Panel** found on the Start menu of Microsoft Windows.
2. Click **Security**.
3. Click **Windows Firewall**.
4. Click **Allow program through Windows Firewall**.
5. Click **Add Port**.
6. Provide a name in the Name text field. Pick a name that is descriptive enough that it will remind you what this port has been created for (e.g., MySQL).
7. Provide a port in the Port Number text field (the suggested port is 3306).
8. Click **TCP Protocol**.
9. Click **OK** to confirm your choices.

Be aware that opening ports in your firewall provides all computers access to MySQL through your firewall. For security reasons, it is suggested that access to MySQL is limited. This can be accomplished by changing the scope of the port. In Windows Vista, you can change the Port Scope, or group of computers that are able to use this port opening by clicking the Change Scope option before clicking OK (Step 9 above). This will limit the number of computers and users that have access to this port opening.

Adding a MySQL Port in Windows 7

1. Click **Control Panel** found in the Start menu of Microsoft Windows.
2. Click **System and Security**.
3. Click **Windows Firewall**.
4. Click **Advanced settings** on the left pane.
5. In the **Windows Firewall with Advanced Security** dialog box in the left pane, click **Inbound Rules**. In the far-right pane, click **New Rule**.

6. Click **Port** from the New Inbound Rule Wizard dialog box and click **Next**.
7. Apply the rule to TCP.
8. Specify the port in the Port Number text field (the suggested port is 3306).
9. Click **Next**.
10. Click **Allow the connection it is secure**. This is only compatible with Windows Vista and later versions of Windows.
11. Click **Next**.
12. Add the list of allowed user connections, as well as exceptions (if applicable).
13. Add the list of allowed computer connections, as well as exceptions (if applicable).
14. Click **Next**.
15. Select whether computers will be connecting to the server from within the domain, a private location, public location, or all locations. The public setting is not recommended unless completely necessary. It allows users and computers from public networks to connect to the MySQL server.
16. Provide a name in the Name text field. Pick a name that is descriptive enough that it will remind you what this port has been created for (e.g., MySQL).
17. Click **Finish** to confirm your choices and add the port.



Windows 7 offers additional customization features when specifying the type of connection for a port (Step 9). Explore these features for enhancement of the security of your database.

Installation Instructions

The installation package installed in this section is the binary distribution file for the Essentials Package, version MySQL 5.5.1 m2, downloaded from <http://dev.mysql.com/downloads/>.

1. Locate the downloaded file. If necessary, unzip the setup file using a compression utility tool. Double-click the **Microsoft Windows Installer (MSI)** file to execute (Figure 3-1).



How you start this wizard depends on the installation package that you download. Some packages require extraction and contain a **setup.exe** file rather than an **MSI** file. If this is the case, double-click **setup.exe** to execute the installation.

2. A security warning may appear that prompts the user to confirm that the system request is approved and intentional. This window may appear several times throughout the installation and configuration process. Click **Run** to begin the installation (Figure 3-2).



You may also encounter an End User License Agreement that you must read and accept to continue the installation.

3. The first window encountered will provide a general overview of the MySQL server release that was downloaded. Verify the accuracy of this window and click **Next** to continue (Figure 3-3).

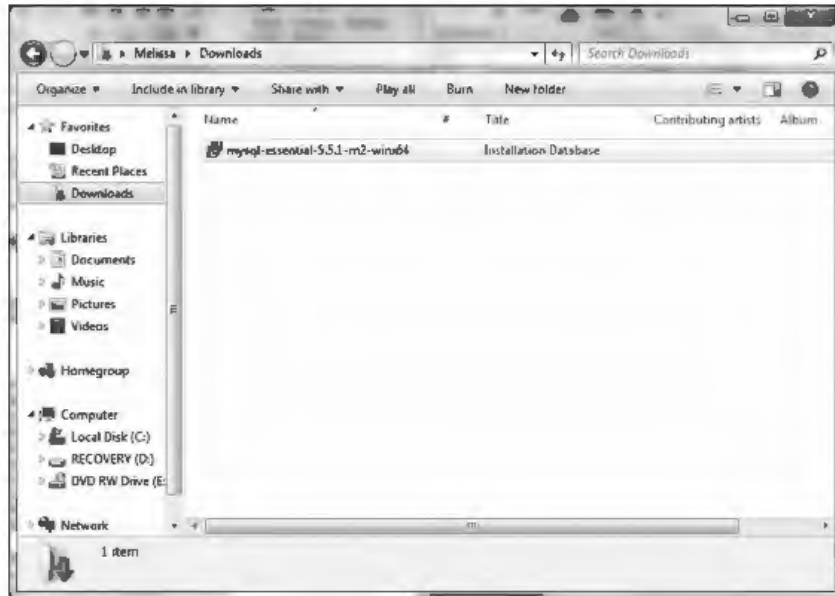


Figure 3-1 Locate the downloaded file

© Cengage Learning 2012



Figure 3-2 Security warning

© Cengage Learning 2012

4. There are three installation types that are available in the Essential and Complete packages: typical, complete, and custom. The type of installation chosen will depend on the needs of the environment:

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.



Figure 3-3 Setup welcome

© Cengage Learning 2012

- *Typical*—Installs the MySQL Server and various command-line utilities for managing the server.
- *Complete*—Installs the full package, which includes the embedded server library, a benchmarks suite, support programs, and documentation.
- *Custom*—Gives the administrator or user full control over the packages installed and the location of the files installed.

Select either **Typical** or **Complete** (Figure 3-4). Click **Next**. Once the choices made have been verified, click **Install** to begin installation (Figure 3-5).

Clicking **Custom** will bring up the Custom installation screen (Figure 3-6). The Custom installation screen lists the components available for installation. If these components are not selected for the current installation, they are marked with an X, as shown in Figure 3-6 for the feature description **C Include Files/Lib Files**.

5. Changes can be made to the way these features are installed by clicking the icon next to the component name. For example, if a user would like to change the installation state of the Client Programs, they would click the icon to the left of the name and pick an option from the drop-down list that appears (Figure 3-7).
6. If the user would like to change the directory in which the client program is installed, they can do so by highlighting the name of the component that the directory needs to change and click **Change** to customize the folder name and directory path of the installed component (Figure 3-8).

In this window (Figure 3-8), type the new directory in the Folder Name field or use the Look In list arrow to choose the custom installation directory from a drop-down menu. Once any changes to the path have been made for that component, click **OK** to return to the Setup Wizard dialog box.

**Figure 3-4** Installation type

© Cengage Learning 2012

**Figure 3-5** Verify installation

© Cengage Learning 2012

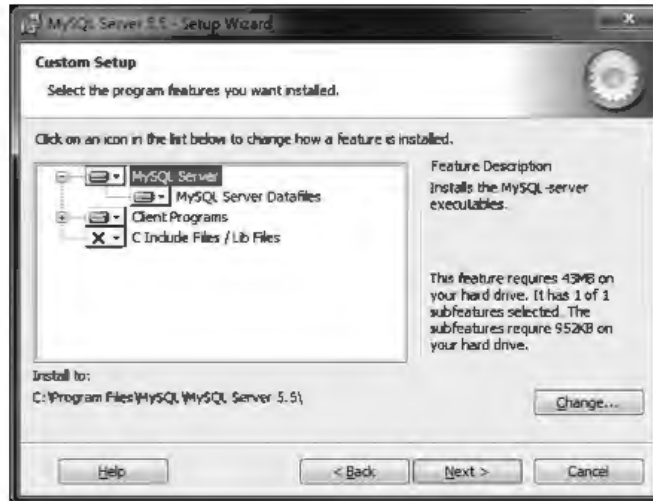


Figure 3-6 Custom installation
© Cengage Learning 2012



Figure 3-7 Select program features to be installed
© Cengage Learning 2012

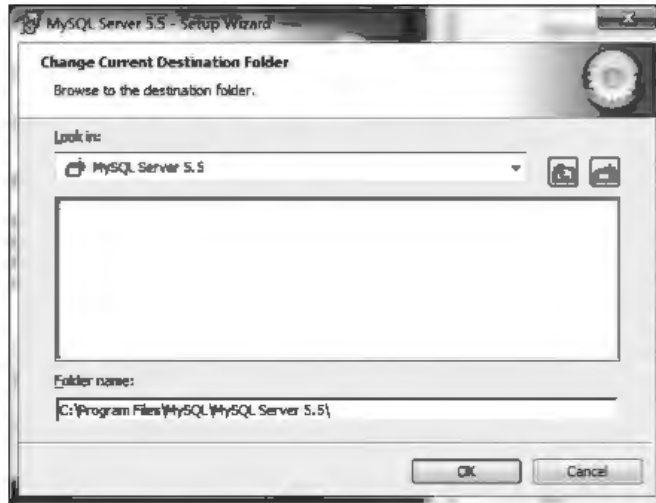


Figure 3-8 Change current destination folder

© Cengage Learning 2012

7. Once the desired installation customizations are complete, click **Next** on the Setup Wizard to advance to the verification screen (Figure 3-9). The verification screen displays all of the current installation selections for review. If everything on this screen is correct, click **Install** to begin the installation.

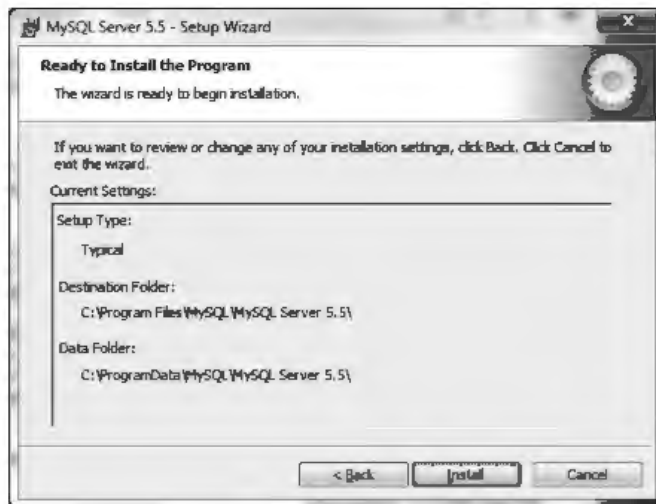


Figure 3-9 Ready to install

© Cengage Learning 2012

Installing on UNIX

UNIX-based systems are also popular platforms on which MySQL is installed. These include Sun Solaris, AIX, BSDi, FreeBSD, HP-UX, IBM, Linux, etc. There are a number of different variations of UNIX-based server package formats to install on these systems. The instructions for installing MySQL on UNIX-based systems are essentially the same, and it is suggested that binary distribution should be used, if possible. The following section will provide information and instructions on installing binary packages for UNIX-based systems.

3

Installing UNIX Binary Distributions

It is quite simple to install binary distributions on a UNIX-based system. As mentioned earlier, MySQL downloads are often combined and packaged into platform-based formats. This is especially true for UNIX-based operating systems.



GNU tar and GNU gunzip are needed to extract the files. These tools are most often included with UNIX-based systems; however, copies of GNU gunzip and GNU tar can be obtained at www.gnu.org.

Installing Tar.gz

1. The initial step in installing MySQL on UNIX-based operating systems is to create users and groups. If users and groups are not created, a default user “mysql” and group “mysql” will automatically be created when installing the package. The following commands are executed as root by typing the following:

```
groupadd mysql          #Creates a group "mysql"
useradd -g mysql mysql  #Creates a user "mysql" and places that
                        user in
                        #the group "mysql"
```

2. Next, input the commands that change the directory where MySQL database files are to be extracted and the tar utility is used to unzip and unpack the distribution file. This can be done using the following commands:

```
cd /usr/local           #Changes the directory
tar xvfz /tmp/mysql-pathname.tar.gz #Unpacks the files
```

On the workstation, the word “pathname” would be the actual file and pathname of the file. For Solaris machines, tar would be replaced with gtar.

3. The final step is to place the appropriate files into the directory in which the machine expecting these files to be located. This is done by making the subdirectory created by tar during the unpacking link with the directory in which UNIX is expecting the software and data to be located. To do this, the following commands are used:

```
ln -s /usr/local/version /usr/local/mysql
```

In this example, version would be replaced with the name of the subdirectory created by tar. There are a few discrepancies between UNIX-based machines as to where the data and the software are expected to be, so be sure to check MySQL’s site for this information.

At this point, the database is essentially installed. It now only requires the granting of privileges and tables, as well as changes in the ownership of the data files and programs. This can be completed by executing the following commands:

```
cd/usr/local/mysql
./scripts/mysql_install_db
chown -R mysql /usr/local/mysql
chgrp -R mysql /usr/local/mysql
```



MySQL server binaries for UNIX area called `mysqld`.

NOTE

Installing Linux Binary Distributions Using RPM

MySQL using RPM packaging is a very fast and simple way to install MySQL on a UNIX-based machine, and for this reason it is the suggested avenue for installing MySQL on a Linux operating system. MySQL is compatible with all versions of Linux that support RPM packages and use the `glibc2.3` library. There are both server and RPMs available that should be installed. MySQL downloads are often combined and packaged as platform-dependent formats such as `.tar`, `.rpm`, `.zip`, and `.gz`. Therefore, additional applications are often necessary to extract the files from within these packages. The installation package installed in this section is the binary distribution file for the RPM package. Like the Windows packages, it can be downloaded from <http://dev.mysql.com/downloads/>.

There are two categories of RPM packages, those that are platform-specific and those that are non-platform specific.

- *Platform specific*—Linked dynamically to libraries that are found on the specific platforms for which they are built.
- *Non-platform specific*—Statically linked to Linux threads.

Installing RPM

1. The initial step in installing MySQL in UNIX-based systems is to create users and groups. If users and groups are not created, a default user “mysql” and group “mysql” will automatically be created when installing the package. The following commands are executed as *root* by typing the following:

```
groupadd mysql          #Creates a group "mysql"
useradd -g mysql mysql  #Creates a user "mysql" and places that
                        #user in
                        #the group "mysql"
```

2. It is important to ensure that no other packages of MySQL are installed on the system. If another MySQL installation is found on a system, an upgrade must be completed. The command for locating other RPM packages is

```
rpm -qa # This will list all rpm files installed on the system.
```


3. Next, we need to install the client and server packages for database connectivity and use. Two simple commands are used to install the RPM distributions of these packages. To install, type the following command into a terminal:

```
rpm -i MySQL - server- VERSION.glibc23.i386.rpm
rpm -i MySQL -client- VERSION.glibc23.i386.rpm
```

There are also additional options available for this command to allow monitoring such as `-v`, which provides a more informative output useful for troubleshooting installation errors, and `-h`, which displays the progression of the installation. Two packages of the same type can be installed at the same time, but the filenames should be separated with spaces:

```
rpm -i MySQL - server- VERSION.glibc23.i386.rpm MySQL -client-
VERSION.glibc23.i386.rpm
```

At this point, the database is essentially installed and data is placed in the `/var/lib/mysql` directory. Data directories and tables now need to be set up. Once complete, you should be able to connect to the server to see if everything works. Test your configuration by running:

```
MySQL -u root
```

3

Configuration

After installation, MySQL must be configured. In Windows, the server type and role need to be identified, and resources need to be allocated for storage engines and the server in general. A root password and file ownership properties must be identified as well. Both Windows and UNIX-based systems (msqld) use a central configuration file to configure MySQL on startup. This file is called `my.cnf`, and in Windows it can also be called `my.ini`. When a server starts up, it looks for this file in several different areas of the file structure. There is a very intentional order in which the operating system looks for this file in order to complete the configuration-related tasks. This section will provide the initial and detailed configuration tasks for both Windows and UNIX.

Configuring MySQL on Windows

The MySQL Server Instance Configuration Wizard creates a custom configuration file (`my.ini`) immediately after installation. As shown in Figure 3-10, the wizard prompts the user at the end of installation to begin the initial configuration of a server. If the user accepts the option to configure the machine, the wizard begins and the custom configuration file comes into fruition.

The first choice that is presented to the user is in regard to configuration types. There are two types of configuration that can be conducted using the Configuration Wizard, *detailed* and *standard*:

- *Detailed configuration*—Gives the user more control over the configuration of the server and provides the user the opportunity to set up an ideal server for the given environment. There are several more options available for users to customize the server within the detailed configuration.
- *Standard configuration*—Meant for initial configuration only, and very little decision-making is left up to the users. With the exception of services and security configurations, the standard configuration runs automatically, setting up and conducting the necessary configuration tasks. The standard configuration type is to

be relied on by users who need a quick install, are installing MySQL for the first time, and plan to reconfigure the server manually at a later time.

Standard Configuration Using the Windows Configuration Wizard

The following steps are for a typical initial standard MySQL configuration using the Windows Configuration Wizard.

1. Check **Configure the MySQL Server now** (Figure 3-10) on the confirmation window at the end of the MySQL installation. Click **Finish** to complete the installation.

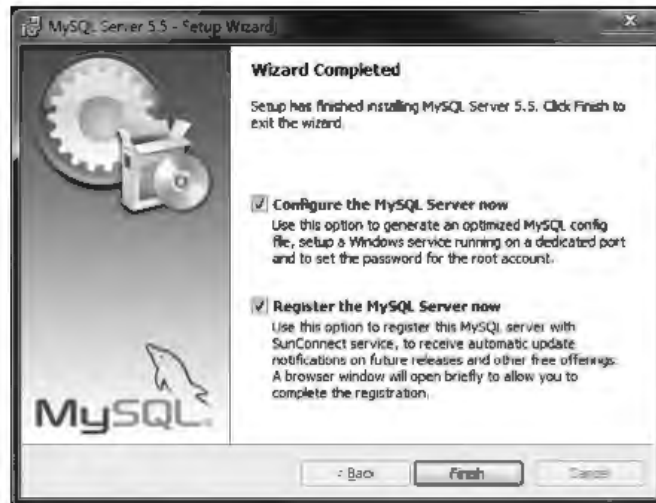


Figure 3-10 Setup Wizard completed

© Cengage Learning 2012

2. Select **Standard Configuration** (Figure 3-11) and click **Next**.
3. The Configuration Wizard automatically creates a service called MySQL (using mysqld) that launches with the start of the machine. From the following screen, users can change the name of the service, disable the service from installing, or disable the service from starting up with the machine. It is suggested that MySQL be installed as a service and automatically initiated at startup. For this example, check **Install As Windows Service** and **Launch the MySQL Server automatically** (Figure 3-12). Click **Next** to continue.
4. To maintain the security of your database server, it is important that the user password is created for the root user, that remote access is denied, and that anonymous accounts are disabled. The root user has full privileges on a MySQL server. Skipping this setting could provide intruders with easy access to the most critical components of your database. Anonymous accounts are often mistakenly left enabled on a system, therefore giving public access to the system without requiring a username and password. This provides an open door into the database for unwanted intruders. Root access allows

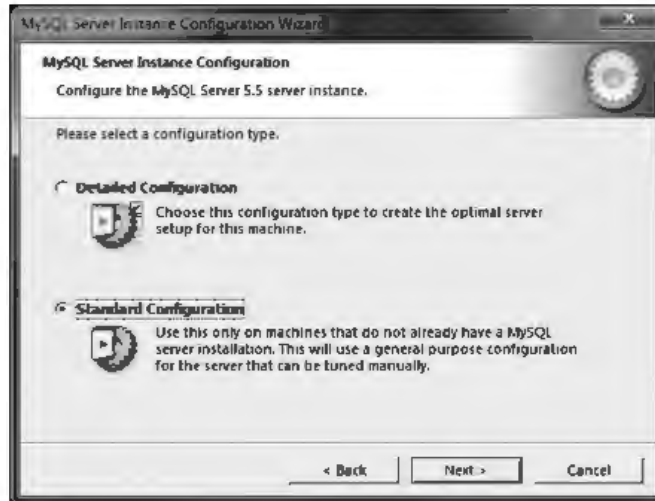


Figure 3-11 Select configuration type

© Cengage Learning 2012



Figure 3-12 Windows options

© Cengage Learning 2012

administrative access to the database server. Unless absolutely necessary, root access should not be given to outside remote machines. This would provide an opportunity for a remote intruder to take full control of your server. Therefore, always ensure that remote root access and anonymous accounts are disabled and that the root password is carefully considered. It is important to assign a strong password to the root account (Figure 3-13). Once the security options are in place, click **Next**.



Figure 3-13 Instance configuration security options

© Cengage Learning 2012

5. The confirmation screen allows a user to save the options to the configuration file (**my.ini**). Click **Execute** on the confirmation screen to save the options to the configurations (Figure 3-14).
6. Now that the **my.ini** file has been created and configured and the root password and security options have been applied, click the **Finish** button on the confirmation screen.

Detailed Configuration Using the Windows Configuration Wizard

The following steps are for a typical initial detailed MySQL configuration using the Windows Configuration Wizard.

1. Check **Configure the MySQL Server now** on the confirmation window at the end of the installation of MySQL (Figure 3-15). Click **Finish** to complete this installation.
2. Select **Detailed Configuration** and click **Next** to continue (Figure 3-16).
3. Three server types are available from which to choose to configure the MySQL machine: developer, server, and dedicated. The choice depends on the role of the machine, the number of other applications that will reside on it, and also will have an impact on the amount of RAM the hard drive space and the CPU will be allotted for the database.

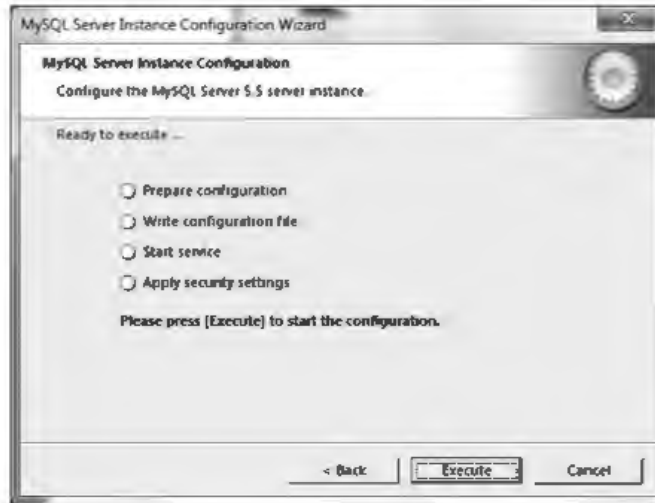


Figure 3-14 Execute instance configuration

© Cengage Learning 2012



Figure 3-15 Setup Wizard exit screen

© Cengage Learning 2012

- **Developer**—The role of this machine is intended for developmental use and will likely house several applications. MySQL server will be allotted only the minimal amount of resources for the database. An administrator would choose this option if the machine on which MySQL is installed will be used to develop, build, and deploy applications.

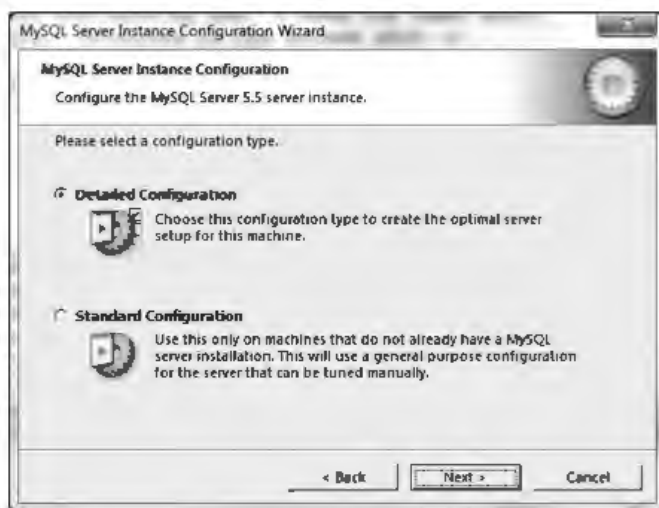


Figure 3-16 Select configuration type

© Cengage Learning 2012

- **Server**—This machine is intended to house several other server applications. MySQL will be allotted a medium amount of resources. An administrator would choose this option if the machine on which MySQL is installed will act as another type of server alongside MySQL.
- **Dedicated**—This machine will be used only as a MySQL database server; no other unrelated applications or services will be housed on this machine. MySQL will utilize all of the available resources of the machine. This is the recommended setting for a MySQL database.

For this example, select **Dedicated MySQL Server Machine** and click **Next** (Figure 3-17).

4. The next step in configuring your MySQL database is to choose the way the database will be used. There are three options from which to choose:

- **Multifunctional**—This database is for general-purpose use. This option will enable both InnoDB and MyISAM storage engines to be available simultaneously. Resources will be equally distributed to both engines.
- **Transactional**—This database will also enable both the InnoDB and MyISAM engines; however, MyISAM does not support transactions. InnoDB will be used more extensively than MyISAM. This is the suggested setting to ensure that the InnoDB obtains more resources and acts as the main storage engine.
- **Non-Transactional**—This database is only used for simple Web applications and monitoring capabilities. InnoDB is disabled with this option. All resources are allocated to the MyISAM.

For this example, select **Transactional Database Only** and click **Next** (Figure 3-18).



Figure 3-17 Select server type

© Cengage Learning 2012



Figure 3-18 Database usage

© Cengage Learning 2012

5. The InnoDB tablespace information is stored inside a file called `ibdata1`. The information saved in this file should be saved on the drive with the greatest amount of storage and speed. This screen allows users and administrators to move the

InnoDB tablespace files to a different location. The drop-down menus provide alternative places to store the files and the screen provides information regarding each of the drives as you select them. For this screen, leave the default unless your machine contains a faster drive with more storage in which to save the tablespaces. Click **Next** (Figure 3-19).

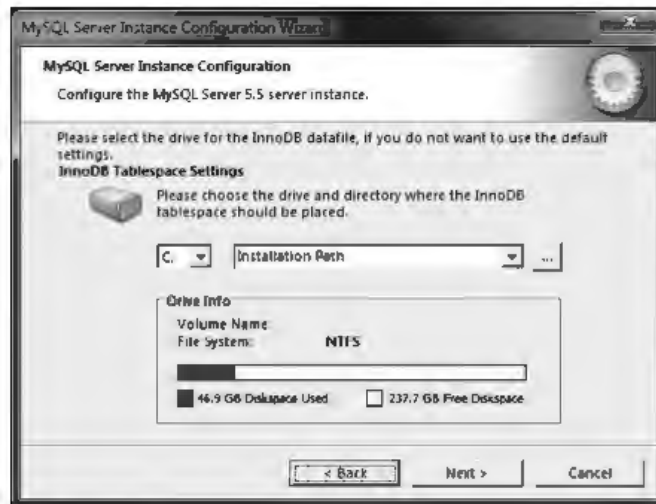


Figure 3-19 InnoDB tablespace settings

© Cengage Learning 2012

6. The next screen allows users and administrators to set the maximum number of concurrent connections in the `mysqld`. Select **Manual Setting** and enter the approximate number of concurrent connections. A good rule of thumb is to use the number of agents that you are monitoring multiplied by two. Resources available on the machine should also be considered in this decision. Click **Next** (Figure 3-20).
7. This next configuration option enables users and administrators to change the port in which TCP/IP connects to MySQL, as well as disables the use of TCP/IP in general. Click **Strict Mode**. Strict Mode ensures that there are no implicit data changes and if one occurs, an error is generated. If this is disabled, implicit data changes can occur and no errors will be generated. It is suggested that Strict Mode remain enabled. For this example, check **Enable TCP/IP Networking**. Click **Next** (Figure 3-21).
8. The next configuration encountered is the default character set within MySQL. This determines the type of characters that can be used globally in the databases and tables. Changes to this option would change the available characters. The default character set is best suited for English and West European Languages. If a database requires a different character set, a change can be made to use multilingualism by choosing from one of the many character sets available through MySQL. For this example, select **Standard Character Set**. Click **Next** (Figure 3-22).



Figure 3-20 Concurrent connections

© Cengage Learning 2012



Figure 3-21 Networking options

© Cengage Learning 2012

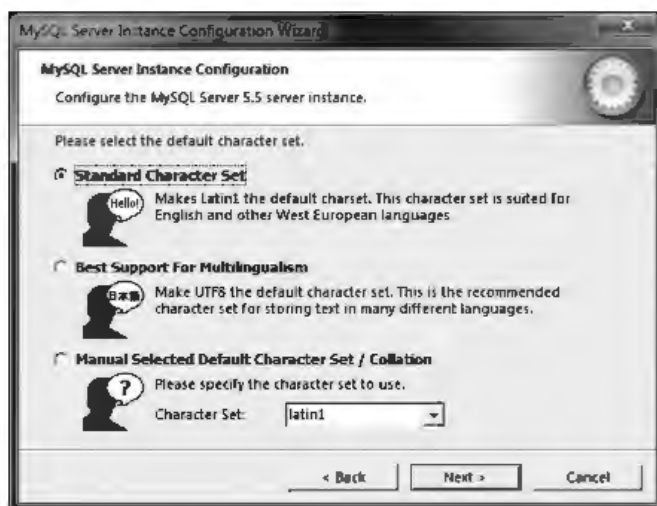


Figure 3-22 Default character set

© Cengage Learning 2012

9. The Configuration Wizard automatically creates a service called MySQL (using mysqld) that launches with the start of the machine. From the following screen, users can change the name of the service, disable the service from installing, and disable the service from starting with the machine. It is suggested that MySQL be installed as a service and automatically initiated at startup. Click **Install As Windows Service and Launch the MySQL Server automatically** (Figure 3-23). Click **Next**.
10. To maintain the security of your database server, it is important that the user password is created for the root user, that remote access is denied, and that anonymous accounts are disabled. The root user has full privileges on a MySQL server. Skipping this setting could provide intruders with easy access to the most critical components of your database. Anonymous accounts are often mistakenly enabled on a system, therefore giving public access to the system without requiring a username and password. This provides an open door into the database for unwanted intruders. Root access allows administrative access to the database server. Unless absolutely necessary, root access should not be given from outside remote machines. This would provide an opportunity for a remote intruder to take full control of your server. Therefore, always ensure that remote route access and anonymous accounts are disabled and that the root password is carefully considered. Assign a strong password to your root account (Figure 3-24). Once the security options are in place, click **Next**.
11. The confirmation screen allows a user to save the options to the configuration file (my.ini). Click **Execute** on the confirmation screen to save the options to the configurations (Figure 3-25).
12. Now that the my.ini file has been created and configured and the root password and security options have been applied, click **Finish**.



Figure 3-23 Set Windows options

© Cengage Learning 2012



Figure 3-24 Set security options

© Cengage Learning 2012

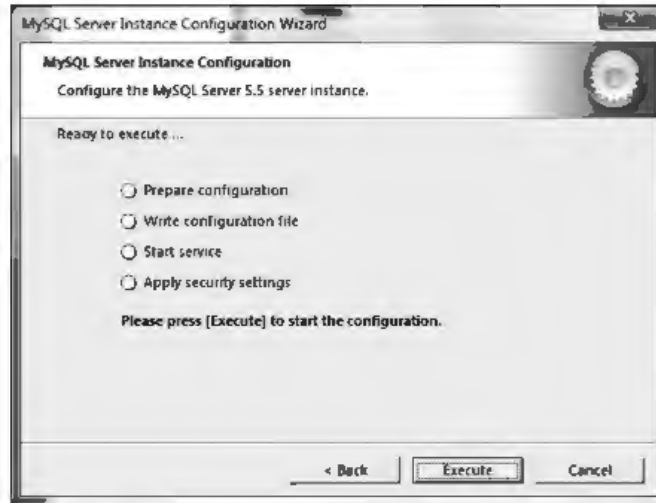


Figure 3-25 Ready to execute

© Cengage Learning 2012

Configuring MySQL on UNIX

As with Windows, UNIX-based machines require a bit of initial configuration after MySQL is installed. The configuration file for these machines is named `my.cnf`. Just as with Windows, the data directory and file permissions need to be created in order to prepare MySQL for startup. This section describes how to create the necessary initial components of a MySQL database.

Using a Configuration Script

A script called `mysql_install_db` creates the basic file directory and grant tables for user permissions automatically. In Solaris and RMP GNU/Linux installations (as with the one shown earlier in this chapter), this script is included in the installation package and therefore is run automatically. To run this script manually, navigate to the installation directory and run the script. The name of this directory depends on the way you installed MySQL:

```
For binary:
CD /usr/local/sysql
/scripts/mysql_install_db
For source installations:
CD /usr/local/sysql-5.5.1
scripts/mysql_install_db
```

Now that a data directory and initial database are in place, view the data directory by executing the following command:

```
Ls -la /path/to/datadir/mysql
```

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Setting Passwords

After the `mysql_install_db` file has completed, the server prompts the user to complete other tasks, including the option to run `mysqladmin`:

```
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 this_host_ password password 'new-
password'
```

3

Just as in Windows, all initial accounts are set up without passwords and `mysqladmin` is a program that changes the root password. The configuration script `mysql_install_db` sets up five different user accounts. There are three root accounts: `root@localhost`, `root@12.0.0.1`, and `root@hostname`; and two anonymous accounts: `'@localhost` and a second account that is named the same as the server (`'@name_of_host`). Each of the root accounts should be given strong passwords, and it is suggested that anonymous accounts be removed from the machine. As just mentioned in the Windows configuration section, anonymous accounts do not require unique usernames or passwords. Keeping these accounts could lead to a security breach. To remove these accounts, execute the following command from a root account:

```
DROP USER ''@localhost;
```

To find all of the anonymous users, issue the following query:

```
SELECT user.host.password FROM mysql.user WHERE user='';
```



NOTE

For added security, it is recommended that the usernames of all root accounts for both Windows and UNIX-based machines be changed, as well as passwords. Changing the usernames for these accounts will further strengthen their protection.



TIP

To locate and display all databases, accounts, and access to individual database, use the following commands:

```
mysql> SHOW DATABASES;
mysql> SELECT User, Host, Password FROM mysql.user;mysql> SELECT
Host, Db, User, Select_priv FROM mysql.db;
```

Additional Security Suggestions

As mentioned throughout this chapter, MySQL is an immensely popular database application, maintaining users that vary from personal PCs to enterprise organizations. Its great popularity puts it at high risk for hacker intrusions and malware attacks. Being steadfast in security practices is important to maintaining the privacy and integrity of information networks. An attack against an organization's database can be quite damaging to its reputation and overall business success. This section addresses the basic security concerns in relation to the installation and early administration of MySQL.

Passwords

As mentioned throughout the chapter, root passwords are left blank by default. Root passwords allow execution of every command available in MySQL; therefore, they should be changed and replaced with strong passwords.

Usernames can be just as important as passwords. Default root usernames as installed on all operating systems are easy to find online. Changing the usernames of root passwords can provide additional security against intruders attempting to discover access.

Never store passwords in plain text format. When a password is stored within the MySQL database, apply strong encryption techniques. If a machine becomes compromised and the password stored is in plain text, the intruder can take full advantage and further compromise a site.

Account Access and User Privileges

MySQL provides great power and control over user privileges. Take advantage of this control. Always follow the principle of least privileges, ensuring the protection of sensitive data. For example, do not share root access. If an individual needs special access, provide privileges based on need.

Remove or disable all anonymous accounts that reside on the system. Anonymous accounts do not require credentials, providing easy access into the database infrastructure.

Implement different logins for each application. If an intruder breaches the system, having different logins for each application can isolate and minimize the damage.

Network Connection Administration

Database administrators often overlook network connections when creating a security plan to secure a database server. Network connections interface with the outside world. Therefore, it is vital for administrators to take the necessary precautions for maintaining privacy. The following are best practices for protecting network connections:

- *Disable remote access*—If not possible, secure the MySQL server behind a firewall. Remote access provides a window for intruders to access the application and acts as an ideal way to implement denial of service attacks.
- *Do not leave your ports wide open*—Although this chapter instructs the reader to open port 3306 on the Windows firewall, it is necessary to restrict this port to only those hosts with permission to access it. Another option is to close the port altogether, only opening it when needed. Ports are the intruder's preferred access to any network.
- *Use IP addresses to restrict access to the database*—MySQL provides administrators with the tools to define allowed access based on IP addresses. This will ensure that only approved persons can obtain information from the database.
- *Encrypt your connection to the server using SSH or SSL*—By default, MySQL is not encrypted, which can lead to communication being intercepted and your data compromised.

Chapter Summary

- Prior to installing MySQL, an administrator must determine which distribution format, version, and edition of MySQL to install.

- MySQL is an open-source application under the GNU public license. Users are encouraged to download, test, and modify the actual source code of the software for free.
- MySQL offers a great number of resources as extensions to its main Web site, including MySQL manuals, product downloads, training and technical support, bug and fix tools, and developer information.
- There are two different ways you can install MySQL: using its prepackaged binary files or by installing and compiling the source code manually.
- Different versions of MySQL are available at different stages of its development. One can download a version of MySQL in its beta form (while it is still being tested and debugged), as a release candidate (somewhat stable and tested), or as a general-availability product (fully tested and debugged).
- MySQL has adopted a milestone model, which improves the release cycle process and allows users to better predict any new versions that will be available. The milestone model dictates that new RC releases will be available every 3–6 months, while new GA releases will be available approximately every 18 months.
- Help with MySQL comes in many different forms. Besides those channels available on MySQL's Web site, users can find help within forums, from bloggers, on Twitter, and through mailing lists.
- The installation packages for Windows operating systems offer wizards to aid users in installing and configuring MySQL.
- UNIX-based servers offer several different platform-dependent installation packages from which to choose.
- Installing MySQL using the RPM binary is the suggested mode of installation for all Linux servers that use glibc2.3 library.
- MySQL can configure a machine to be either a developer machine, a server machine, or a dedicated MySQL server machine.
- Anonymous accounts are created in both Windows and UNIX. These accounts should be removed from the system as an extra security precaution.
- Passwords stored within the database should be encrypted to ensure their safety in case of a breach.
- The principle of least privilege should be followed when choosing access and privilege rights of users within the MySQL database.
- It is important for database administrators to use best practices in the process of protecting network connections.

Key Terms

binary code installations Binary files that are packaged and ready to be installed without the need for compiling the code to enable it to be run as an executable file on a particular machine.

binary file A file that contains code that can be read by machines and run as an executable file.

compile The act of converting source code that is written in one language into a different programming language or machine language.

digital signature Code that uses cryptography to verify the authenticity of a source of information.

programming language A type of synthetic language developed with a specific syntax and semantic rules that allows individuals to create statements or functions to interface and control the behavior or functionality of a machine.

source code A group of statements or functions written using a specific programming language and that are combined to create a specific type of application or utility.

source code installation Allows a user to download the actual MySQL source code, change it, and compile it into a binary file for installation execution.

Review Questions

1. List the considerations a network administrator should take prior to database installation.
2. Identify two main installation formats that can be used to install MySQL.
3. Explain why installing a source code installation format can be time consuming.
4. Explain the importance of reviewing the available versions of a database application prior to installation.
5. Identify and define the four phases of maturity for a release of MySQL.
6. Identify the main goals of the milestone model.
7. What is the recommended version of MySQL to install for the first time?
8. List six online sources of help for MySQL.
9. Explain the main difference between the two different editions of MySQL available for downloading and installation.
10. List two ways to verify that a package has not been tampered with and is clean of Trojans.
11. List two security practices employed regarding root passwords. Failure to employ these practices can result in what consequences?
12. List five practices to follow to ensure the protection of sensitive data.
13. List four best practices for the protection of network connections.

Case Projects



Case Project 3-1: MySQL Bugs

Search MySQL's bug home page and compile a list of three known MySQL bugs for both UNIX-based and Windows platforms.

Case Project 3-2: MySQL Releases

Using MySQL's download page, list all available binary versions of MySQL and include their filenames. Identify the most current release of MySQL.

Case Project 3-3: UNIX MySQL Script

Using the Internet as a resource, research and define the `mysql_secure_installation` UNIX MySQL script. Include the security-related tasks that it performs.

3

Case Project 3-4: Researching Changelog

Using the Internet as a resource, research the changelog pages of MySQL. Identify the content that can be found within them and describe the reason you would use these pages.

Case Project 3-5: Implementing Security Measures

Suppose that you are a security-conscious database manager for Haphazard, Inc. Outline the security measures that you would implement to protect a new installation of MySQL.

Hands-On Projects



Hands-On Project 3-1: Install an Archive Package on a UNIX-based Server

Haphazard, Inc. has hired you as a database administrator for their organization. You are instructed to install MySQL on their existing UNIX-based server using a noinstall archive package. Identify the steps that are necessary to do the following:

1. Download the file
2. Install MySQL
3. Configure the server
4. Apply necessary security

Hands-On Project 3-2: Install an Archive Package on a Windows Server

Haphazard, Inc. has hired you as a database administrator for their organization. You are instructed to install MySQL on their existing Windows server using a noinstall archive package. Identify the steps that are necessary to do the following:

1. Download the file
2. Install MySQL
3. Configure the server
4. Apply necessary security