MySQL Installation Guide
Abstract

This is the MySQL Installation Guide from the MySQL 5.1 Reference Manual.

For legal information, see the Legal Notices.

For help with using MySQL, please visit either the MySQL Forums or MySQL Mailing Lists, where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Document generated on: 2016-01-11 (revision: 46052)
# Table of Contents

Preface and Legal Notices .......................................................... vii
1 Installing and Upgrading MySQL .................................................. 1
2 General Installation Guidance ...................................................... 3
   2.1 Choosing Which MySQL Distribution to Install .............................. 3
      2.1.1 Choosing Which Version of MySQL to Install ........................ 3
      2.1.2 Choosing a Distribution Format ........................................ 5
      2.1.3 How and When Updates Are Released ................................... 6
   2.2 How to Get MySQL .............................................................. 6
   2.3 Verifying Package Integrity Using MD5 Checksums or GnuPG ............ 7
      2.3.1 Verifying the MD5 Checksum ............................................. 7
      2.3.2 Signature Checking Using GnuPG ........................................ 7
      2.3.3 Signature Checking Using Gpg4win for Windows ......................... 10
      2.3.4 Signature Checking Using RPM .......................................... 15
   2.4 Installation Layouts ............................................................ 16
   2.5 Compiler-Specific Build Characteristics .................................... 16
3 Installing MySQL on Unix/Linux Using Generic Binaries ....................... 17
4 Installing MySQL from Source .................................................... 21
   4.1 MySQL Layout for Source Installation ....................................... 22
   4.2 Installing MySQL Using a Standard Source Distribution .................. 22
   4.3 Installing MySQL Using a Development Source Tree ....................... 25
   4.4 MySQL Source-Configuration Options ....................................... 27
   4.5 Dealing with Problems Compiling MySQL .................................... 37
   4.6 Compiling and Linking an Optimized mysqld Server ......................... 40
   4.7 Installing MySQL from Source on Windows .................................. 41
   4.8 Notes on Installing MySQL on Solaris from Source ......................... 45
   4.9 Notes on Installing MySQL on AIX from Source ............................ 47
   4.10 Notes on Installing MySQL on HP-UX from Source ......................... 47
5 Installing MySQL on Microsoft Windows ....................................... 49
   5.1 MySQL Installation Layout on Microsoft Windows .......................... 51
   5.2 Choosing the Installation Package for Microsoft Windows ................. 52
   5.3 MySQL Notifier ..................................................................... 54
      5.3.1 Remote monitoring set up and installation instructions .............. 61
   5.4 Installing MySQL on Microsoft Windows Using an MSI Package .......... 66
      5.4.1 Using the MySQL Installation Wizard for Microsoft Windows ....... 68
      5.4.2 Automating MySQL Installation on Microsoft Windows Using the MSI Package .................................................. 71
      5.4.3 Removing MySQL When Installed from the MSI Package .............. 72
   5.5 Using the MySQL Server Instance Config Wizard ........................... 73
      5.5.1 Starting the MySQL Server Instance Config Wizard .................... 75
      5.5.2 MySQL Server Instance Config Wizard: Choosing a Maintenance Option .......................................................... 76
      5.5.3 MySQL Server Instance Config Wizard: Choosing a Configuration Type .......................................................... 76
      5.5.4 MySQL Server Instance Config Wizard: The Server Type Dialog .......... 77
      5.5.5 MySQL Server Instance Config Wizard: The Database Usage Dialog .......................................................... 79
      5.5.6 MySQL Server Instance Config Wizard: The InnoDB Tablespace Dialog .......................................................... 80
      5.5.7 MySQL Server Instance Config Wizard: The Concurrent Connections Dialog .......................................................... 81
      5.5.8 MySQL Server Instance Config Wizard: The Networking and Strict Mode Options Dialog .......................................................... 82
      5.5.9 MySQL Server Instance Config Wizard: The Character Set Dialog .......... 83
      5.5.10 MySQL Server Instance Config Wizard: The Service Options Dialog .......................................................... 83
      5.5.11 MySQL Server Instance Config Wizard: The Security Options Dialog .......................................................... 84
      5.5.12 MySQL Server Instance Config Wizard: The Confirmation Dialog .......................................................... 86
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 Perl Installation Notes</td>
<td>185</td>
</tr>
<tr>
<td>11.2 Downgrading MySQL</td>
<td>172</td>
</tr>
<tr>
<td>11.1.1 Changes Affecting Ugrades to 5.1</td>
<td>157</td>
</tr>
<tr>
<td>11.1 Upgrading MySQL</td>
<td>157</td>
</tr>
<tr>
<td>11.1.1 Changes Affecting Ugrades to 5.1</td>
<td>161</td>
</tr>
<tr>
<td>11.1 Downgrading MySQL</td>
<td>172</td>
</tr>
<tr>
<td>11.1.2 Downgrading from MySQL Enterprise Edition to MySQL Community Server</td>
<td>176</td>
</tr>
<tr>
<td>11.3 Checking Whether Tables or Indexes Must Be Rebuilt</td>
<td>177</td>
</tr>
<tr>
<td>11.4 Rebuilding or Repairing Tables or Indexes</td>
<td>179</td>
</tr>
<tr>
<td>11.5 Copying MySQL Databases to Another Machine</td>
<td>180</td>
</tr>
<tr>
<td>12 Environment Variables</td>
<td>183</td>
</tr>
<tr>
<td>13 Perl Installation Notes</td>
<td>185</td>
</tr>
<tr>
<td>13.1 Installing Perl on Unix</td>
<td>185</td>
</tr>
<tr>
<td>13.2 Installing ActiveState Perl on Windows</td>
<td>186</td>
</tr>
<tr>
<td>13.3 Problems Using the Perl DBI/DBD Interface</td>
<td>187</td>
</tr>
<tr>
<td>A Licenses for Third-Party Components</td>
<td>189</td>
</tr>
<tr>
<td>A.1 ANTLR 3 License</td>
<td>191</td>
</tr>
<tr>
<td>A.2 dtoa.c License</td>
<td>191</td>
</tr>
<tr>
<td>A.3 Editline Library (libedit) License</td>
<td>192</td>
</tr>
</tbody>
</table>
MySQL Installation Guide

A.4 FindGTest.cmake License ................................................................. 194
A.5 Fred Fish's Dbug Library License ..................................................... 195
A.6 getarg License .................................................................................. 196
A.7 GNU General Public License Version 2.0, June 1991 ............................. 196
A.8 GNU Lesser General Public License Version 2.1, February 1999 ............. 202
A.9 GNU Libtool License ........................................................................... 210
A.10 GNU Readline License ................................................................. 210
A.11 Google Controlling Master Thread I/O Rate Patch License ................... 211
A.12 Google Perftools (TCMalloc utility) License ....................................... 211
A.13 Google SMP Patch License .............................................................. 212
A.14 lib_sql.cc License ............................................................................ 213
A.15 libevent License ................................................................................ 213
A.16 Linux-PAM License .......................................................................... 215
A.17 md5 (Message-Digest Algorithm 5) License ....................................... 216
A.18 memcached License .......................................................................... 216
A.19 nt_servc (Windows NT Service class library) License .......................... 217
A.20 OpenPAM License ............................................................................ 217
A.21 Paramiko License ............................................................................... 217
A.22 Percona Multiple I/O Threads Patch License ...................................... 218
A.23 RegEX-Spencer Library License ......................................................... 218
A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License ................... 219
A.25 Richard A. O'Keefe String Library License ........................................ 219
A.26 SHA-1 in C License .......................................................................... 220
A.27 zlib License ..................................................................................... 220
Preface and Legal Notices

This is the MySQL Installation Guide from the MySQL 5.1 Reference Manual.

Legal Notices

Copyright © 1997, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.
End of Product Lifecycle. Active development for MySQL Database Server version 5.1 has ended. Oracle offers various support offerings which may be of interest. For details and more information, see the MySQL section of the Lifetime Support Policy for Oracle Technology Products (http://www.oracle.com/us/support/lifetime-support/index.html). Please consider upgrading to a recent version.

This chapter describes how to obtain and install MySQL. A summary of the procedure follows and later sections provide the details. If you plan to upgrade an existing version of MySQL to a newer version rather than install MySQL for the first time, see Section 11.1, “Upgrading MySQL”, for information about upgrade procedures and about issues that you should consider before upgrading.

If you are interested in migrating to MySQL from another database system, see MySQL 5.1 FAQ: Migration, which contains answers to some common questions concerning migration issues.

If you are migrating from MySQL Enterprise Edition to MySQL Community Server, see Section 11.2.2, “Downgrading from MySQL Enterprise Edition to MySQL Community Server”.

Installation of MySQL generally follows the steps outlined here:

1. **Determine whether MySQL runs and is supported on your platform.**

   Please note that not all platforms are equally suitable for running MySQL, and that not all platforms on which MySQL is known to run are officially supported by Oracle Corporation.

2. **Choose which distribution to install.**

   Several versions of MySQL are available, and most are available in several distribution formats. You can choose from pre-packaged distributions containing binary (precompiled) programs or source code. When in doubt, use a binary distribution. Oracle also provides access to the MySQL source code for those who want to see recent developments and test new code. To determine which version and type of distribution you should use, see Section 2.1, “Choosing Which MySQL Distribution to Install”.

3. **Download the distribution that you want to install.**

   For instructions, see Section 2.2, “How to Get MySQL”. To verify the integrity of the distribution, use the instructions in Section 2.3, “Verifying Package Integrity Using MD5 Checksums or GnuPG”.

4. **Install the distribution.**

   To install MySQL from a binary distribution, use the instructions in Chapter 3, *Installing MySQL on Unix/Linux Using Generic Binaries*.

   To install MySQL from a source distribution or from the current development source tree, use the instructions in Chapter 4, * Installing MySQL from Source*.

5. **Perform any necessary postinstallation setup.**

   After installing MySQL, see Chapter 10, *Postinstallation Setup and Testing* for information about making sure the MySQL server is working properly. Also refer to the information provided in Section 10.4, “Securing the Initial MySQL Accounts”. This section describes how to secure the initial MySQL user accounts, which have no passwords until you assign passwords. The section applies whether you install MySQL using a binary or source distribution.

6. **If you want to run the MySQL benchmark scripts, Perl support for MySQL must be available.** See Chapter 13, *Perl Installation Notes*. 
Instructions for installing MySQL on different platforms and environments is available on a platform by platform basis:

- **Unix, Linux, FreeBSD**

  For instructions on installing MySQL on most Linux and Unix platforms using a generic binary (for example, a `.tar.gz` package), see Chapter 3, *Installing MySQL on Unix/Linux Using Generic Binaries*.

  For information on building MySQL entirely from the source code distributions or the source code repositories, see Chapter 4, *Installing MySQL from Source*.

  For specific platform help on installation, configuration, and building from source see the corresponding platform section:

  - Linux, including notes on distribution specific methods, see Chapter 7, *Installing MySQL on Linux*.
  - Solaris and OpenSolaris, including PKG and IPS formats, see Chapter 8, *Installing MySQL on Solaris and OpenSolaris*.
  - IBM AIX, see Chapter 8, *Installing MySQL on Solaris and OpenSolaris*.
  - Hewlett-Packard HP-UX, including the DEPOT package format, see *Installing MySQL on HP-UX*.
  - FreeBSD, see *Installing MySQL on FreeBSD*.

- **Microsoft Windows**

  For instructions on installing MySQL on Microsoft Windows, using either a Zipped binary or an MSI package, see Chapter 5, *Installing MySQL on Microsoft Windows*.

  For information on using the MySQL Server Instance Config Wizard, see Section 5.5, “Using the MySQL Server Instance Config Wizard”.

  For details and instructions on building MySQL from source code using Microsoft Visual Studio, see Section 4.7, “Installing MySQL from Source on Windows”.

- **OS X**

  For installation on OS X, including using both the binary package and native PKG formats, see Chapter 6, *Installing MySQL on OS X*.

  For information on making use of the MySQL Startup Item to automatically start and stop MySQL, see Section 6.4, “Installing the MySQL Startup Item”.

  For information on the MySQL Preference Pane, see Section 6.5, “Installing and Using the MySQL Preference Pane”.

- **IBM i5/OS**

  For instructions on installing, starting, and stopping MySQL on i5/OS, see Chapter 9, *Installing MySQL on i5/OS*.
Chapter 2 General Installation Guidance

Table of Contents

2.1 Choosing Which MySQL Distribution to Install ................................................................. 3
  2.1.1 Choosing Which Version of MySQL to Install .............................................................. 3
  2.1.2 Choosing a Distribution Format ....................................................................................... 5
  2.1.3 How and When Updates Are Released ............................................................................. 6
2.2 How to Get MySQL .................................................................................................................. 6
2.3 Verifying Package Integrity Using MD5 Checksums or GnuPG ......................................... 7
  2.3.1 Verifying the MD5 Checksum ......................................................................................... 7
  2.3.2 Signature Checking Using GnuPG .................................................................................... 7
  2.3.3 Signature Checking Using Gpg4win for Windows ......................................................... 10
  2.3.4 Signature Checking Using RPM ..................................................................................... 15
2.4 Installation Layouts .................................................................................................................. 16
2.5 Compiler-Specific Build Characteristics .................................................................................. 16

The immediately following sections contain the information necessary to choose, download, and verify your distribution. The instructions in later sections of the chapter describe how to install the distribution that you choose. For binary distributions, see the instructions at Chapter 3, *Installing MySQL on Unix/Linux Using Generic Binaries* or the corresponding section for your platform if available. To build MySQL from source, use the instructions in Chapter 4, *Installing MySQL from Source*.

2.1 Choosing Which MySQL Distribution to Install

MySQL is available on a number of operating systems and platforms. For information about those platforms that are officially supported, see [http://www.mysql.com/support/supportedplatforms/database.html](http://www.mysql.com/support/supportedplatforms/database.html) on the MySQL Web site.

When preparing to install MySQL, you should decide which version to use. MySQL development occurs in several release series, and you can pick the one that best fits your needs. After deciding which version to install, you can choose a distribution format. Releases are available in binary or source format.

2.1.1 Choosing Which Version of MySQL to Install

The first decision to make is whether you want to use a production (stable) release or a development release. In the MySQL development process, multiple release series co-exist, each at a different stage of maturity.

**Production Releases**

- MySQL 5.6: Latest General Availability (Production) release
- MySQL 5.5: Previous General Availability (Production) release
- MySQL 5.1: Older General Availability (Production) release
- MySQL 5.0: Older Production release nearing the end of the product lifecycle

MySQL 4.1, 4.0, and 3.23 are old releases that are no longer supported.


Normally, if you are beginning to use MySQL for the first time or trying to port it to some system for which there is no binary distribution, use the most recent General Availability series listed in the preceding
Choosing Which Version of MySQL to Install

descriptions. All MySQL releases, even those from development series, are checked with the MySQL benchmarks and an extensive test suite before being issued.

If you are running an older system and want to upgrade, but do not want to take the chance of having a nonseamless upgrade, you should upgrade to the latest version in the same release series you are using (where only the last part of the version number is newer than yours). We have tried to fix only fatal bugs and make only small, relatively “safe” changes to that version.

If you want to use new features not present in the production release series, you can use a version from a development series. Be aware that development releases are not as stable as production releases.

We do not use a complete code freeze because this prevents us from making bugfixes and other fixes that must be done. We may add small things that should not affect anything that currently works in a production release. Naturally, relevant bugfixes from an earlier series propagate to later series.

If you want to use the very latest sources containing all current patches and bugfixes, you can use one of our source code repositories (see Section 4.3, “Installing MySQL Using a Development Source Tree”). These are not “releases” as such, but are available as previews of the code on which future releases are to be based.

The naming scheme in MySQL 5.1 uses release names that consist of three numbers and a suffix; for example, mysql-5.1.29-rc. The numbers within the release name are interpreted as follows:

- The first number (5) is the major version and describes the file format. All MySQL 5 releases have the same file format.
- The second number (1) is the release level. Taken together, the major version and release level constitute the release series number.
- The third number (29) is the version number within the release series. This is incremented for each new release. Usually you want the latest version for the series you have chosen.

For each minor update, the last number in the version string is incremented. When there are major new features or minor incompatibilities with previous versions, the second number in the version string is incremented. When the file format changes, the first number is increased.

Release names also include a suffix that indicates the stability level of the release. Releases within a series progress through a set of suffixes to indicate how the stability level improves. The possible suffixes are:

- **alpha** indicates that the release is for preview purposes only. Known bugs should be documented in the Release Notes. Most alpha releases implement new commands and extensions. Active development that may involve major code changes can occur in an alpha release. However, we do conduct testing before issuing a release.
- **beta** indicates that the release is appropriate for use with new development. Within beta releases, the features and compatibility should remain consistent. However, beta releases may contain numerous and major unaddressed bugs.

  All APIs, externally visible structures, and columns for SQL statements will not change during future beta, release candidate, or production releases.

- **rc** indicates a Release Candidate. Release candidates are believed to be stable, having passed all of MySQL's internal testing, and with all known fatal runtime bugs fixed. However, the release has not been in widespread use long enough to know for sure that all bugs have been identified. Only minor fixes are added. (A release candidate is what formerly was known as a gamma release.)
• If there is no suffix, it indicates that the release is a General Availability (GA) or Production release. GA releases are stable, having successfully passed through all earlier release stages and are believed to be reliable, free of serious bugs, and suitable for use in production systems. Only critical bugfixes are applied to the release.

All releases of MySQL are run through our standard tests and benchmarks to ensure that they are relatively safe to use. Because the standard tests are extended over time to check for all previously found bugs, the test suite keeps getting better.

All releases have been tested at least with these tools:

• **An internal test suite.** The mysql-test directory contains an extensive set of test cases. We run these tests for every server binary. See The MySQL Test Suite, for more information about this test suite.

• **The MySQL benchmark suite.** This suite runs a range of common queries. It is also a test to determine whether the latest batch of optimizations actually made the code faster. See The MySQL Benchmark Suite.

We also perform additional integration and nonfunctional testing of the latest MySQL version in our internal production environment. Integration testing is done with different connectors, storage engines, replication modes, backup, partitioning, stored programs, and so forth in various combinations. Additional nonfunctional testing is done in areas of performance, concurrency, stress, high volume, upgrade and downgrade.

### 2.1.2 Choosing a Distribution Format

After choosing which version of MySQL to install, you should decide whether to use a binary distribution or a source distribution. In most cases, you should probably use a binary distribution, if one exists for your platform. Binary distributions are available in native format for many platforms, such as RPM packages for Linux, DMG packages for OS X, and PKG packages for Solaris. Distributions are also available in more generic formats such as Zip archives or compressed tar files.

Reasons to choose a binary distribution include the following:

• Binary distributions generally are easier to install than source distributions.

• To satisfy different user requirements, we provide several servers in binary distributions. mysqld is an optimized server that is a smaller, faster binary. mysqld-debug is compiled with debugging support.

  Each of these servers is compiled from the same source distribution, though with different configuration options. All native MySQL clients can connect to servers from either MySQL version.

Under some circumstances, you may be better off installing MySQL from a source distribution:

• You want to install MySQL at some explicit location. The standard binary distributions are ready to run at any installation location, but you might require even more flexibility to place MySQL components where you want.

• You want to configure mysqld to ensure that features are available that might not be included in the standard binary distributions. Here is a list of the most common extra options that you may want to use to ensure feature availability:

  • --with-libwrap
  • --with-named-z-libs (this is done for some of the binaries)
  • --with-debug[=full]
How and When Updates Are Released

For additional information, see Section 4.4, “MySQL Source-Configuration Options”.

• You want to configure mysqld without some features that are included in the standard binary distributions. For example, distributions normally are compiled with support for all character sets. If you want a smaller MySQL server, you can recompile it with support for only the character sets you need.

• You want to use the latest sources from one of the Bazaar repositories to have access to all current bugfixes. For example, if you have found a bug and reported it to the MySQL development team, the bugfix is committed to the source repository and you can access it there. The bugfix does not appear in a release until a release actually is issued.

• You want to read (or modify) the C and C++ code that makes up MySQL. For this purpose, you should get a source distribution, because the source code is always the ultimate manual.

• Source distributions contain more tests and examples than binary distributions.

2.1.3 How and When Updates Are Released

MySQL is evolving quite rapidly and we want to share new developments with other MySQL users. We try to produce a new release whenever we have new and useful features that others also seem to have a need for.

We also try to help users who request features that are easy to implement. We take note of what our licensed users want, and we especially take note of what our support customers want and try to help them in this regard.

No one is required to download a new release. The Release Notes help you determine whether the new release has something you really want.

We use the following policy when updating MySQL:

• Enterprise Server releases are meant to appear every 18 months, supplemented by quarterly service packs and monthly rapid updates. Community Server releases are meant to appear 2 to 3 times per year.

• Releases are issued within each series. For each release, the last number in the version is one more than the previous release within the same series.

• Binary distributions for some platforms are made by us for major releases. Other people may make binary distributions for other systems, but probably less frequently.

• We make fixes available as soon as we have identified and corrected small or noncritical but annoying bugs. The fixes are available in source form immediately from our public Bazaar repositories, and are included in the next release.

• If by any chance a security vulnerability or critical bug is found in a release, our policy is to fix it in a new release as soon as possible. (We would like other companies to do this, too!)

2.2 How to Get MySQL

Check our downloads page at http://dev.mysql.com/downloads/ for information about the current version of MySQL and for downloading instructions. For a complete up-to-date list of MySQL download mirror sites, see http://dev.mysql.com/downloads/mirrors.html. You can also find information there about becoming a MySQL mirror site and how to report a bad or out-of-date mirror.

To obtain the latest development source, see Section 4.3, “Installing MySQL Using a Development Source Tree”.
2.3 Verifying Package Integrity Using MD5 Checksums or GnuPG

After you have downloaded the MySQL package that suits your needs and before you attempt to install it, you should make sure that it is intact and has not been tampered with. There are three means of integrity checking:

- MD5 checksums
- Cryptographic signatures using GnuPG, the GNU Privacy Guard
- For RPM packages, the built-in RPM integrity verification mechanism

The following sections describe how to use these methods.

If you notice that the MD5 checksum or GPG signatures do not match, first try to download the respective package one more time, perhaps from another mirror site.

2.3.1 Verifying the MD5 Checksum

After you have downloaded a MySQL package, you should make sure that its MD5 checksum matches the one provided on the MySQL download pages. Each package has an individual checksum that you can verify against the package that you downloaded. The correct MD5 checksum is listed on the downloads page for each MySQL product, and you will compare it against the MD5 checksum of the file (product) that you download.

Each operating system and setup offers its own version of tools for checking the MD5 checksum. Typically the command is named `md5sum`, or it may be named `md5`, and some operating systems do not ship it at all. On Linux, it is part of the GNU Text Utilities package, which is available for a wide range of platforms. You can also download the source code from http://www.gnu.org/software/textutils/. If you have OpenSSL installed, you can use the command `openssl md5 package_name` instead. A Windows implementation of the `md5` command line utility is available from http://www.fourmilab.ch/md5/. `winMd5Sum` is a graphical MD5 checking tool that can be obtained from http://www.nullriver.com/index/products/winmd5sum. Our Microsoft Windows examples will assume the name `md5.exe`.

Linux and Microsoft Windows examples:

```
shell> md5sum mysql-standard-5.1.73-linux-i686.tar.gz
aaab65abbec64d5e907dcd41b8699945  mysql-standard-5.1.73-linux-i686.tar.gz

shell> md5.exe mysql-installer-community-5.1.73.msi
aaab65abbec64d5e907dcd41b8699945  mysql-installer-community-5.1.73.msi
```

You should verify that the resulting checksum (the string of hexadecimal digits) matches the one displayed on the download page immediately below the respective package.

**Note**

Make sure to verify the checksum of the archive file (for example, the `.zip`, `.tar.gz`, or `.msi` file) and not of the files that are contained inside of the archive. In other words, verify the file before extracting its contents.

2.3.2 Signature Checking Using GnuPG

Another method of verifying the integrity and authenticity of a package is to use cryptographic signatures. This is more reliable than using MD5 checksums, but requires more work.
We sign MySQL downloadable packages with GnuPG (GNU Privacy Guard). GnuPG is an Open Source alternative to the well-known Pretty Good Privacy (PGP) by Phil Zimmermann. See http://www.gnupg.org/ for more information about GnuPG and how to obtain and install it on your system. Most Linux distributions ship with GnuPG installed by default. For more information about GnuPG, see http://www.openpgp.org/.

To verify the signature for a specific package, you first need to obtain a copy of our public GPG build key, which you can download from http://pgp.mit.edu/. The key that you want to obtain is named mysql-build@oss.oracle.com. Alternatively, you can cut and paste the key directly from the following text:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.9 (SunOS)
mQGiBD4+owwRBAC14GIfUTcYEDESEvEw3SAfUdJbtcoQJH/i/jKJqyQT9h9pblUW3 CRODQjReyCTIrrdwvYKGuJK2fMeVGoW2uZwMDNABLNprwRWDk9c6+oMSLN9brz f2w2VoQgCMvZ2w80YH0yDHuV1Qa/BThQQoAdqj8AN6/0L7V1N98/Vu7H0qQgCvqVz3 BqXQrXnNCXCRXAvuAuzUTzHRrE3AoqQk1+iLs1nZvMDl1WufeXfshc57+/ye3KegNW hxfv96p9WRVhNYJdRT++r2ZUeu3vpxqKQNU/hJeUIJHRQYfio8g7TxxvXMWc7f7JLYV K2hTrkPbP7/2wbasE8MXYhrh0eKehTlgs9rj6AsQgAcY/Yb7h1wpxh6Dz1TTnE kyXpBzACmUpMPN7KNкеmNdcPia2zOQOHADNwpUK43l7JdsMqmtobZX9qranxw+/uNDI QJXEM6FbL0ILnci1zNiHsaPrVEOMDVkpmKQ6/k1IstNtPvaidl8lH0bPAmUqyefep rv0xaxqEGMCm3o7wWQfN83POKdsZaB5p3jwOjxvzv6/62zJQ72XT1UUwqUmVs ZWfZ2SBfPbdpmv1cmLuUyABsIzCnW1yvPbgRABA1n3m9YWaNzS25j5b0+j1GKe ExEACkCcGygMCwkJWbMBBUCMAEMFpGd1Qae1QxGQxIAIZCQWUHJQUBGmbXyAR CRMCtC70U1H3v9/ADACkJSjQgwV1/eut+5+l2v3ybl+2gcCdZoA341lhorozVJU 6AR0f9f9qeg0OQ0151UMFhBTzZuG21hnu0jyu5RZxgKhd3d35eXABs5j b2p0z1dXkm12EbnXnc5z5b0+j1GMEMBCAE8CAF53pAo0s1Bd1k3E8BeXhXN k5bC09zd2I1sCBzdK5w9hDvctmcbmgc29vbgAKCRMCWEYU0HHL/9u9J99Xx0o xJL9F19T3G4C1k9Axw/CCRsC9znCjYx9arEl0s2/Ce1jdyI2QlTECIAHQC R6yUtAIIJDBYtqGULBwdDMABAMwVDpGd1QaeIABA1JExIjcTJ9cuhiB2HUEvAcQAq kgCfzz4fGe4jxZx017l1CvCgCxtgbeAN30qLr19jCrbrWX01ms4mAbf01piEw EExEACPAFj4+o9EFgwLmALsACgKQ5gFvHlC0mGW0I9rnc39K3v0Km/k3ys1Q5SammInAuj0xwFwP1x0f8038QNGlbc0U9lWEEXECAAWAFAJ55gMfGwll0ioKa cgQxvTNCtA6d+KWCqClGmGoGjMojypnPwWKx0YkfnkA0K7E96h8wBrkSDKou 1z7NMk3h8uyIEYESBECAAYAFA+aj46ACgAcQrUJSJ3bIA/2lQg2l1iHNMBATy P/BjusF/JQL/eoAnrMwyEtUqj2/S0n0H1KeheeIAIEYESBECAAYAFA+0Q3oA cgQxvQarvqz5zB3mgWcDpQdF1frv17tCRw8GQK0s7hn3N32KAni1bwp旺盛T6aNQO Q3nk1l1a2dePhIEYESBECAAYAFAKnbv+aRcGq1qgubZ5ClX/aOdAC6lKoQjKkWNJ0 gZztsh7kjsiyGmFMNRBMHBQ7V9KQO6pJ0A0i7tqGEYCAFAAYAFAKxtMyYA cgQxK9hE9CtUuwKcqcqgibak/2SwxWHn1i1JRhgyC05cM4 uncomfortable lc5w7aciK1Gx m7nVenqU7f1/EwEEXECAAWAFAKGBywEfGwly1ZsACgKQFQgnFjRd27e0qexCQcDd8jN Ndc/m580Gd0XUV99McngKn1jYWDQ/QVQo3m1/U190EahNkncfJw1EwEEXEAFAW AfkbY2oPgwY11V2bAcGqQcL36+ITtpf11WcdFYNVBU8xe8mXoFm4pdm45FjTPmA nISPa/2fsj30CMKlarr4qFQpQrdr1iEwEEXECAAWAFAKGBy2Ifgwi21soAcoGqAs3d 2V3d9N9m3qQbCbyzrGPGXOgP8jKiebdwq3eXenAsrs42G93q7uh0TL3PSTyS HjwYXJEwBEwwEEXECAAWAFAKQkYQwFgQwwEwXEQwQkCQXECXGs6p8/d1JgXQgVxWqKpQwOws8 w7KcseCt1fwl1kAni1v1/Liq1y0LZkYn5kJk1di5fviEwEEXECAAWAFAKIr7wO qgw5SVN1ACqskq5hukIRXrav5zwEqksL5qk5ycpw9LHicFSxl/3oVA4N3nXkum g6C1CNJF8214NCWzIrq1EfEwEEXAECAWAFQwTo/fgwD6s11ACaqKQPERKNrT6+Gm eXcaAcA35sACothWqCqcigibak/2SwxWHn1i1JRhgyC05cM4 uncomfortable lc5w7aciK1Gx m7nVenqU7f1/EwEEXECAAWAFAKGBywEfGwly1ZsACgKQFQgnFjRd27e0qexCQcDd8jN Ndc/m580Gd0XUV99McngKn1jYWDQ/QVQo3m1/U190EahNkncfJw1EwEEXEAFAW AfkbY2oPgwY11V2bAcGqQcL36+ITtpf11WcdFYNVBU8xe8mXoFm4pdm45FjTPmA nISPa/2fsj30CMKlarr4qFQpQrdr1iEwEEXECAAWAFAKGBy2Ifgwi21soAcoGqAs3d 2V3d9N9m3qQbCbyzrGPGXOgP8jKiebdwq3eXenAsrs42G93q7uh0TL3PSTyS HjwYXJEwBEwwEEXECAAWAFAKQkYQwFgQwwEwXEQwQkCQXECXGs6p8/d1JgXQgVxWqKpQwOws8 w7KcseCt1fwl1kAni1v1/Liq1y0LZkYn5kJk1di5fviEwEEXECAAWAFAKIr7wO qgw5SVN1ACqskq5hukIRXrav5zwEqksL5qk5ycpw9LHicFSxl/3oVA4N3nXkum g6C1CNJF8214NCWzIrq1EfEwEEXAECAWAFQwTo/fgwD6s11ACaqKQPERKNrT6+Gm eXcaAcA35sACothWqCqcigibak/2SwxWHn1i1JRhgyC05cM4 uncomfortable lc5w7aciK1Gx m7nVenqU7f1/EwEEXECAAWAFAKGBywEfGwly1ZsACgKQFQgnFjRd27e0qexCQcDd8jN Ndc/m580Gd0XUV99McngKn1jYWDQ/QVQo3m1/U190EahNkncfJw1EwEEXEAFAW
To import the build key into your personal public GPG keyring, use `gpg --import`. For example, if you have saved the key in a file named `mysql_pubkey.asc`, the import command looks like this:

```
shell> gpg --import mysql_pubkey.asc
gpg: key 5072E1F5: public key "MySQL Release Engineering <mysql-build@oss.oracle.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
gpg: no ultimately trusted keys found
```

You can also download the key from the public keyserver using the public key id, `5072E1F5`:

```
shell> gpg --recv-keys 5072E1F5
```

You can use the imported key to validate RPM install packages by running `gpg --verify`. For example:

```
shell> gpg --verify some.rpm
```

If you want to import the key into your RPM configuration to validate RPM install packages, you should be able to import the key directly:

```
gpg: Total number processed: 1
gpg:           new user IDs: 1
gpg:         new signatures: 53
```

You should verify the key's authenticity by checking the signature to ensure it was signed by the correct entity.
If you experience problems or require RPM specific information, see Section 2.3.4, “Signature Checking Using RPM”.

After you have downloaded and imported the public build key, download your desired MySQL package and the corresponding signature, which also is available from the download page. The signature file has the same name as the distribution file with an `.asc` extension, as shown by the examples in the following table.

**Table 2.1 MySQL Package and Signature Files for Source files**

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution file</td>
<td>mysql-standard-5.1.73-linux-i686.tar.gz</td>
</tr>
<tr>
<td>Signature file</td>
<td>mysql-standard-5.1.73-linux-i686.tar.gz.asc</td>
</tr>
</tbody>
</table>

Make sure that both files are stored in the same directory and then run the following command to verify the signature for the distribution file:

```
shell> gpg --verify package_name.asc
```

If the downloaded package is valid, you will see a "Good signature" similar to:

```
shell> gpg --verify mysql-standard-5.1.73-linux-i686.tar.gz.asc
gpg: Signature made Tue 01 Feb 2011 02:38:30 AM CST using DSA key ID 5072E1F5
gpg: Good signature from "MySQL Release Engineering <mysql-build@oss.oracle.com>"
```

The *Good signature* message indicates that the file signature is valid, when compared to the signature listed on our site. But you might also see warnings, like so:

```
shell> gpg --verify mysql-standard-5.1.73-linux-i686.tar.gz.asc
gpg: Signature made Wed 23 Jan 2013 02:25:45 AM PST using DSA key ID 5072E1F5
gpg: checking the trustdb
   gpg: no ultimately trusted keys found
   gpg: Good signature from "MySQL Release Engineering <mysql-build@oss.oracle.com>"
   gpg: WARNING: This key is not certified with a trusted signature!
   gpg: WARNING: This key is not certified with a trusted signature!
   Primary key fingerprint: A4A9 4068 76FC BD3C 4567  70C8 8C71 8D3B 5072 E1F5
```

That is normal, as they depend on your setup and configuration. Here are explanations for these warnings:

- *gpg: no ultimately trusted keys found:* This means that the specific key is not "ultimately trusted" by you or your web of trust, which is okay for the purposes of verifying file signatures.

- **WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.** This refers to your level of trust in your belief that you possess our real public key. This is a personal decision. Ideally, a MySQL developer would hand you the key in person, but more commonly, you downloaded it. Was the download tampered with? Probably not, but this decision is up to you. Setting up a web of trust is one method for trusting them.

See the GPG documentation for more information on how to work with public keys.

### 2.3.3 Signature Checking Using Gpg4win for Windows

The Section 2.3.2, “Signature Checking Using GnuPG” section describes how to verify MySQL downloads using GPG. That guide also applies to Microsoft Windows, but another option is to use a GUI tool like
**Gpg4win.** You may use a different tool but our examples are based on Gpg4win, and utilize its bundled **Kleopatra** GUI.

Download and install Gpg4win, and then load Kleopatra. The dialog should look similar to:

**Figure 2.1 Initial screen after loading Kleopatra**

Next, add the MySQL Release Engineering certificate. Do this by clicking **File, Lookup Certificates on Server.** Type "Mysql Release Engineering" into the search box and press **Search.**

**Figure 2.2 Finding the MySQL Release Engineering certificate**
Select the “MySQL Release Engineering” certificate. The Fingerprint and Key-ID must be "5072E1F5", or choose Details... to confirm the certificate is valid. Now, import it by clicking Import. An import dialog will be displayed, choose Okay, and this certificate will now be listed under the Imported Certificates tab.

Next, configure the trust level for our certificate. Select our certificate, then from the main menu select Certificates, Change Owner Trust... We suggest choosing I believe checks are very accurate for our certificate, as otherwise you might not be able to verify our signature. Select I believe checks are very accurate and then press OK.

Figure 2.3 Changing the Trust level

Next, verify the downloaded MySQL package file. This requires files for both the packaged file, and the signature. The signature file must have the same name as the packaged file but with an appended .asc extension, as shown by the example in the following table. The signature is linked to on the downloads page for each MySQL product. You must create the .asc file with this signature.

Table 2.2 MySQL Package and Signature Files for MySQL Installer for Microsoft Windows

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution file</td>
<td>mysql-installer-community-5.1.73.msi</td>
</tr>
<tr>
<td>Signature file</td>
<td>mysql-installer-community-5.1.73.msi.asc</td>
</tr>
</tbody>
</table>

Make sure that both files are stored in the same directory and then run the following command to verify the signature for the distribution file. Either drag and drop the signature (.asc) file into Kleopatra, or load the dialog from File, Decrypt/Verify Files..., and then choose either the .msi or .asc file.
Figure 2.4 The Decrypt/Verify Files dialog

Click Decrypt/Verify to check the file. The two most common results will look like the following, and although the yellow warning looks problematic, the following means that the file check passed with success. You may now run this installer.
Figure 2.5 The Decrypt/Verify Results: Good

Seeing a red "The signature is bad" error means the file is invalid. Do not execute the MSI file if you see this error.
Section 2.3.2, “Signature Checking Using GnuPG” section explains why you probably don’t see a green Good signature result.

2.3.4 Signature Checking Using RPM

For RPM packages, there is no separate signature. RPM packages have a built-in GPG signature and MD5 checksum. You can verify a package by running the following command:

```
shell> rpm --checksig package_name.rpm
```

Example:

```
shell> rpm --checksig MySQL-server-5.1.73-0.glibc23.i386.rpm
MySQL-server-5.1.73-0.glibc23.i386.rpm: md5 gpg OK
```

Note

If you are using RPM 4.1 and it complains about (GPG) NOT OK (MISSING KEYS: GPG#5072e1f5), even though you have imported the MySQL public build
key into your own GPG keyring, you need to import the key into the RPM keyring
first. RPM 4.1 no longer uses your personal GPG keyring (or GPG itself). Rather,
RPM maintains a separate keyring because it is a system-wide application and a
user's GPG public keyring is a user-specific file. To import the MySQL public key
into the RPM keyring, first obtain the key, then use `rpm --import` to import the
key. For example:

```
shell> gpg --export -a 5072e1f5 > 5072e1f5.asc
shell> rpm --import 5072e1f5.asc
```

Alternatively, `rpm` also supports loading the key directly from a URL, and you can use this manual page:

```
```

If you need to obtain the MySQL public key, see Section 2.3.2, “Signature Checking Using GnuPG”.

### 2.4 Installation Layouts

The installation layout differs for different installation types (for example, native packages, binary tarballs,
and source tarballs), which can lead to confusion when managing different systems or using different
installation sources. The individual layouts are given in the corresponding installation type or platform
chapter, as described following. Note that the layout of installations from vendors other than Oracle may
differ from these layouts.

- Section 5.1, “MySQL Installation Layout on Microsoft Windows”
- Section 4.1, “MySQL Layout for Source Installation”
- Table 3.1, “MySQL Installation Layout for Generic Unix/Linux Binary Package”
- Table 7.1, “MySQL Installation Layout for Linux RPM Packages”
- Table 6.2, “MySQL Installation Layout on OS X”

### 2.5 Compiler-Specific Build Characteristics

In some cases, the compiler used to build MySQL affects the features available for use. The notes in this
section apply for binary distributions provided by Oracle Corporation or that you compile yourself from
source.

**icc (Intel C++ Compiler) Builds**

A server built with `icc` has these characteristics:

- SSL support is not included.
- **InnoDB Plugin** is not included.
Chapter 3 Installing MySQL on Unix/Linux Using Generic Binaries

Oracle provides a set of binary distributions of MySQL. These include generic binary distributions in the form of compressed tar files (files with a .tar.gz extension) for a number of platforms, and binaries in platform-specific package formats for selected platforms.

This section covers the installation of MySQL from a compressed tar file binary distribution. For other platform-specific package formats, see the other platform-specific sections. For example, for Windows distributions, see Chapter 5, Installing MySQL on Microsoft Windows.

To obtain MySQL, see Section 2.2, “How to Get MySQL”.

MySQL compressed tar file binary distributions have names of the form mysql-VERSION-OS.tar.gz, where VERSION is a number (for example, 5.1.73), and OS indicates the type of operating system for which the distribution is intended (for example, pc-linux-i686 or winx64).

**Warning**

If you have previously installed MySQL using your operating system native package management system, such as yum or apt-get, you may experience problems installing using a native binary. Make sure your previous MySQL installation has been removed entirely (using your package management system), and that any additional files, such as old versions of your data files, have also been removed. You should also check for configuration files such as /etc/my.cnf or the /etc/mysql directory and delete them.

If you run into problems and need to file a bug report, please use the instructions in How to Report Bugs or Problems.

On Unix, to install a compressed tar file binary distribution, unpack it at the installation location you choose (typically /usr/local/mysql). This creates the directories shown in the following table.

**Table 3.1 MySQL Installation Layout for Generic Unix/Linux Binary Package**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin, scripts</td>
<td>mysqlld server, client and utility programs</td>
</tr>
<tr>
<td>data</td>
<td>Log files, databases</td>
</tr>
<tr>
<td>docs</td>
<td>MySQL manual in Info format</td>
</tr>
<tr>
<td>man</td>
<td>Unix manual pages</td>
</tr>
<tr>
<td>include</td>
<td>Include (header) files</td>
</tr>
<tr>
<td>lib</td>
<td>Libraries</td>
</tr>
<tr>
<td>share</td>
<td>Miscellaneous support files, including error messages, sample configuration files, SQL for database installation</td>
</tr>
<tr>
<td>sql-bench</td>
<td>Benchmarks</td>
</tr>
</tbody>
</table>

Debug versions of the mysqlld binary are available as mysqlld-debug. To compile your own debug version of MySQL from a source distribution, use the appropriate configuration options to enable debugging support. See Chapter 4, Installing MySQL from Source.

To install and use a MySQL binary distribution, the command sequence looks like this:
Create a mysql User and Group

Note
This procedure assumes that you have root (administrator) access to your system. Alternatively, you can prefix each command using the sudo (Linux) or pfexec (OpenSolaris) command.

Note
The procedure does not assign passwords to MySQL accounts. To do so, use the instructions in Section 10.4, “Securing the Initial MySQL Accounts”.

A more detailed version of the preceding description for installing a binary distribution follows.

Create a mysql User and Group

If your system does not already have a user and group to use for running mysqld, you may need to create one. The following commands add the mysql group and the mysql user. You might want to call the user and group something else instead of mysql. If so, substitute the appropriate name in the following instructions. The syntax for useradd and groupadd may differ slightly on different versions of Unix, or they may have different names such as adduser and addgroup.

```
shell> groupadd mysql
shell> useradd -r -g mysql -s /bin/false mysql
```

Note
Because the user is required only for ownership purposes, not login purposes, the useradd command uses the -r and -s /bin/false options to create a user that does not have login permissions to your server host. Omit these options if your useradd does not support them.

Obtain and Unpack the Distribution

Pick the directory under which you want to unpack the distribution and change location into it. The example here unpacks the distribution under /usr/local. The instructions, therefore, assume that you have permission to create files and directories in /usr/local. If that directory is protected, you must perform the installation as root.

```
shell> cd /usr/local
```
Perform Postinstallation Setup

Obtain a distribution file using the instructions in Section 2.2, “How to Get MySQL”. For a given release, binary distributions for all platforms are built from the same MySQL source distribution.

Unpack the distribution, which creates the installation directory. Then create a symbolic link to that directory. tar can uncompress and unpack the distribution if it has z option support:

```
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.gz
shell> ln -s full-path-to-mysql-VERSION-OS mysql
```

The tar command creates a directory named mysql-VERSION-OS. The ln command makes a symbolic link to that directory. This enables you to refer more easily to the installation directory as /usr/local/mysql.

To install MySQL from a compressed tar file binary distribution, your system must have GNU gunzip to uncompress the distribution and a reasonable tar to unpack it. If your tar program supports the z option, it can both uncompress and unpack the file.

GNU tar is known to work. The standard tar provided with some operating systems is not able to unpack the long file names in the MySQL distribution. You should download and install GNU tar, or if available, use a preinstalled version of GNU tar. Usually this is available as gnutar, gtar, or as tar within a GNU or Free Software directory, such as /usr/sfw/bin or /usr/local/bin. GNU tar is available from http://www.gnu.org/software/tar/.

If your tar does not have z option support, use gunzip to unpack the distribution and tar to unpack it. Replace the preceding tar command with the following alternative command to uncompress and extract the distribution:

```
shell> gunzip < /path/to/mysql-VERSION-OS.tar.gz | tar xvf -
```

Perform Postinstallation Setup

The remainder of the installation process involves setting distribution ownership and access permissions, initializing the data directory, starting the MySQL server, and setting up the configuration file. For instructions, see Chapter 10, Postinstallation Setup and Testing.
Chapter 4 Installing MySQL from Source

Table of Contents

4.1 MySQL Layout for Source Installation ................................................................. 22
4.2 Installing MySQL Using a Standard Source Distribution ...................................... 22
4.3 Installing MySQL Using a Development Source Tree ........................................... 25
4.4 MySQL Source-Configuration Options ................................................................. 27
4.5 Dealing with Problems Compiling MySQL ......................................................... 37
4.6 Compiling and Linking an Optimized mysqld Server .......................................... 40
4.7 Installing MySQL from Source on Windows ....................................................... 41
4.8 Notes on Installing MySQL on Solaris from Source ........................................... 45
4.9 Notes on Installing MySQL on AIX from Source ............................................... 47
4.10 Notes on Installing MySQL on HP-UX from Source ......................................... 47

Building MySQL from the source code enables you to customize build parameters, compiler optimizations, and installation location. For a list of systems on which MySQL is known to run, see http://www.mysql.com/support/supportedplatforms/database.html.

Before you proceed with an installation from source, check whether we produce a precompiled binary distribution for your platform and whether it works for you. We put a great deal of effort into ensuring that our binaries are built with the best possible options for optimal performance. Instructions for installing binary distributions are available in Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

To obtain a source distribution for MySQL, see Section 2.2, “How to Get MySQL”. MySQL source distributions are available as compressed .tar.gz files, Zip archives, or RPM packages. Distribution files have names of the form mysql-VERSION.tar.gz, mysql-VERSION.zip, or mysql-VERSION.rpm, where VERSION is a number like 5.1.73.

To perform a MySQL installation using the source code:

• To build MySQL from source on Unix-like systems, including Linux, commercial Unix, BSD, OS X and others using a .tar.gz or RPM-based source code distribution, see Section 4.2, “Installing MySQL Using a Standard Source Distribution”.

• To build MySQL from source on Windows (Windows XP or newer required), see Section 4.7, “Installing MySQL from Source on Windows”.

• For information on building from one of our development trees, see Section 4.3, “Installing MySQL Using a Development Source Tree”.

• For information on using the configure command to specify the source build parameters, including links to platform specific parameters that you might need, see Section 4.4, “MySQL Source-Configuration Options”.

To install MySQL from source, the following system requirements must be satisfied:

• GNU gunzip to uncompress the distribution and a reasonable tar to unpack it (if you use a .tar.gz distribution), or WinZip or another tool that can read .zip files (if you use a .zip distribution).

GNU tar is known to work. The standard tar provided with some operating systems is not able to unpack the long file names in the MySQL distribution. You should download and install GNU tar, or if available, use a preinstalled version of GNU tar. Usually this is available as gnutar, gtar, or as tar within a GNU or Free Software directory, such as /usr/sfw/bin or /usr/local/bin. GNU tar is available from http://www.gnu.org/software/tar/.
MySQL Layout for Source Installation

• A working ANSI C++ compiler. GCC 3.4.6 or later, Sun Studio 10 or later, Visual Studio 2005 or later, and many current vendor-supplied compilers are known to work.

• A good `make` program. Although some platforms come with their own `make` implementations, it is highly recommended that you use GNU `make` 3.75 or newer. It may already be available on your system as `gmake`. GNU `make` is available from http://www.gnu.org/software/make/.

• `libtool` 1.5, available from http://www.gnu.org/software/libtool/. 1.5.24 or later is recommended.

If you run into problems and need to file a bug report, please use the instructions in How to Report Bugs or Problems.

4.1 MySQL Layout for Source Installation

By default, when you install MySQL after compiling it from a source distribution, the installation step installs files under `/usr/local`. Components are installed in the directories shown in the following table. To configure particular installation locations, use the options described at Section 4.4, “MySQL Source-Configuration Options”.

Table 4.1 MySQL Layout for Installation from Source

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Client programs and scripts</td>
</tr>
<tr>
<td>include/mysql</td>
<td>Include (header) files</td>
</tr>
<tr>
<td>Docs</td>
<td>Manual in Info format</td>
</tr>
<tr>
<td>man</td>
<td>Unix manual pages</td>
</tr>
<tr>
<td>lib/mysql</td>
<td>Libraries</td>
</tr>
<tr>
<td>libexec</td>
<td>The <code>mysqld</code> server</td>
</tr>
<tr>
<td>share/mysql</td>
<td>Miscellaneous support files, including error messages, sample configuration files, SQL for database installation</td>
</tr>
<tr>
<td>sql-bench</td>
<td>Benchmarks</td>
</tr>
<tr>
<td>var</td>
<td>Log files, databases</td>
</tr>
</tbody>
</table>

Within its installation directory, the layout of a source installation differs from that of a binary installation in the following ways:

• The `mysqld` server is installed in the `libexec` directory rather than in the `bin` directory.

• The data directory is `var` rather than `data`.

• `mysql_install_db` is installed in the `bin` directory rather than in the `scripts` directory.

• The header file and library directories are `include/mysql` and `lib/mysql` rather than `include` and `lib`.

4.2 Installing MySQL Using a Standard Source Distribution

To install MySQL from source, first configure, build, and install from a source package. Then follow the same postinstallation setup sequence as for a binary installation.

If you start from a source RPM, use the following command to make a binary RPM that you can install. If you do not have `rpmbuild`, use `rpm` instead.

    shell> rpmbuild --rebuild --clean MySQL-VERSION.src.rpm
The result is one or more binary RPM packages that you install as indicated in Section 7.1, “Installing MySQL on Linux Using RPM Packages”.

The sequence for installation from a compressed tar file source distribution is similar to the process for installing from a generic binary distribution that is detailed in Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries. For a MySQL .tar.gz source distribution, the basic installation command sequence looks like this:

```
# Preconfiguration setup
shell> groupadd mysql
shell> useradd -g mysql -s /bin/false mysql
# Beginning of source-build specific instructions
shell> tar zxvf mysql-VERSION.tar.gz
shell> cd mysql-VERSION
shell> ./configure --prefix=/usr/local/mysql
shell> make
shell> make install
# End of source-build specific instructions
# Postinstallation setup
shell> cd /usr/local/mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> bin/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql var
# Next command is optional
shell> cp support-files/my-medium.cnf /etc/my.cnf
shell> bin/mysqld_safe --user=mysql &
# Next command is optional
shell> cp support-files/mysql.server /etc/init.d/mysql.server
```

A more detailed version of the source-build specific instructions is shown following. Perform the following steps as the mysql user, except as noted.

Note

The procedure shown here does not set up any passwords for MySQL accounts. After following the procedure, proceed to Chapter 10, Postinstallation Setup and Testing, for postinstallation setup and testing.

1. Set up the mysql user and group that will be used to run and execute the MySQL server and own the database directory. For details, see Creating a mysql System User and Group, in Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

2. Pick the directory under which you want to unpack the distribution and change location into it.

3. Obtain a distribution file using the instructions in Section 2.2, “How to Get MySQL”.

4. Unpack the distribution into the current directory. tar can uncompress and unpack the distribution if it has z option support:

```
shell> tar zxvf /path/to/mysql-VERSION.tar.gz
```

This command creates a directory named mysql-VERSION.

If your tar does not have z option support, use gunzip to unpack the distribution and tar to unpack it:

```
shell> gunzip < /path/to/mysql-VERSION.tar.gz | tar xvzf -
```
5. Change location into the top-level directory of the unpacked distribution:

```
shell> cd mysql-VERSION
```

6. Configure the source directory:

```
shell> ./configure --prefix=/usr/local/mysql
```

When you run `configure`, you might want to specify other options. For example, if you need to debug mysqld or a MySQL client, run `configure` with the `--with-debug` option, and then recompile and link your clients with the new client library. See Debugging and Porting MySQL.

Run `./configure --help` for a list of options. Section 4.4, “MySQL Source-Configuration Options”, discusses some of the more useful options.

If `configure` fails and you are going to send mail to a MySQL mailing list to ask for assistance, please include any lines from `config.log` that you think can help solve the problem. Also include the last couple of lines of output from `configure`. To file a bug report, please use the instructions in How to Report Bugs or Problems.

7. Compile the source distribution:

```
shell> make
```

Use `gmake` instead on systems where you are using GNU `make` and it has been installed as `gmake`.

If the compile fails, see Section 4.5, “Dealing with Problems Compiling MySQL”, for help.

8. Install the distribution:

```
shell> make install
```

You might need to run this command as `root`.

The remainder of the installation process, including setting up the configuration file, creating the core databases, and starting the MySQL server, are identical to the remainder of the process as shown in Generic Binary Install.

After everything has been installed, test the distribution. To start the MySQL server, use the following command:

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

If you run the command as `root`, you should use the `--user` option as shown. The option value is the name of the login account that you created in the first step to use for running the server. If you run the `mysqld_safe` command while logged in as that user, you can omit the `--user` option.

If the command fails immediately and prints `mysqld ended`, look for information in the error log (which by default is the `host_name.err` file in the data directory).

For more information about `mysqld_safe`, see `mysqld_safe — MySQL Server Startup Script`.

To make it more convenient to invoke programs installed in `/usr/local/mysql/bin`, you can add that directory to your `PATH` environment variable setting. That enables you to run a program by typing only its name, not its entire path name. See Setting Environment Variables.
4.3 Installing MySQL Using a Development Source Tree

This section discusses how to install MySQL from the latest development source code.

To obtain the source tree, you must have Bazaar installed. The Bazaar VCS Web site has instructions for downloading and installing Bazaar on different platforms. Bazaar is supported on any platform that supports Python, and is therefore compatible with any Linux, Unix, Windows, or OS X host.

MySQL development projects are hosted on Launchpad. MySQL projects, including MySQL Server, MySQL Workbench, and others are available from the Oracle/MySQL Engineering page. For the repositories related only to MySQL Server, see the MySQL Server page.

To build under Unix/Linux, you must have the following tools installed:

- A good make program. Although some platforms come with their own make implementations, it is highly recommended that you use GNU make 3.75 or newer. It may already be available on your system as gmake. GNU make is available from http://www.gnu.org/software/make/.
- libtool 1.5, available from http://www.gnu.org/software/libtool/ 1.5.24 or later is recommended.
- bison 2.1 or newer, available from http://www.gnu.org/software/bison/ (Version 1 is no longer supported.) Use the latest version of bison where possible; if you experience problems, upgrade to a later version, rather than revert to an earlier one.


Once the necessary tools are installed, create a local branch of the MySQL development tree on your machine using this procedure:

1. To obtain a copy of the MySQL source code, you must create a new Bazaar branch. If you do not already have a Bazaar repository directory set up, you must initialize a new directory:

        shell> mkdir mysql-server
        shell> bzr init-repo --trees mysql-server

This is a one-time operation.

2. Assuming that you have an initialized repository directory, you can branch from the public MySQL server repositories to create a local source tree. To create a branch of a specific version:

        shell> cd mysql-server
        shell> bzr branch lp:mysql-server/5.1 mysql-5.1

Note

The accounts that are listed in the MySQL grant tables initially have no passwords. After starting the server, you should set up passwords for them using the instructions in Chapter 10, Postinstallation Setup and Testing.
This is a one-time operation per source tree. You can branch the source trees for several versions of MySQL under the mysql-server directory.

3. The initial download will take some time to complete, depending on the speed of your connection. Please be patient. Once you have downloaded the first tree, additional trees should take significantly less time to download.

4. When building from the Bazaar branch, you may want to create a copy of your active branch so that you can make configuration and other changes without affecting the original branch contents. You can achieve this by branching from the original branch:

```
shell> bzr branch mysql-5.1 mysql-5.1-build
```

5. To obtain changes made after you have set up the branch initially, update it using the pull option periodically. Use this command in the top-level directory of the local copy:

```
shell> bzr pull
```

To examine the changeset comments for the tree, use the log option to bzr:

```
shell> bzr log
```

You can also browse changesets, comments, and source code online at the Launchpad MySQL Server page.

If you see diffs (changes) or code that you have a question about, do not hesitate to send email to the MySQL internals mailing list. See MySQL Mailing Lists. If you think you have a better idea on how to do something, send an email message to the list with a patch.

After you have the local branch, you can build MySQL server from the source code. On Windows, the build process is different from Unix/Linux: see Section 4.7, “Installing MySQL from Source on Windows”.

On Unix/Linux, use the autoconf system to create the configure script so that you can configure the build environment before building. The following example shows the typical commands required to build MySQL from a source tree.

1. Change location to the top-level directory of the source tree; replace mysql-5.1 with the appropriate directory name.

```
shell> cd mysql-5.1
```

2. Prepare the source tree for configuration.

Prior to MySQL 5.1.12, you must separately configure the InnoDB storage engine. Run the following command from the main source directory:

```
shell> (cd storage/innobase; autoreconf --force --install)
```

You can omit the previous command for MySQL 5.1.12 and later, or if you do not require InnoDB support.

Prepare the remainder of the source tree:
MySQL Source-Configuration Options

As an alternative to the preceding `autoreconf` command, you can use `BUILD/autorun.sh`, which acts as a shortcut for the following sequence of commands:

```shell
aslocal; autoheader
libtoolize --automake --force
automake --force --add-missing; autoconf
```

If you get some strange errors during this stage, verify that you have the correct version of `libtool` installed.

3. Configure the source tree and compile MySQL:

```shell
./configure  # Add your favorite options here
make
```

For a description of some `configure` options, see Section 4.4, “MySQL Source-Configuration Options”.

A collection of configuration scripts is located in the `BUILD/` subdirectory. For example, you may find it more convenient to use the `BUILD/compile-pentium-debug` script than the preceding set of shell commands. To compile on a different architecture, modify the script by removing flags that are Pentium-specific, or use another script that may be more appropriate. These scripts are provided on an “as-is” basis. They are not supported and their contents may change from release to release.

4. When the build is done, run `make install`. Be careful with this on a production machine; the installation command may overwrite your live release installation. If you already have MySQL installed and do not want to overwrite it, run `./configure` with values for the `--prefix`, `--with-tcp-port`, and `--with-unix-socket-path` options different from those used by your production server. For additional information about preventing multiple servers from interfering with each other, see Running Multiple MySQL Instances on One Machine.

5. Play hard with your new installation. For example, try to make new features crash. Start by running `make test`. See The MySQL Test Suite.

6. If you have gotten to the `make` stage, but the distribution does not compile, please enter the problem into our bugs database using the instructions given in How to Report Bugs or Problems. If you have installed the latest versions of the required tools, and they crash trying to process our configuration files, please report that also. However, if you get a `command not found` error or a similar problem for required tools, do not report it. Instead, make sure that all the required tools are installed and that your `PATH` variable is set correctly so that your shell can find them.

4.4 MySQL Source-Configuration Options

The `configure` script provides a great deal of control over how you configure a MySQL source distribution. Typically, you do this using options on the `configure` command line. For a full list of options supported by `configure`, run this command:

```shell
./configure --help
```

You can also affect `configure` using certain environment variables. See Chapter 12, Environment Variables.

The following table shows the available `configure` options.
### Formats MySQL Source-Configuration Option Reference (configure)

<table>
<thead>
<tr>
<th>Formats</th>
<th>Description</th>
<th>Default</th>
<th>Introduced</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--bindir</td>
<td>User executables</td>
<td>EPREFIX/bin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--build</td>
<td>Configure for building on BUILD</td>
<td>guessed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--cache-file</td>
<td>Cache test results in FILE</td>
<td>disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--config-cache</td>
<td>Alias for `--cache-file=config.cache'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--datadir</td>
<td>Read-only architecture-independent data</td>
<td>PREFIX/share</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-FEATURE</td>
<td>Do not include FEATURE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-community-features</td>
<td>Disable additional features provided by the community</td>
<td>5.1.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-dependency-tracking</td>
<td>Disable dependency tracking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-grant-options</td>
<td>Disable GRANT options</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-largefile</td>
<td>Omit support for large files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-libtool-lock</td>
<td>Disable libtool lock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--disable-thread-safe-client</td>
<td>Compile the client without threads</td>
<td>5.1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-FEATURE</td>
<td>Enable FEATURE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-assembler</td>
<td>Use assembler versions of some string functions if available</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-debug-sync</td>
<td>Compile in Debug Sync facility</td>
<td>5.1.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-dependency-tracking</td>
<td>Do not reject slow dependency extractors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-fast-install</td>
<td>Optimize for fast installation</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-local-infile</td>
<td>Enable LOCAL for LOAD DATA INFILE</td>
<td>disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-profiling</td>
<td>Build a version with query profiling code</td>
<td>5.1.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-shared</td>
<td>Build shared libraries</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-static</td>
<td>Build static libraries</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-thread-safe-client</td>
<td>Compile the client with threads</td>
<td>5.1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--exec-prefix</td>
<td>Install architecture-dependent files in EPREFIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--help</td>
<td>Display help message and exit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--host</td>
<td>Cross-compile to build programs to run on HOST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--includedir</td>
<td>C header files</td>
<td>PREFIX/include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formats</td>
<td>Description</td>
<td>Default</td>
<td>Introduced</td>
<td>Removed</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>--infodir</td>
<td>Info documentation</td>
<td>PREFIX/info</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--libdir</td>
<td>Object code libraries</td>
<td>EPREFIX/lib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--libexecdir</td>
<td>Program executables</td>
<td>EPREFIX/libexec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--localstatedir</td>
<td>Modifiable single-machine data</td>
<td>PREFIX/var</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--mandir</td>
<td>man documentation</td>
<td>PREFIX/man</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--no-create</td>
<td>Do not create output files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--oldincludedir</td>
<td>C header files for non-gcc</td>
<td>/usr/include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--prefix</td>
<td>Install architecture-independent files in PREFIX</td>
<td>EPREFIX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--program-prefix</td>
<td>Prepend PREFIX to installed program names</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--program-suffix</td>
<td>Append SUFFIX to installed program names</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--program-transform-name</td>
<td>run sed PROGRAM on installed program names</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--quiet</td>
<td>Do not print `checking...' messages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--sbindir</td>
<td>System administrative executables</td>
<td>EPREFIX/sbin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--sharedstatedir</td>
<td>Modifiable architecture-independent data</td>
<td>PREFIX/com</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--srcdir</td>
<td>Find the sources in DIR</td>
<td>configure directory or ..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--sysconfdir</td>
<td>Read-only single-machine data</td>
<td>PREFIX/etc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--target</td>
<td>Configure for building compilers for TARGET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--version</td>
<td>Display version information and exit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-PACKAGE</td>
<td>Use PACKAGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-archive-storage-engine</td>
<td>Enable the Archive Storage Engine</td>
<td>no</td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--with-atomic-ops</td>
<td>Implement atomic operations using pthread rwlocks or atomic CPU instructions for multi-processor</td>
<td></td>
<td>5.1.12</td>
<td></td>
</tr>
<tr>
<td>--with-berkeley-db</td>
<td>Use BerkeleyDB located in DIR</td>
<td>no</td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-berkeley-db-includes</td>
<td>Find Berkeley DB headers in DIR</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-berkeley-db-libs</td>
<td>Find Berkeley DB libraries in DIR</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>Formats</td>
<td>Description</td>
<td>Default</td>
<td>Introduced</td>
<td>Removed</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>--with-big-tables</td>
<td>Support tables with more than 4 G rows even on 32 bit platforms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-blackhole-storage-engine</td>
<td>Enable the Blackhole Storage Engine</td>
<td>no</td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--with-collation</td>
<td>Default collation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-comment</td>
<td>Comment about compilation environment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-csv-storage-engine</td>
<td>Enable the CSV Storage Engine</td>
<td>yes</td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--with-darwin-mwcc</td>
<td>Use Metrowerks CodeWarrior wrappers on OS X/Darwin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-debug</td>
<td>Add debug code (optionally with memory checker, very slow)</td>
<td></td>
<td>5.1.7</td>
<td></td>
</tr>
<tr>
<td>--with-embedded-privilege-control</td>
<td>Build parts to check user’s privileges (only affects embedded library)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-embedded-server</td>
<td>Build the embedded server</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-error-inject</td>
<td>Enable error injection in MySQL Server</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-example-storage-engine</td>
<td>Enable the Example Storage Engine</td>
<td>no</td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--with-extra-charsets</td>
<td>Use charsets in addition to default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-fast-mutexes</td>
<td>Compile with fast mutexes</td>
<td>disabled</td>
<td>5.1.5</td>
<td></td>
</tr>
<tr>
<td>--with-federated-storage-engine</td>
<td>Enable federated storage engine</td>
<td>no</td>
<td>5.1.3</td>
<td>5.1.9</td>
</tr>
<tr>
<td>--with-gnu-ld</td>
<td>Assume the C compiler uses GNU ld</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-innodb</td>
<td>Enable innobase storage engine</td>
<td>no</td>
<td>5.1.3</td>
<td>5.1.9</td>
</tr>
<tr>
<td>--with-lib-ccflags</td>
<td>Extra CC options for libraries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-libwrap</td>
<td>Compile in libwrap (tcp_wrappers) support</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-low-memory</td>
<td>Try to use less memory to compile to avoid memory limitations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-machine-type</td>
<td>Set the machine type, like &quot;powerpc&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formats</td>
<td>Description</td>
<td>Default</td>
<td>Introduced</td>
<td>Removed</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--with-maria-temp-tables</code></td>
<td>Make the temporary tables within MySQL use the Maria storage engine</td>
<td></td>
<td>5.1.24</td>
<td></td>
</tr>
<tr>
<td><code>--with-max-indexes</code></td>
<td>Sets the maximum number of indexes per table</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-mysqld-ldflags</code></td>
<td>Extra linking arguments for mysqld</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-mysqld-libs</code></td>
<td>Extra libraries to link with for mysqld</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-mysqld-user</code></td>
<td>What user the mysqld daemon shall be run as</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-mysqlmanager</code></td>
<td>Build the mysqlmanager binary</td>
<td>Build if server is built</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-named-curses-libs</code></td>
<td>Use specified curses libraries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-named-thread-libs</code></td>
<td>Use specified thread libraries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-ccflags</code></td>
<td>Extra CC options for ndb compile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-docs</code></td>
<td>Include the NDB Cluster ndbapi and mgmapi documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-port</code></td>
<td>Port for NDB Cluster management server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-port-base</code></td>
<td>Port for NDB Cluster management server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-sci</code></td>
<td>Provide MySQL with a custom location of sci library</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndb-test</code></td>
<td>Include the NDB Cluster ndbapi test programs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-ndbcluster</code></td>
<td>Include the NDB Cluster table handler</td>
<td>no</td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td><code>--with-openssl</code></td>
<td>Include the OpenSSL support</td>
<td></td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td><code>--with-openssl-includes</code></td>
<td>Find OpenSSL headers in DIR</td>
<td></td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td><code>--with-openssl-libs</code></td>
<td>Find OpenSSL libraries in DIR</td>
<td></td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td><code>--with-other-libc</code></td>
<td>Link against libc and other standard libraries installed in the specified nonstandard location</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-pic</code></td>
<td>Try to use only PIC/non-PIC objects</td>
<td>Use both</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--with-plugin-PLUGIN</code></td>
<td>Forces the named plugin to be linked into mysqld statically</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
</tbody>
</table>
MySQL Source-Configuration Options

<table>
<thead>
<tr>
<th>Formats</th>
<th>Description</th>
<th>Default</th>
<th>Introduced</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--with-plugins</td>
<td>Plugins to include in mysqld</td>
<td>none</td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-pstack</td>
<td>Use the pstack backtrace library</td>
<td></td>
<td>5.1.54</td>
<td></td>
</tr>
<tr>
<td>--with-pthread</td>
<td>Force use of pthread library</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-row-based-replication</td>
<td>Include row-based replication</td>
<td></td>
<td>5.1.5</td>
<td>5.1.6</td>
</tr>
<tr>
<td>--with-server-suffix</td>
<td>Append value to the version string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-ssl</td>
<td>Include SSL support</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--with-system-type</td>
<td>Set the system type, like &quot;sun-solaris10&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-tags</td>
<td>Include additional configurations</td>
<td>automatic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-tcp-port</td>
<td>Which port to use for MySQL services</td>
<td></td>
<td>3306</td>
<td></td>
</tr>
<tr>
<td>--with-unix-socket-path</td>
<td>Where to put the unix-domain socket</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--with-yassl</td>
<td>Include the yaSSL support</td>
<td></td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--with-zlib-dir</td>
<td>Provide MySQL with a custom location of compression library</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-PACKAGE</td>
<td>Do not use PACKAGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-bench</td>
<td>Skip building of the benchmark suite</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--without-debug</td>
<td>Build a production version without debugging code</td>
<td></td>
<td>5.1.6</td>
<td></td>
</tr>
<tr>
<td>--without-docs</td>
<td>Skip building of the documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-extra-tools</td>
<td>Skip building utilities in the tools directory</td>
<td></td>
<td>5.1.9</td>
<td></td>
</tr>
<tr>
<td>--without-geometry</td>
<td>Do not build geometry-related parts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-libedit</td>
<td>Use system libedit instead of bundled copy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-man</td>
<td>Skip building of the man pages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-ndb-binlog</td>
<td>Disable ndb binlog</td>
<td></td>
<td>5.1.6</td>
<td></td>
</tr>
<tr>
<td>--without-ndb-debug</td>
<td>Disable special ndb debug features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-plugin-PLUGIN</td>
<td>Exclude PLUGIN</td>
<td></td>
<td>5.1.11</td>
<td></td>
</tr>
<tr>
<td>--without-query-cache</td>
<td>Do not build query cache</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-readline</td>
<td>Use system readline instead of bundled copy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MySQL Source-Configuration Options

<table>
<thead>
<tr>
<th>Formats</th>
<th>Description</th>
<th>Default</th>
<th>Introduced</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--without-row-based-replication</td>
<td>Don't include row-based replication</td>
<td></td>
<td>5.1.7</td>
<td>5.1.14</td>
</tr>
<tr>
<td>--without-server</td>
<td>Only build the client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--without-uca</td>
<td>Skip building of the national Unicode collations</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you are using a version of gcc recent enough to understand the -fno-exceptions option, it is very important that you use this option. Otherwise, you may compile a binary that crashes randomly. Also use -felide-constructors and -fno-rtti along with -fno-exceptions. When in doubt, do the following:

```
CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-rtti" ./configure \
--prefix=/usr/local/mysql --enable-assembler \
--with-mysqld-ldflags=-all-static
```

On most systems, this gives you a fast and stable binary.

When compiling from source, you should also be aware of any platform specific considerations that may influence and impact the build process. Knowing and applying this information will help to ensure you get the best performance and most stable binary for your chosen platform. For more information, use the following sections:

- Section 4.9, “Notes on Installing MySQL on AIX from Source”
- Section 4.10, “Notes on Installing MySQL on HP-UX from Source”
- Section 4.8, “Notes on Installing MySQL on Solaris from Source”

Some of the configure options available are described here. For options that may be of use if you have difficulties building MySQL, see Section 4.5, “Dealing with Problems Compiling MySQL”.

Many options configure compile-time defaults that can be overridden at server startup. For example, the --prefix, --with-tcp-port, and --with-unix-socket-path options that configure the default installation base directory location, TCP/IP port number, and Unix socket file can be changed at server startup with the --basedir, --port, and --socket options for mysqld.

- To compile just the MySQL client libraries and client programs and not the server, use the --without-server option:
  ```
  shell> ./configure --without-server
  ```

If you have no C++ compiler, some client programs such as mysql cannot be compiled because they require C++. In this case, you can remove the code in configure that tests for the C++ compiler and then run ./configure with the --without-server option. The compile step should still try to build all clients, but you can ignore any warnings about files such as mysql.cc. (If make stops, try make -k to tell it to continue with the rest of the build even if errors occur.)

- To build the embedded MySQL library (libmysqld.a), use the --with-embedded-server option.
- To place your log files and database directories elsewhere than under /usr/local/var, use a configure command something like one of these:
  ```
  shell> ./configure --prefix=/usr/local/mysql
  ```
  ```
  shell> ./configure --prefix=/usr/local \n  ```
The first command changes the installation prefix so that everything is installed under `/usr/local/mysql` rather than the default of `/usr/local/`. The second command preserves the default installation prefix, but overrides the default location for database directories (normally `/usr/local/var`) and changes it to `/usr/local/mysql/data`.

You can also specify the installation directory and data directory locations at server startup time by using the `--basedir` and `--datadir` options. These can be given on the command line or in an MySQL option file, although it is more common to use an option file. See Using Option Files.

- The `--with-tcp-port` option specifies the port number on which the server listens for TCP/IP connections. The default is port 3306. To listen on a different port, use a `configure` command like this:

  ```shell
  ./configure --with-tcp-port=3307
  ```

- On Unix, if you want the MySQL socket file location to be somewhere other than the default location (normally in the directory `/tmp` or `/var/run`), use a `configure` command like this:

  ```shell
  ./configure --with-unix-socket-path=/usr/local/mysql/tmp/mysql.sock
  ```

The socket file name must be an absolute path name. You can also change the location of `mysql.sock` at server startup by using a MySQL option file. See How to Protect or Change the MySQL Unix Socket File.

- To compile statically linked programs (for example, to make a binary distribution, to get better performance, or to work around problems with some Red Hat Linux distributions), run `configure` like this:

  ```shell
  ./configure --with-client-ldflags=-all-static \ 
  --with-mysqld-ldflags=-all-static
  ```

- If you are using `gcc` and do not have `libg++` or `libstdc++` installed, you can tell `configure` to use `gcc` as your C++ compiler:

  ```shell
  CC=gcc CXX=gcc ./configure
  ```

When you use `gcc` as your C++ compiler, it does not attempt to link in `libg++` or `libstdc++`. This may be a good thing to do even if you have those libraries installed. Some versions of them have caused strange problems for MySQL users in the past.

In most cases, you can get a reasonably optimized MySQL binary by using the following options on the `configure` line:

```bash
--prefix=/usr/local/mysql --enable-assembler \ 
--with-mysqld-ldflags=-all-static
```

The full `configure` line would, in other words, be something like the following for all recent `gcc` versions:

```bash
CFLAGS="-O3 -mpentiumpro" CXX=gcc CXXFLAGS="-O3 -mpentiumpro \ 
-feilde-constructors -fno-exceptions -fno-rtti" ./configure \ 
--prefix=/usr/local/mysql --enable-assembler \ 
--with-mysqld-ldflags=-all-static
```
The binaries we provide on the MySQL Web site at http://dev.mysql.com/downloads/ are all compiled with full optimization and should work well for most users. See Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

- If the build fails and produces errors about your compiler or linker not being able to create the shared library libmysqlclient.so.N (where N is a version number), you can work around this problem by giving the --disable-shared option to configure. In this case, configure does not build a shared libmysqlclient.so.N library.

- By default, MySQL uses the latin1 (cp1252 West European) character set. To change the default set, use the --with-charset option:

```shell
./configure --with-charset=CHARSET
```

CHARSET may be one of binary, armSCII8, ascii, big5, cp1250, cp1251, cp1256, cp1257, cp850, cp852, cp866, cp932, dec8, eucjpms, euckr, gb2312, gbk, geostd8, greek, hebrew, hp8, keybcs2, koi8r, koi8u, latin1, latin2, latin5, latin7, macce, macroman, sjis, swe7, tis620, ucs2, ujis, utf8. (Additional character sets might be available. Check the output from ./configure --help for the current list.)

The default collation may also be specified. MySQL uses the latin1_swedish_ci collation by default. To change this, use the --with-collation option:

```shell
./configure --with-collation COLLATION
```

To change both the character set and the collation, use both the --with-charset and --with-collation options. The collation must be a legal collation for the character set. (Use the SHOW COLLATION statement to determine which collations are available for each character set.)

With the configure option --with-extra-charsets=LIST, you can define which additional character sets should be compiled into the server. LIST is one of the following:

- A list of character set names separated by spaces
- complex to include all character sets that can't be dynamically loaded
- all to include all character sets into the binaries

Clients that want to convert characters between the server and the client should use the SET NAMES statement. See Connection Character Sets and Collations.

- To configure MySQL with debugging code, use the --with-debug option:

```shell
./configure --with-debug
```

This causes a safe memory allocator to be included that can find some errors and that provides output about what is happening. See Debugging and Porting MySQL.

As of MySQL 5.1.12, using --with-debug to configure MySQL with debugging support enables you to use the --debug="d,parser_debug" option when you start the server. This causes the Bison parser that is used to process SQL statements to dump a parser trace to the server's standard error output. Typically, this output is written to the error log.
To cause the Debug Sync facility to be compiled into the server, use the `--enable-debug-sync` option. This facility is used for testing and debugging. When compiled in, Debug Sync is disabled by default at runtime. To enable it, start `mysqld` with the `--debug-sync-timeout=N` option, where `N` is a timeout value greater than 0. (The default value is 0, which disables Debug Sync.) `N` becomes the default timeout for individual synchronization points.

Debug Sync is also compiled in if you configure with the `--with-debug` option (which implies `--enable-debug-sync`), unless you also use the `--disable-debug-sync` option.

For a description of the Debug Sync facility and how to use synchronization points, see MySQL Internals: Test Synchronization.

The `--enable-debug-sync` and `--disable-debug-sync` options were added in MySQL 5.1.41.

If your client programs are using threads, you must compile a thread-safe version of the MySQL client library with the `--enable-thread-safe-client` configure option. This creates a `libmysqlclient_r` library with which you should link your threaded applications. See Writing C API Threaded Client Programs.

Some features require that the server be built with compression library support, such as the `COMPRESS()` and `UNCOMPRESS()` functions, and compression of the client/server protocol. The `--with-zlib-dir=no|bundled|DIR` option provides control over compression library support. The value `no` explicitly disables compression support. `bundled` causes the `zlib` library bundled in the MySQL sources to be used. A `DIR` path name specifies the directory in which to find the compression library sources.

It is possible to build MySQL with large table support using the `--with-big-tables` option.

This option causes the variables that store table row counts to be declared as `unsigned long long` rather than `unsigned long`. This enables tables to hold up to approximately 1.844E+19 (~2\(^{32}\times\)) rows rather than 2\(^{32}\) (~4.295E+09) rows. Previously it was necessary to pass `--DBIG_TABLES` to the compiler manually in order to enable this feature.

Run `configure` with the `--disable-grant-options` option to cause the `--bootstrap`, `--skip-grant-tables`, and `--init-file` options for `mysqld` to be disabled. For Windows, the `configure.js` script recognizes the `DISABLE_GRANT_OPTIONS` flag, which has the same effect. The capability is available as of MySQL 5.1.15.

This option allows MySQL Community Server features to be enabled. Additional options may be required for individual features, such as `--enable-profiling` to enable statement profiling. This option was added in MySQL 5.1.24. It is enabled by default as of MySQL 5.1.28; to disable it, use `--disable-community-features`.

When given with `--enable-community-features`, the `--enable-profiling` option enables the statement profiling capability exposed by the SHOW PROFILE and SHOW PROFILES statements. (See SHOW PROFILES Syntax.) This option was added in MySQL 5.1.24. It is enabled by default as of MySQL 5.1.28; to disable it, use `--disable-profiling`.

See Chapter 2, General Installation Guidance, for options that pertain to particular operating systems.

See Building MySQL with SSL Support, for options that pertain to configuring MySQL to support secure (encrypted) connections.

Several `configure` options apply to plugin selection and building:

```
--with-plugins=PLUGIN[,PLUGIN]...
--with-plugins=GROUP
```
Dealing with Problems Compiling MySQL

```
--with-plugin-PLUGIN
--without-plugin-PLUGIN
```

`PLUGIN` is an individual plugin name such as `csv` or `archive`.

As shorthand, `GROUP` is a configuration group name such as `none` (select no plugins) or `all` (select all plugins).

You can build a plugin as static (compiled into the server) or dynamic (built as a dynamic library that must be installed using the `INSTALL PLUGIN` statement or the `--plugin-load` option before it can be used). Some plugins might not support static or dynamic build.

`configure --help` shows the following information pertaining to plugins:

- The plugin-related options
- The names of all available plugins
- For each plugin, a description of its purpose, which build types it supports (static or dynamic), and which plugin groups it is a part of.

`--with-plugins` can take a list of one or more plugin names separated by commas, or a plugin group name. The named plugins are configured to be built as static plugins.

`--with-plugin-PLUGIN` configures the given plugin to be built as a static plugin.

`--without-plugin-PLUGIN` disables the given plugin from being built.

If a plugin is named both with a `--with` and `--without` option, the result is undefined.

For any plugin that is not explicitly selected or disabled, it is selected to be built dynamically if it supports dynamic build, and not built if it does not support dynamic build. (Thus, in the case that no plugin options are given, all plugins that support dynamic build are selected to be built as dynamic plugins. Plugins that do not support dynamic build are not built.)

4.5 Dealing with Problems Compiling MySQL

All MySQL programs compile cleanly for us with no warnings on Solaris or Linux using `gcc`. On other systems, warnings may occur due to differences in system include files. For other problems, check the following list.

The solution to many problems involves reconfiguring. If you do need to reconfigure, take note of the following:

- If `configure` is run after it has previously been run, it may use information that was gathered during its previous invocation. This information is stored in `config.cache`. When `configure` starts up, it looks for that file and reads its contents if it exists, on the assumption that the information is still correct. That assumption is invalid when you reconfigure.

- Each time you run `configure`, you must run `make` again to recompile. However, you may want to remove old object files from previous builds first because they were compiled using different configuration options.

To prevent old configuration information or object files from being used, run these commands before re-running `configure`:

```
shell> rm config.cache
```
Dealing with Problems Compiling MySQL

shell> **make clean**

Alternatively, you can run **make distclean**.

The following list describes some of the problems that have been found to occur most often when compiling MySQL:

- If you get errors such as the ones shown here when compiling `sql_yacc.cc`, you probably have run out of memory or swap space:

```
Internal compiler error: program cc1plus got fatal signal 11
Out of virtual memory
Virtual memory exhausted
```

The problem is that **gcc** requires a huge amount of memory to compile `sql_yacc.cc` with inline functions. Try running **configure** with the **--with-low-memory** option:

```
shell> ./configure --with-low-memory
```

This option causes `--fno-inline` to be added to the compile line if you are using **gcc** and `-O0` if you are using something else. You should try the **--with-low-memory** option even if you have so much memory and swap space that you think you can't possibly have run out. This problem has been observed to occur even on systems with generous hardware configurations, and the **--with-low-memory** option usually fixes it.

- By default, **configure** picks `c++` as the compiler name and GNU **c++** links with `-lg++`. If you are using **gcc**, that behavior can cause problems during configuration such as this:

```
configure: error: installation or configuration problem:
C++ compiler cannot create executables.
```

You might also observe problems during compilation related to `g++, libg++`, or `libstdc++`.

One cause of these problems is that you may not have `g++`, or you may have `g++` but not `libg++`, or `libstdc++`. Take a look at the *config.log* file. It should contain the exact reason why your C++ compiler did not work. To work around these problems, you can use **gcc** as your C++ compiler. Try setting the environment variable `CXX` to "**gcc -O3**". For example:

```
shell> CXX="gcc -O3" ./configure
```

This works because **gcc** compiles C++ source files as well as `g++` does, but does not link in `libg++` or `libstdc++` by default.

Another way to fix these problems is to install `g++, libg++`, and `libstdc++`. However, do not use `libg++` or `libstdc++` with MySQL because this only increases the binary size of `mysqld` without providing any benefits. Some versions of these libraries have also caused strange problems for MySQL users in the past.

- To define flags to be used by your C or C++ compilers, specify them using the **CFLAGS** and **CXXFLAGS** environment variables. You can also specify the compiler names this way using **CC** and **CXX**. For example:

```
shell> CC=gcc
shell> CFLAGS=-O3
shell> CXX=gcc
```
Dealing with Problems Compiling MySQL

To see what flags you might need to specify, invoke `mysql_config` with the `--cflags` option.

- If you get errors such as those shown here when compiling `mysqld`, `configure` did not correctly detect the type of the last argument to `accept()`, `getsockname()`, or `getpeername()`:

```plaintext
cxx: Error: mysqld.cc, line 645: In this statement, the referenced
type of the pointer value ''length'' is ''unsigned long'',
which is not compatible with ''int''.
new_sock = accept(sock, {struct sockaddr *)&cAddr, &length);
```

To fix this, edit the `config.h` file (which is generated by `configure`). Look for these lines:

```c
/* Define as the base type of the last arg to accept */
#define SOCKET_SIZE_TYPE XXX
```

Change `XXX` to `size_t` or `int`, depending on your operating system. (You must do this each time you run `configure` because `configure` regenerates `config.h`.)

- If your compile fails with errors such as any of the following, you must upgrade your version of `make` to GNU `make`:

```plaintext
make: Fatal error in reader: Makefile, line 18:
Badly formed macro assignment

Or:

make: file `Makefile' line 18: Must be a separator (:

Or:

pthread.h: No such file or directory
```

Solaris and FreeBSD are known to have troublesome `make` programs.

GNU `make` 3.75 is known to work.

- The `sql_yacc.cc` file is generated from `sql_yacc.yy`. Normally, the build process does not need to create `sql_yacc.cc` because MySQL comes with a pregenerated copy. However, if you do need to re-create it, you might encounter this error:

```plaintext
"sql_yacc.yy", line xxx fatal: default action causes potential...
```

This is a sign that your version of `yacc` is deficient. You probably need to install `bison` (the GNU version of `yacc`) and use that instead.

Versions of `bison` older than 1.75 may report this error:

```plaintext
sql_yacc.yy:#####: fatal error: maximum table size (32767) exceeded
```

The maximum table size is not actually exceeded; the error is caused by bugs in older versions of `bison`. 
• On Debian Linux 3.0, you need to install `gawk` instead of the default `mawk`.

• If you get a compilation error on Linux (for example, SuSE Linux 8.1 or Red Hat Linux 7.3) similar to the following one, you probably do not have `g++` installed:

```plaintext
libmysql.c:1329: warning: passing arg 5 of `gethostbyname_r' from incompatible pointer type
libmysql.c:1329: too few arguments to function `gethostbyname_r'
libmysql.c:1329: warning: assignment makes pointer from integer without a cast
make[2]: *** [libmysql.lo] Error 1
```

By default, the `configure` script attempts to determine the correct number of arguments by using `g++` (the GNU C++ compiler). This test yields incorrect results if `g++` is not installed. There are two ways to work around this problem:

• Make sure that the GNU C++ `g++` is installed. On some Linux distributions, the required package is called `gpp`; on others, it is named `gcc-c++`.

• Use `gcc` as your C++ compiler by setting the `CXX` environment variable to `gcc`:

```plaintext
export CXX="gcc"
```

You must run `configure` again after making either of those changes.

For information about acquiring or updating tools, see the system requirements in Chapter 4, `Installing MySQL from Source`.

### 4.6 Compiling and Linking an Optimized mysqld Server

Most of the following tests were performed on Linux with the MySQL benchmarks, but they should give some indication for other operating systems and workloads.

You obtain the fastest executables when you link with `-static`.

By using better compiler and compilation options, you can obtain a 10% to 30% speed increase in applications. This is particularly important if you compile the MySQL server yourself.

When we tested both the Cygnus CodeFusion and Fujitsu compilers, neither was sufficiently bug-free to enable MySQL to be compiled with optimizations enabled.

The standard MySQL binary distributions are compiled with support for all character sets. When you compile MySQL yourself, include support only for the character sets that you are going to use. This is controlled by the `--with-charset` option to `configure`.

Here is a list of some measurements that we have made:

• If you link dynamically (without `-static`), the result is 13% slower on Linux. Note that you still can use a dynamically linked MySQL library for your client applications. It is the server that is most critical for performance.

• For a connection from a client to a server running on the same host, if you connect using TCP/IP rather than a Unix socket file, performance is 7.5% slower. (On Unix, if you connect to the host name `localhost`, MySQL uses a socket file by default.)

• For TCP/IP connections from a client to a server, connecting to a remote server on another host is 8% to 11% slower than connecting to a server on the same host, even for connections faster than 100Mb/s Ethernet.
• When running our benchmark tests using secure connections (all data encrypted with internal SSL support) performance was 55% slower than with unencrypted connections.

• On a Sun UltraSPARC-Ile, a server compiled with Forte 5.0 is 4% faster than one compiled with gcc 3.2.

• On a Sun UltraSPARC-Ile, a server compiled with Forte 5.0 is 4% faster in 32-bit mode than in 64-bit mode.

• Compiling on Linux-x86 using gcc without frame pointers (-fomit-frame-pointer or -fomit-frame-pointer -ffixed-ebp) makes mysql 1% to 4% faster.

4.7 Installing MySQL from Source on Windows

These instructions describe how to build binaries from source for MySQL 5.1 on Windows. Instructions are provided for building binaries from a standard source distribution or from the Bazaar tree that contains the latest development source.

Note

The instructions here are strictly for users who want to test MySQL on Microsoft Windows from the latest source distribution or from the Bazaar tree. For production use, we do not advise using a MySQL server built by yourself from source. Normally, it is best to use precompiled binary distributions of MySQL that are built specifically for optimal performance on Windows by Oracle Corporation. Instructions for installing binary distributions are available in Chapter 5, Installing MySQL on Microsoft Windows.

To build MySQL on Windows from source, you must satisfy the following system, compiler, and resource requirements:

• Windows 2000, Windows XP, or newer version.

Windows Vista is supported when using Visual Studio 2005 provided you have installed the following updates:

• Microsoft Visual Studio 2005 Professional Edition - ENU Service Pack 1 (KB926601)

• Security Update for Microsoft Visual Studio 2005 Professional Edition - ENU (KB937061)

• Update for Microsoft Visual Studio 2005 Professional Edition - ENU (KB932232)

• CMake, which can be downloaded from http://www.cmake.org. After installing, modify your PATH environment variable to include the directory where cmake is located.


• If you are compiling from a Bazaar tree or making changes to the parser, you need bison for Windows, which can be downloaded from http://gnuwin32.sourceforge.net/packages/bison.htm. Download the package labeled “Complete package, excluding sources”. After installing the package, modify your PATH environment variable to include the directory where bison is located.
Installing MySQL from Source on Windows

**Note**

On Windows, the default location for `bison` is the `C:\Program Files \GnuWin32` directory. Some utilities, including `m4`, may fail to find `bison` because of the space in the directory name. You can resolve this by installing into a directory that does not contain a space; for example `C:\GnuWin32`.

- Cygwin might be necessary if you want to run the test script or package the compiled binaries and support files into a Zip archive. (Cygwin is needed only to test or package the distribution, not to build it.) Cygwin is available from [http://cygwin.com](http://cygwin.com).
- 3GB to 5GB of disk space.

You also need a MySQL source distribution for Windows, which can be obtained two ways:

- Obtain a source distribution packaged by Oracle Corporation. These are available from [http://dev.mysql.com/downloads/](http://dev.mysql.com/downloads/).
- Package a source distribution yourself from the latest Bazaar developer source tree. For instructions on pulling the latest source files, see Section 4.3, "Installing MySQL Using a Development Source Tree".

If you find something not working as expected, or you have suggestions about ways to improve the current build process on Windows, please send a message to the `win32` mailing list. See MySQL Mailing Lists.

**Note**

To compile from the source code on Windows you must use the standard source distribution (for example, `mysql-5.1.73.zip` or `mysql-5.1.73.tar.gz`). You build from the same distribution as used to build MySQL on Unix, Linux and other platforms. Do not use the Windows Source distributions as they do not contain the necessary configuration script and other files.

Follow this procedure to build MySQL:

1. If you are installing from a packaged source distribution, create a work directory (for example, `C:\workdir`), and unpack the source distribution there using `WinZip` or another Windows tool that can read `.zip` files. This directory is the work directory in the following instructions.

2. Start a command shell. If you have not configured the `PATH` and other environment variables for all command shells, you may be able to start a command shell from the `Start Menu` within the Windows Visual Studio menu that contains the necessary environment changes.

3. Within the command shell, navigate to the work directory and run the following command:

   ```
   C:\workdir>win\configure.js options
   ```

   If you have associated the `.js` file extension with an application such as a text editor, then you may need to use the following command to force `configure.js` to be executed as a script:

   ```
   C:\workdir>cscript win\configure.js options
   ```
Installing MySQL from Source on Windows

These options are available for `configure.js`:

- **WITH_INNODB_STORAGE_ENGINE**: Enable the InnoDB storage engine.
- **WITH_PARTITION_STORAGE_ENGINE**: Enable user-defined partitioning.
- **WITH_ARCHIVE_STORAGE_ENGINE**: Enable the ARCHIVE storage engine.
- **WITH_BLACKHOLE_STORAGE_ENGINE**: Enable the BLACKHOLE storage engine.
- **WITH_EXAMPLE_STORAGE_ENGINE**: Enable the EXAMPLE storage engine.
- **WITH_FEDERATED_STORAGE_ENGINE**: Enable the FEDERATED storage engine.
- **WITH_NDBCLUSTER_STORAGE_ENGINE**: Enable the NDBCLUSTER storage engine in the MySQL server; cause binaries for the MySQL Cluster management and data node, management client, and other programs to be built.

This option is supported only in MySQL Cluster NDB 7.0 and later (NDBCLUSTER storage engine versions 6.4.0 and later) using the MySQL Cluster sources. It cannot be used to enable clustering support in other MySQL source trees or distributions.

- **MYSQL_SERVER_SUFFIX=suffix**: Server suffix, default none.
- **COMPILATION_COMMENT=comment**: Server comment, default "Source distribution".
- **MYSQL_TCP_PORT=port**: Server port, default 3306.
- **DISABLE_GRANT_OPTIONS**: Disables the --bootstrap, --skip-grant-tables, and --init-file options for mysqld. This option is available as of MySQL 5.1.15.

For example (type the command on one line):

```
C:\workdir>win\configure.js WITH_INNODB_STORAGE_ENGINE
WITH_PARTITION_STORAGE_ENGINE MYSQL_SERVER_SUFFIX=-pro
```

4. From the work directory, execute the `win\build-vs9.bat` (Windows Visual Studio 2008), `win\build-vs8.bat` (Windows Visual Studio 2005), or `win\build-vs71.bat` (Windows Visual Studio 2003) script, depending on the version of Visual Studio you have installed. The script invokes CMake, which generates the `mysql.sln` solution file.

You can also use the corresponding 64-bit file (for example `win\build-vs8_x64.bat` or `win\build-vs9_x64.bat`) to build the 64-bit version of MySQL. However, you cannot build the 64-bit version with Visual Studio Express Edition. You must use Visual Studio 2005 (8.0) or higher.

5. From the work directory, open the generated `mysql.sln` file with Visual Studio and select the proper configuration using the **Configuration** menu. The menu provides Debug, Release, RelwithDebInfo, MinRelInfo options. Then select **Solution > Build** to build the solution.

Remember the configuration that you use in this step. It is important later when you run the test script because that script needs to know which configuration you used.

6. Test the server. The server built using the preceding instructions expects that the MySQL base directory and data directory are **C:\mysql** and **C:\mysql\data** by default. If you want to test your server using the source tree root directory and its data directory as the base directory and data directory, you need to tell the server their path names. You can either do this on the command line with
the \texttt{--basedir} and \texttt{--datadir} options, or by placing appropriate options in an option file. (See \textit{Using Option Files}.) If you have an existing data directory elsewhere that you want to use, you can specify its path name instead.

When the server is running in standalone fashion or as a service based on your configuration, try to connect to it from the \texttt{mysql} interactive command-line utility.

You can also run the standard test script, \texttt{mysql-test-run.pl}. This script is written in Perl, so you'll need either Cygwin or ActiveState Perl to run it. You may also need to install the modules required by the script. To run the test script, change location into the \texttt{mysql-test} directory under the work directory, set the \texttt{MTR\_VS\_CONFIG} environment variable to the configuration you selected earlier (or use the \texttt{--vs-config} option), and invoke \texttt{mysql-test-run.pl}. For example (using Cygwin and the bash shell):

\begin{verbatim}
shell> cd mysql-test
shell> export MTR_VS_CONFIG=debug
shell> ./mysql-test-run.pl --force --timer
shell> ./mysql-test-run.pl --force --timer --ps-protocol
\end{verbatim}

When you are satisfied that the programs you have built are working correctly, stop the server. Now you can install the distribution. One way to do this is to use the \texttt{make_win_bin_dist} script in the \texttt{scripts} directory of the MySQL source distribution (see \texttt{make_win_bin_dist — Package MySQL Distribution as Zip Archive}). This is a shell script, so you must have Cygwin installed if you want to use it. It creates a Zip archive of the built executables and support files that you can unpack in the location at which you want to install MySQL.

It is also possible to install MySQL by copying directories and files directly:

1. Create the directories where you want to install MySQL. For example, to install into \texttt{C:\mysql}, use these commands:

\begin{verbatim}
C:\> mkdir C:\mysql
C:\> mkdir C:\mysql\bin
C:\> mkdir C:\mysql\data
C:\> mkdir C:\mysql\share
C:\> mkdir C:\mysql\scripts
\end{verbatim}

If you want to compile other clients and link them to MySQL, you should also create several additional directories:

\begin{verbatim}
C:\> mkdir C:\mysql\include
C:\> mkdir C:\mysql\lib
C:\> mkdir C:\mysql\lib\debug
C:\> mkdir C:\mysql\lib\opt
\end{verbatim}

If you want to benchmark MySQL, create this directory:

\begin{verbatim}
C:\> mkdir C:\mysql\sql-bench
\end{verbatim}

Benchmarking requires Perl support for MySQL. See Chapter 13, \textit{Perl Installation Notes}.

2. From the work directory, copy into the \texttt{C:\mysql} directory the following files and directories:

\begin{verbatim}
C:\> cd \workdir
C:\workdir> mkdir C:\mysql
C:\workdir> mkdir C:\mysql\bin
\end{verbatim}
If you want to compile other clients and link them to MySQL, you should also copy several libraries and header files:

```plaintext
C:\workdir> copy lib\Release\mysqlclient.lib C:\mysql\lib\debug
C:\workdir> copy lib\Release\libmysql.* C:\mysql\lib\debug
C:\workdir> copy lib\Release\zlib.* C:\mysql\lib\debug
C:\workdir> copy lib\Release\mysqlclient.lib C:\mysql\lib\opt
C:\workdir> copy lib\Release\libmysql.* C:\mysql\lib\opt
C:\workdir> copy lib\Release\zlib.* C:\mysql\lib\opt
C:\workdir> copy include\*.h C:\mysql\include
C:\workdir> copy libmysql\libmysql.def C:\mysql\include
```

**Note**

If you have compiled a Debug solution, rather than a Release solution, install it by replacing `Release` with `Debug` in the source file names just shown.

If you want to benchmark MySQL, you should also do this:

```plaintext
C:\workdir> xcopy sql-bench\*.* C:\mysql\bench /E
```

After installation, set up and start the server in the same way as for binary Windows distributions. This includes creating the system tables by running `mysql_install_db`. For more information, see Chapter 5, *Installing MySQL on Microsoft Windows.*

### 4.8 Notes on Installing MySQL on Solaris from Source

When building MySQL on Solaris you can use either the Sun Studio or GNU cc compilers. For more information on specific notes and environments, use the following hints.

- When building you should ensure that your `PATH` variable includes the necessary tools, including `ar` for building libraries. Some tools are located in `/usr/ccs/bin`.

- When running `configure`, you should specify the C and C++ compiler explicitly to ensure that the right C compiler combination is used:

  ```plaintext
  CC=gcc CXX=g++ ./configure
  ```

- If you have an UltraSPARC system, you can get 4% better performance by adding `-mcpu=v8 -Wa,-xarch=v8plusa` to the `CFLAGS` and `CXXFLAGS` environment variables.

- If you have Sun's Forte 5.00 (or newer) or Sun Studio compiler, you can run `configure` like this:

  ```plaintext
  CC=cc CXX=cc CFLAGS="-Xa -fast -native -xstrconst -mt" \
  CXXFLAGS="-noex -mt" \
  ./configure --prefix=/usr/local/mysql --enable-assembler
  ```

- To create a 64-bit SPARC binary with Sun's Forte or Sun Studio compiler, use the following configuration options:

  ```plaintext
  CC=cc CFLAGS="-Xa -fast -native -xstrconst -mt -xarch=v9" \
  CXX=cc CXXFLAGS="-noex -mt -xarch=v9" ASFLAGS="-xarch=v9"
  ```
Notes on Installing MySQL on Solaris from Source

./configure --prefix=/usr/local/mysql --enable-assembler

To create a 64-bit Solaris binary using gcc, add -m64 to CFLAGS and CXXFLAGS and remove --enable-assembler from the configure line.

In the MySQL benchmarks, we obtained a 4% speed increase on UltraSPARC when using Forte 5.0 in 32-bit mode, as compared to using gcc 3.2 with the -mcpu flag.

If you create a 64-bit mysqld binary, it is 4% slower than the 32-bit binary, but can handle more threads and memory.

• If you get a problem with fdatasync or sched_yield, you can fix this by adding LIBS=-lrt to the configure line.

• Solaris does not provide static versions of all system libraries (libpthreads and libdl), so you cannot compile MySQL with --static. If you try to do so, you get one of the following errors:

```
ld: fatal: library -ldl: not found
undefined reference to `dlopen'
cannot find -lrt
```

• If you link your own MySQL client programs, you may see the following error at runtime:

```
ld.so.1: fatal: libmysqlclient.so.#{:
open failed: No such file or directory
```

To avoid this problem, use one of the following methods:

• Use the crle tool to add the directory containing the libmysqlclient library file to the list of standard library directories. You need administrator privileges to do this. Make sure you update the library information, rather than replace it with the new path. For example, the following command adds the directory to the list of standard directories searched for libraries.

```
crle -u -l /usr/local/mysql/lib
```

For 64-bit libraries, add the -64 option:

```
crle -64 -u -l /usr/local/mysql/lib
```

• Link clients with the -Wl,/full/path/to/libmysqlclient.so flag rather than with -Lpath).

• Copy libmysqlclient.so to /usr/lib.

• Add the path name of the directory where libmysqlclient.so is located to the LD_RUN_PATH environment variable before running your client.

• If you have problems with configure trying to link with -lz when you do not have zlib installed, you have two options:

  • If you want to be able to use the compressed communication protocol, obtain and install zlib from ftp.gnu.org.

  • To build without zlib, run configure with the --with-named-z-libs=no option when building MySQL.
Notes on Installing MySQL on AIX from Source

• If you are using gcc and have problems with loading user-defined functions (UDFs) into MySQL, try adding `-lgcc` to the link line for the UDF.

4.9 Notes on Installing MySQL on AIX from Source

General notes on building MySQL from source on IBM AIX:

• Automatic `xlC` detection is missing from Autoconf, so a number of variables need to be set before running `configure`. The following example uses the IBM compiler:

```bash
export CC="xlc_r -ma -O3 -qstrict -qoptimize=3 -qmaxmem=8192 
export CXX="xlC_r -ma -O3 -qstrict -qoptimize=3 -qmaxmem=8192 
export CFLAGS="-I /usr/local/include" 
export LDFLAGS="-L /usr/local/lib" 
export CPPFLAGS=$CFLAGS 
export CXXFLAGS=$CFLAGS 
./configure --prefix=/usr/local 
   --localstatedir=/var/mysql 
   --sbindir='/usr/local/bin' 
   --libexecdir='/usr/local/bin' 
   --enable-thread-safe-client 
   --enable-large-files
```

The preceding options are used to compile the MySQL distribution that can be found at [http://www-frec.bull.com/](http://www-frec.bull.com/).

• If you change the `-O3` to `-O2` in the preceding `configure` line, you must also remove the `-qstrict` option. This is a limitation in the IBM C compiler.

• If you compile MySQL with gcc, you must use the `-fno-exceptions` flag because the exception handling in gcc is not thread-safe. There are also some known problems with IBM's assembler that may cause it to generate bad code when used with gcc.

• If you have problems with signals (MySQL dies unexpectedly under high load), you may have found an OS bug with threads and signals. In this case, you can tell MySQL not to use signals by configuring as follows:

```bash
CFLAGS="-DDONT_USE_THR_ALARM" CXX=gcc 
CXXFLAGS="-felide-constructors -fno-exceptions -fno-rtti 
-DDONT_USE_THR_ALARM" 
./configure --prefix=/usr/local/mysql --with-debug 
   --with-low-memory
```

This does not affect the performance of MySQL, but has the side effect that you cannot kill clients that are "sleeping" on a connection with `mysqladmin kill` or `mysqladmin shutdown`. Instead, the client dies when it issues its next command.

• On some versions of AIX, linking with `libbind.a` makes `getservbyname()` dump core. This is an AIX bug and should be reported to IBM.

4.10 Notes on Installing MySQL on HP-UX from Source

General notes on compiling MySQL on HP-UX.

• If you are using HP-UX compiler, you can use the following command (which has been tested with cc B.11.11.04):
CC=cc CXX=aCC CFLAGS=+DD64 CXXFLAGS=+DD64 ./configure \
   --with-extra-character-set=complex

You can ignore any errors of the following type:

aCC: warning 901: unknown option: `-3': use +help for online documentation

• If you get the following error from configure, verify that you do not have the path to the K&R compiler before the path to the HP-UX C and C++ compiler:

checking for cc option to accept ANSI C... no
configure: error: MySQL requires an ANSI C compiler (and a C++ compiler).

• Another reason for compile failure is that you did not define the +DD64 flags as just described.
Consider installing the latest stable branch, which today is MySQL server 5.7.

MySQL is available for Microsoft Windows, for both 32-bit and 64-bit versions. For supported Windows platform information, see http://www.mysql.com/support/supportedplatforms/database.html.

It is possible to run MySQL as a standard application or as a Windows service. By using a service, you can monitor and control the operation of the server through the standard Windows service management tools. For more information, see Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”.

Generally, you should install MySQL on Windows using an account that has administrator rights. Otherwise, you may encounter problems with certain operations such as editing the PATH environment.
variable or accessing the Service Control Manager. Once installed, MySQL does not need to be executed using a user with Administrator privileges.

For a list of limitations on the use of MySQL on the Windows platform, see Windows Platform Limitations.

In addition to the MySQL Server package, you may need or want additional components to use MySQL with your application or development environment. These include, but are not limited to:

- To connect to the MySQL server using ODBC, you must have a Connector/ODBC driver. For more information, including installation and configuration instructions, see MySQL Connector/ODBC Developer Guide.

- To use MySQL server with .NET applications, you must have the Connector/Net driver. For more information, including installation and configuration instructions, see MySQL Connector/Net Developer Guide.


MySQL for Windows is available in several distribution formats, detailed here. Generally speaking, you should use a binary distribution that includes an installer. It is simpler to use than the others, and you need no additional tools to get MySQL up and running. The installer for the Windows version of MySQL, combined with a GUI Config Wizard, automatically installs MySQL, creates an option file, starts the server, and secures the default user accounts.

- Binary installer distribution. The installable distribution comes packaged as a Microsoft Windows Installer (MSI) package that you can install manually or automatically on your systems. Two formats are available, an essentials package that contains all the files you need to install and configure MySQL, but no additional components, and a complete package that includes MySQL, configuration tools, benchmarks and other components. For more information on the specific differences, see Section 5.2, “Choosing the Installation Package for Microsoft Windows”.

  For instructions on installing MySQL using one of the MSI installation packages, see Section 5.4, “Installing MySQL on Microsoft Windows Using an MSI Package”.

- The standard binary distribution (packaged as a Zip file) contains all of the necessary files that you unpack into your chosen location. This package contains all of the files in the full Windows MSI Installer package, but does not include an installation program.

  For instructions on installing MySQL using the Zip file, see Section 5.6, “Installing MySQL on Microsoft Windows Using a noinstall Zip Archive”.

- The source distribution format contains all the code and support files for building the executables using the Visual Studio compiler system.

  For instructions on building MySQL from source on Windows, see Section 4.7, “Installing MySQL from Source on Windows”.

MySQL on Windows considerations:

- Large Table Support

  If you need tables with a size larger than 4GB, install MySQL on an NTFS or newer file system. Do not forget to use MAX_ROWS and AVG_ROW_LENGTH when you create tables. See CREATE TABLE Syntax.

- MySQL and Virus Checking Software
Virus-scanning software such as Norton/Symantec Anti-Virus on directories containing MySQL data and temporary tables can cause issues, both in terms of the performance of MySQL and the virus-scanning software misidentifying the contents of the files as containing spam. This is due to the fingerprinting mechanism used by the virus-scanning software, and the way in which MySQL rapidly updates different files, which may be identified as a potential security risk.

After installing MySQL Server, it is recommended that you disable virus scanning on the main directory (datadir) used to store your MySQL table data. There is usually a system built into the virus-scanning software to enable specific directories to be ignored.

In addition, by default, MySQL creates temporary files in the standard Windows temporary directory. To prevent the temporary files also being scanned, configure a separate temporary directory for MySQL temporary files and add this directory to the virus scanning exclusion list. To do this, add a configuration option for the tmpdir parameter to your my.ini configuration file. For more information, see Section 5.6.2, "Creating an Option File".

### 5.1 MySQL Installation Layout on Microsoft Windows

For MySQL 5.1 on Windows, the default installation directory is `C:\Program Files\MySQL\MySQL Server 5.1`. Some Windows users prefer to install in `C:\mysql`, the directory that formerly was used as the default. However, the layout of the subdirectories remains the same.

For MySQL 5.1.23 and earlier, all of the files are located within the parent directory, using the structure shown in the following table.

**Table 5.1 Installation Layout for Windows Using MySQL 5.1.23 and Earlier**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Client programs and the mysql server</td>
</tr>
<tr>
<td>data</td>
<td>Log files, databases</td>
</tr>
<tr>
<td>examples</td>
<td>Example programs and scripts</td>
</tr>
<tr>
<td>include</td>
<td>Include (header) files</td>
</tr>
<tr>
<td>lib</td>
<td>Libraries</td>
</tr>
<tr>
<td>scripts</td>
<td>Utility scripts</td>
</tr>
<tr>
<td>share</td>
<td>Miscellaneous support files, including error messages, character set files, sample configuration files, SQL for database installation</td>
</tr>
</tbody>
</table>

For MySQL 5.1.24 and later, the default location of data directory was changed. The remainder of the directory structure remains the same.

**Table 5.2 Installation Layout for Microsoft Windows using MySQL 5.1.24 and later**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin, scripts</td>
<td>mysql server, client and utility programs</td>
<td></td>
</tr>
<tr>
<td>%ALLUSERSPROFILE%\MySQL\MySQL Server 5.1\</td>
<td>Log files, databases (Windows XP, Windows Server 2003)</td>
<td>The Windows system variable %ALLUSERSPROFILE% defaults to C:\Documents and Settings\All Users \Application Data</td>
</tr>
</tbody>
</table>
Choosing the Installation Package for Microsoft Windows

For MySQL 5.1, there are multiple installation package formats to choose from when installing MySQL on Windows.

Note

Using MySQL Installer is the recommended installation method for Microsoft Windows users. The MySQL Server 5.1 release does not include its own MySQL Installer release, but a MySQL Installer version 5.5 and above can optionally install MySQL Server 5.1. Follow the standard Installing MySQL on Microsoft Windows Using MySQL Installer documentation but choose Custom Install after executing it. A MySQL Server 5.1 option will be available, and choosing it will cause MySQL Installer to download it for you.

Table 5.3 Microsoft Windows MySQL Installation package comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Essentials</th>
<th>Complete</th>
<th>Zip (No-install)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installer</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Directory-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MySQL Server Instance Config Wizard</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Test Suite</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MySQL Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MySQL Client Programs</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>C Headers/Libraries</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Embedded Server</td>
<td>No</td>
<td>Optional</td>
<td>Yes</td>
</tr>
<tr>
<td>Scripts and Examples</td>
<td>No</td>
<td>Optional</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In the above table:

- **Yes** indicates that the component is installed by default.
- **No** indicates that the component is not installed or included.
- **Optional** indicates that the component is included with the package, but not installed unless explicitly requested using the Custom installation mode.
Choosing the Installation Package for Microsoft Windows

The workflow for installing using the MSI installer is shown here:

**Figure 5.1 Installation Workflow for Windows Using MSI**

- Download MSI
- Run MSI
- Install MySQL
  - Run MySQL Instance Wizard
- Register
  - Use MySQL

The workflow for installing using the Zip package is shown here:

**Figure 5.2 Installation Workflow for Windows Using Zip**

- Download Zip
- Extract Zip
- Move/Rename Directory
- Post-install
- Install MySQL as Service
- Use MySQL

**Note**

For the Essentials and Complete packages in the MSI installer, you can select individual components to be installed by using the Custom mode, including disable the components configured for installation by default.
Full details on the components are suggested uses are provided here for reference:

- **Windows Essentials**: This package has a file name similar to `mysql-essential-5.1.73-win32.msi` and is supplied as a Microsoft Installer (MSI) package. The package includes the minimum set of files needed to install MySQL on Windows, including the MySQL Server Instance Config Wizard. This package does not include optional components such as the embedded server, developer headers and libraries or benchmark suite.

  To install using this package, see Section 5.4, “Installing MySQL on Microsoft Windows Using an MSI Package”.

- **Windows MSI Installer (Complete)**: This package has a file name similar to `mysql-5.1.73-win32.msi` and contains all files needed for a complete Windows installation, including the MySQL Server Instance Config Wizard. This package includes optional components such as the embedded server and benchmark suite.

  To install using this package, see Section 5.4, “Installing MySQL on Microsoft Windows Using an MSI Package”.

- **Without installer**: This package has a file name similar to `mysql-noinstall-5.1.73-win32.zip` and contains all the files found in the Complete install package, with the exception of the MySQL Server Instance Config Wizard. This package does not include an automated installer, and must be manually installed and configured.

  The Essentials package is recommended for most users. Both the Essentials and Complete distributions are available as an `.msi` file for use with the Windows Installer. The Noinstall distribution is packaged as a Zip archive. To use a Zip archive, you must have a tool that can unpack `.zip` files.

  When using the MSI installers you can automate the installation process. For more information, see Section 5.4.2, “Automating MySQL Installation on Microsoft Windows Using the MSI Package”. To automate the creation of a MySQL instance, see Section 5.5.13, “MySQL Server Instance Config Wizard: Creating an Instance from the Command Line”.

  Your choice of install package affects the installation process you must follow. If you choose to install either an Essentials or Complete install package, see Section 5.4, “Installing MySQL on Microsoft Windows Using an MSI Package”. If you choose to install a Noinstall archive, see Section 5.6, “Installing MySQL on Microsoft Windows Using a noinstall Zip Archive”.

5.3 **MySQL Notifier**

The MySQL Notifier is a tool that enables you to monitor and adjust the status of your local and remote MySQL Server instances through an indicator that resides in the system tray. The MySQL Notifier also gives quick access to several MySQL GUI tools (such as MySQL Workbench) through its context menu.

The MySQL Notifier is installed by MySQL Installer, and (by default) will start-up when Microsoft Windows is started.

**Note**

To install, download and execute the MySQL Installer, be sure the MySQL Notifier product is selected, then proceed with the installation. See the MySQL Installer manual for additional details.

For notes detailing the changes in each release of MySQL Notifier, see the MySQL Notifier Release Notes.

Visit the MySQL Notifier forum for additional MySQL Notifier help and support.
Features include:

- Start, Stop, and Restart instances of the MySQL Server.

- Automatically detects (and adds) new MySQL Server services. These are listed under Manage Monitored Items, and may also be configured.

- The Tray icon changes, depending on the status. It's green if all monitored MySQL Server instances are running, or red if at least one service is stopped. The **Update MySQL Notifier tray icon based on service status** option, which dictates this behavior, is enabled by default for each service.

- Links to other applications like MySQL Workbench, MySQL Installer, and the MySQL Utilities. For example, choosing Configure Instance will load the MySQL Workbench Server Administration window for that particular instance.

- If MySQL Workbench is also installed, then the Configure Instance and SQL Editor options are available for local (but not remote) MySQL instances.

- Monitoring of both local and remote MySQL instances.

  **Note**

  Remote monitoring is available since MySQL Notifier 1.1.0.

The MySQL Notifier resides in the system tray and provides visual status information for your MySQL Server instances. A green icon is displayed at the top left corner of the tray icon if the current MySQL Server is running, or a red icon if the service is stopped.

The MySQL Notifier automatically adds discovered MySQL Services on the local machine, and each service is saved and configurable. By default, the **Automatically add new services whose name contains** option is enabled and set to `mysql`. Related **Notifications Options** include being notified when new services are either discovered or experience status changes, and are also enabled by default. And uninstalling a service will also remove the service from the MySQL Notifier.

  **Note**

  The **Automatically add new services whose name contains** option default changed from `.*mysqld.*` to `mysql` in Notifier 1.1.0.

Clicking the system tray icon will reveal several options, as seen in the screenshots below:

The Service Instance menu is the main MySQL Notifier window, and enables you to Stop, Start, and Restart the MySQL Server.

**Figure 5.3 MySQL Notifier Service Instance menu**
The **Actions** menu includes several links to external applications (if they are installed), and a **Refresh Status** option to manually refresh the status of all monitored services (in both local and remote computers) and MySQL instances.

**Note**

The main menu will not show the Actions menu when there are no services being monitored by MySQL Notifier.

**Note**

The **Refresh Status** feature is available since MySQL Notifier 1.1.0.

**Figure 5.4 MySQL Notifier Actions menu**

The **Actions, Options** menu configures MySQL Notifier and includes options to:

- **Use colorful status icons**: Enables a colorful style of icons for the tray of the MySQL Notifier.

- **Run at Windows Startup**: Allows the application to be loaded when Microsoft Windows starts.

- **Automatically Check For Updates Every # Weeks**: Checks for a new version of MySQL Notifier, and runs this check every # weeks.

- **Automatically add new services whose name contains**: The text used to filter services and add them automatically to the monitored list of the local computer running MySQL Notifier, and on remote computers already monitoring Windows services. monitored services, and also filters the list of the Microsoft Windows services for the Add New Service dialog.

Prior to version 1.1.0, this option was named “Automatically add new services that match this pattern.”

- **Notify me when a service is automatically added**: Will display a balloon notification from the taskbar when a newly discovered service is added to the monitored services list.

- **Notify me when a service changes status**: Will display a balloon notification from the taskbar when a monitored service changes its status.
Figure 5.5 MySQL Notifier Options menu

The **Actions, Manage Monitored Items** menu enables you to configure the monitored services and MySQL instances. First, with the **Services** tab open:
Figure 5.6 MySQL Notifier Manage Services menu

The **Instances** tab is similar:
Adding a service or instance (after clicking Add in the Manage Monitored Items window) enables you to select a running Microsoft Windows service or instance connection, and configure MySQL Notifier to monitor it. Add a new service or instance by clicking service name from the list, then OK to accept. Multiple services and instances may be selected.
Figure 5.8 MySQL Notifier Adding new services

And instances:
5.3.1 Remote monitoring set up and installation instructions

The MySQL Notifier uses Windows Management Instrumentation (WMI) to manage and monitor services in remote computers running Windows XP or later. This guide explains how it works, and how to set up your system to monitor remote MySQL instances.

In order to configure WMI, it is important to understand that the underlying Distributed Component Object Model (DCOM) architecture is doing the WMI work. Specifically, MySQL Notifier is using asynchronous notification queries on remote Microsoft Windows hosts as .NET events. These events send an asynchronous callback to the computer running the MySQL Notifier so it knows when a service status has changed on the remote computer. Asynchronous notifications offer the best performance compared to semisynchronous notifications or synchronous notifications that use timers.
Asynchronous notifications requires the remote computer to send a callback to the client computer (thus opening a reverse connection), so the Windows Firewall and DCOM settings must be properly configured for the communication to function properly.

**Figure 5.10 MySQL Notifier Distributed Component Object Model (DCOM)**

Most of the common errors thrown by asynchronous WMI notifications are related to Windows Firewall blocking the communication, or to DCOM / WMI settings not being set up properly. For a list of common errors with solutions, see Common Errors.

The following steps are required to make WMI function. These steps are divided between two machines. A single host computer that runs MySQL Notifier (Computer A), and multiple remote machines that are being monitored (Computer B).

**Computer running MySQL Notifier (Computer A)**

1. Allow for remote administration by either editing the Group Policy Editor, or using NETSH:

   Using the **Group Policy Editor**:
   
   a. Click **Start**, click **Run**, type `GPEDIT.MSC`, and then click **OK**.
   
   b. Under the **Local Computer Policy** heading, double-click **Computer Configuration**.
   
   c. Double-click **Administrative Templates**, then **Network, Network Connections**, and then **Windows Firewall**.
   
   d. If the computer is in the domain, then double-click **Domain Profile**; otherwise, double-click **Standard Profile**.
   
   e. Click **Windows Firewall: Allow inbound remote administration exception**.
   
   f. On the Action menu either select **Edit**, or double-click the selection from the previous step.
   
   g. Check the **Enabled** radio button, and then click **OK**.

   Using the **NETSH** command:

   ```
   Note
   
   The "netsh firewall" command is deprecated as of Microsoft Server 2008 and Vista, and replaced with "netsh advfirewall firewall".
   ```

   a. Open a command prompt window with Administrative rights (you can right-click the Command Prompt icon and click **Run as Administrator**).
   
   b. Execute the following command:
Remote monitoring set up and installation instructions

NETSH advfirewall firewall set service RemoteAdmin enable

2. Open the DCOM port TCP 135:
   
   a. Open a command prompt window with Administrative rights (you can right-click the Command Prompt icon and click **Run as Administrator**) .
   
   b. Execute the following command:

   NETSH advfirewall firewall add portopening protocol=tcp port=135 name=DCOM_TCP135

3. Add the client application which contains the sink for the callback (**MySqlNotifier.exe**) to the Windows Firewall Exceptions List (use either the Windows Firewall configuration or **NETSH**):

   Using the Windows Firewall configuration:
   
   a. In the Control Panel, double-click **Windows Firewall**.
   
   b. In the Windows Firewall window's left panel, click **Allow a program or feature through Windows Firewall**.
   
   c. In the Allowed Programs window, click **Change Settings**.
   
   d. If **MySqlNotifier.exe** is in the Allowed programs and features list, make sure it is checked for the type of networks the computer connects to (Private, Public or both).
   
   e. If **MySqlNotifier.exe** is not in the list, click **Allow another program...**.
   
   f. In the **Add a Program** window, select the **MySqlNotifier.exe** if it exists in the Programs list, otherwise click Browse... and go to the directory where **MySqlNotifier.exe** was installed to select it, then click **Add**.
   
   g. Make sure **MySqlNotifier.exe** is checked for the type of networks the computer connects to (Private, Public or both).

   Using the **NETSH** command:
   
   a. Open a command prompt window with Administrative rights (you can right-click the Command Prompt icon and click **Run as Administrator**).
   
   b. Execute the following command, where you change "**[YOUR_INSTALL_DIRECTORY]**":

   NETSH advfirewall firewall add allowedprogram program=[YOUR_INSTALL_DIRECTORY]\MySqlNotifier.exe name=MySqlNotifier

4. If Computer B is either a member of **WORKGROUP** or is in a different domain that is untrusted by Computer A, then the callback connection (Connection 2) is created as an Anonymous connection. To grant Anonymous connections DCOM Remote Access permissions:

   a. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**.
   
   b. In the Component Services dialog box, expand Component Services, expand Computers, and then right-click **My Computer** and click **Properties**.
   
   c. In the My Computer Properties dialog box, click the **COM Security** tab.
d. Under Access Permissions, click **Edit Limits**.

e. In the Access Permission dialog box, select **ANONYMOUS LOGON name** in the Group or user names box. In the Allow column under Permissions for User, select **Remote Access**, and then click OK.

**Monitored Remote Computer (Computer B)**

If the user account that is logged into the computer running the MySQL Notifier (Computer A) is a local administrator on the remote computer (Computer B), such that the same account is an administrator on Computer B, you can skip to the "Allow for remote administration" step.

Setting DCOM security to allow a non-administrator user to access a computer remotely:

1. Grant "DCOM remote launch" and activation permissions for a user or group:

   a. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**.

   b. In the Component Services dialog box, expand Component Services, expand Computers, and then right-click **My Computer** and click **Properties**.

   c. In the My Computer Properties dialog box, click the **COM Security** tab.

   d. Under Access Permissions, click **Edit Limits**.

   e. In the **Launch Permission** dialog box, follow these steps if your name or your group does not appear in the Groups or user names list:

      i. In the **Launch Permission** dialog box, click **Add**.

      ii. In the Select Users, Computers, or Groups dialog box, add your name and the group in the "Enter the object names to select" box, and then click **OK**.

   f. In the **Launch Permission** dialog box, select your user and group in the Group or user names box. In the Allow column under Permissions for User, select **Remote Launch**, select **Remote Activation**, and then click **OK**.

Grant DCOM remote access permissions:

a. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**.

b. In the Component Services dialog box, expand Component Services, expand Computers, and then right-click **My Computer** and click **Properties**.

c. In the My Computer Properties dialog box, click the **COM Security** tab.

d. Under Access Permissions, click **Edit Limits**.

e. In the Access Permission dialog box, select **ANONYMOUS LOGON name** in the Group or user names box. In the Allow column under Permissions for User, select **Remote Access**, and then click **OK**.

2. Allowing non-administrator users access to a specific WMI namespace:

   a. In the Control Panel, double-click **Administrative Tools**.

   b. In the Administrative Tools window, double-click **Computer Management**.
Remote monitoring set up and installation instructions

c. In the Computer Management window, expand the **Services and Applications** tree and double-click the **WMI Control**.
d. Right-click the WMI Control icon and select **Properties**.
e. In the WMI Control Properties window, click the **Security** tab.
f. In the Security tab, select the namespace and click **Security**.
g. Locate the appropriate account and check **Remote Enable in the Permissions list**.

3. **Allow for remote administration by either editing the Group Policy Editor or using NETSH:**

   **Using the Group Policy Editor:**
   a. Click **Start**, click **Run**, type **GPEDIT.MSC**, and then click **OK**.
   b. Under the Local Computer Policy heading, double-click **Computer Configuration**.
   c. Double-click **Administrative Templates**, then **Network, Network Connections**, and then **Windows Firewall**.
   d. If the computer is in the domain, then double-click **Domain Profile**; otherwise, double-click **Standard Profile**.
   e. Click **Windows Firewall: Allow inbound remote administration exception**.
   f. On the Action menu either select **Edit**, or double-click the selection from the previous step.
   g. Check the **Enabled** radio button, and then click **OK**.

   **Using the NETSH command:**
   a. Open a command prompt window with Administrative rights (you can right-click the Command Prompt icon and click Run as Administrator).
   b. Execute the following command:

   ```
   NETSH advfirewall firewall set service RemoteAdmin enable
   ```

4. **Now, be sure the user you are logging in with uses the Name value and not the Full Name value:**

   a. In the **Control Panel**, double-click **Administrative Tools**.
   b. In the **Administrative Tools** window, double-click **Computer Management**.
   c. In the Computer Management window, expand the **System Tools then Local Users and Groups**.
   d. Click the **Users** node, and on the right side panel locate your user and make sure it uses the **Name** value to connect, and not the **Full Name** value.

5. **If the remote computer is running on Windows XP Professional, make sure that remote logins are not being forcefully changed to the guest account user (also known as ForceGuest), which is enabled by default on computers that are not attached to a domain.**

   a. Click Start, click Run, type **SECPOL.MSC**, and then click **OK**.
b. Under the **Local Policies** node, double-click **Security Options**.
c. Select **Network Access: Sharing and security model for local accounts** and save.

### Common Errors

- **0x80070005**
  - DCOM Security was not configured properly (see Computer B, the Setting DCOM security... step).
  - The remote computer (Computer B) is a member of WORKGROUP or is in a domain that is untrusted by the client computer (Computer A) (see Computer A, the Grant Anonymous connections DCOM Remote Access permissions step).

- **0x8007000E**
  - The remote computer (Computer B) is a member of WORKGROUP or is in a domain that is untrusted by the client computer (Computer A) (see Computer A, the Grant Anonymous connections DCOM Remote Access permissions step).

- **0x80041003**
  - Access to the remote WMI namespace was not configured properly (see Computer B, the Allowing non-administrator users access to a specific WMI namespace step).

- **0x800706BA**
  - The DCOM port is not open on the client computers (Computer A) firewall. See the Open the DCOM port TCP 135 step for Computer A.
  - The remote computer (Computer B) is inaccessible because its network location is set to Public. Make sure you can access it through the Windows Explorer.

### 5.4 Installing MySQL on Microsoft Windows Using an MSI Package

The MSI package is designed to install and configure MySQL in such a way that you can immediately get started using MySQL.

The MySQL Installation Wizard and MySQL Configuration Wizard are available in the Essentials and Complete install packages. They are recommended for most standard MySQL installations. Exceptions include users who need to install multiple instances of MySQL on a single server host and advanced users who want complete control of server configuration.

- For information on installing using the GUI MSI installer process, see Section 5.4.1, “Using the MySQL Installation Wizard for Microsoft Windows”.

- For information on installing using the command line using the MSI package, see Section 5.4.2, “Automating MySQL Installation on Microsoft Windows Using the MSI Package”.

- If you have previously installed MySQL using the MSI package and want to remove MySQL, see Section 5.4.3, “Removing MySQL When Installed from the MSI Package”.

The workflow sequence for using the installer is shown in the figure below:
Microsoft Windows XP and later include a firewall which specifically blocks ports. If you plan on using MySQL through a network port then you should open and create an exception for this port before performing the installation. To check and if necessary add an exception to the firewall settings:

1. First ensure that you are logged in as an Administrator or a user with Administrator privileges.
2. Go to the Control Panel, and double click the Windows Firewall icon.
3. Choose the Allow a program through Windows Firewall option and click the Add port button.
4. Enter MySQL into the Name text box and 3306 (or the port of your choice) into the Port number text box.
5. Also ensure that the **TCP** protocol radio button is selected.

6. If you wish, you can also limit access to the MySQL server by choosing the **Change scope** button.

7. Confirm your choices by clicking the **OK** button.

Additionally, when running the MySQL Installation Wizard on Windows Vista or newer, ensure that you are logged in as a user with administrative rights.

**Note**

When using Windows Vista or newer, you may want to disable User Account Control (UAC) before performing the installation. If you do not do so, then MySQL may be identified as a security risk, which will mean that you need to enable MySQL. You can disable the security checking by following these instructions:

1. Open **Control Panel**.

2. Under the **User Accounts and Family Safety**, select **Add or remove user accounts**.

3. Click the **Got to the main User Accounts page** link.

4. Click on **Turn User Account Control on or off**. You may be prompted to provide permission to change this setting. Click **Continue**.

5. Deselect or uncheck the check box next to **Use User Account Control (UAC) to help protect your computer**. Click **OK** to save the setting.

You will need to restart to complete the process. Click **Restart Now** to reboot the machine and apply the changes. You can then follow the instructions below for installing Windows.

### 5.4.1 Using the MySQL Installation Wizard for Microsoft Windows

MySQL Installation Wizard is an installer for the MySQL server that uses the latest installer technologies for Microsoft Windows. The MySQL Installation Wizard, in combination with the MySQL Config Wizard, enables a user to install and configure a MySQL server that is ready for use immediately after installation.

The MySQL Installation Wizard uses the standard Microsoft Installer Engine (MSI) system is the standard installer for all MySQL server distributions. See Section 5.4.1.6, “MySQL Installation Wizard: Upgrading MySQL”, for more information on upgrading from a previous version.

If you are upgrading an installation from MySQL 5.1.31 or earlier to MySQL 5.1.32 or later, read the notes provided in Section 5.4.1.6, “MySQL Installation Wizard: Upgrading MySQL”.

The Microsoft Windows Installer Engine was updated with the release of Windows XP; those using a previous version of Windows can reference this Microsoft Knowledge Base article for information on upgrading to the latest version of the Windows Installer Engine.

In addition, Microsoft has introduced the WiX (Windows Installer XML) toolkit. This is the first highly acknowledged Open Source project from Microsoft. We have switched to WiX because it is an Open Source project and it enables us to handle the complete Windows installation process in a flexible manner using scripts.
Improving the MySQL Installation Wizard depends on the support and feedback of users like you. If you find that the MySQL Installation Wizard is lacking some feature important to you, or if you discover a bug, please report it in our bugs database using the instructions given in How to Report Bugs or Problems.

5.4.1.1 MySQL Installation Wizard: Downloading and Starting

The MySQL installation packages can be downloaded from http://dev.mysql.com/downloads/. If the package you download is contained within a Zip archive, you need to extract the archive first.

The process for starting the wizard depends on the contents of the installation package you download. If there is a setup.exe file present, double-click it to start the installation process. If there is an .msi file present, double-click it to start the installation process.

5.4.1.2 MySQL Installation Wizard: Choosing an Install Type

There are three installation types available: Typical, Complete, and Custom.

The Typical installation type installs the MySQL server, the mysql command-line client, and the command-line utilities. The command-line clients and utilities include mysqldump, myisamchk, and several other tools to help you manage the MySQL server.

The Complete installation type installs all components included in the installation package. The full installation package includes components such as the embedded server library, the benchmark suite, support scripts, and documentation.

The Custom installation type gives you complete control over which packages you wish to install and the installation path that is used. See Section 5.4.1.3, “MySQL Installation Wizard: The Custom Install Dialog”, for more information on performing a custom install.

If you choose the Typical or Complete installation types and click the Next button, you advance to the confirmation screen to verify your choices and begin the installation. If you choose the Custom installation type and click the Next button, you advance to the custom installation dialog, described in Section 5.4.1.3, “MySQL Installation Wizard: The Custom Install Dialog”.

5.4.1.3 MySQL Installation Wizard: The Custom Install Dialog

If you wish to change the installation path or the specific components that are installed by the MySQL Installation Wizard, choose the Custom installation type.

A tree view on the left side of the custom install dialog lists all available components. Components that are not installed have a red X icon; components that are installed have a gray icon. To change whether a component is installed, click that component's icon and choose a new option from the drop-down list that appears.

You can change the default installation path by clicking the Change... button to the right of the displayed installation path.

After choosing your installation components and installation path, click the Next button to advance to the confirmation dialog.

5.4.1.4 MySQL Installation Wizard: The Confirmation Dialog

Once you choose an installation type and optionally choose your installation components, you advance to the confirmation dialog. Your installation type and installation path are displayed for you to review.

To install MySQL if you are satisfied with your settings, click the Install button. To change your settings, click the Back button. To exit the MySQL Installation Wizard without installing MySQL, click the Cancel button.
In MySQL 5.1.47 and earlier, after installation is complete, you have the option of registering with the MySQL web site. Registration gives you access to post in the MySQL forums at forums.mysql.com, along with the ability to report bugs at bugs.mysql.com and to subscribe to our newsletter.

The final screen of the installer provides a summary of the installation and gives you the option to launch the MySQL Config Wizard, which you can use to create a configuration file, install the MySQL service, and configure security settings.

### 5.4.1.5 MySQL Installation Wizard: Changes Made

Once you click the Install button, the MySQL Installation Wizard begins the installation process and makes certain changes to your system which are described in the sections that follow.

**Changes to the Registry**

The MySQL Installation Wizard creates one Windows registry key in a typical install situation, located in `HKEY_LOCAL_MACHINE\SOFTWARE\MySQL AB`. For 64-bit Windows, the registry location is `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\MySQL AB`. A server version specific entry will be created for each release series of MySQL that you install.

The MySQL Installation Wizard creates a key named after the release series of the server that is being installed, such as MySQL Server 5.1. It contains two string values, Location and Version. The Location string contains the path to the installation directory. In a default installation it contains `C:\Program Files\MySQL\MySQL Server 5.1\`. The Version string contains the release number. For example, for an installation of MySQL Server 5.1.73, the key contains a value of 5.1.73.

These registry keys are used to help external tools identify the installed location of the MySQL server, preventing a complete scan of the hard-disk to determine the installation path of the MySQL server. The registry keys are not required to run the server, and if you install MySQL using the noinstall Zip archive, the registry keys are not created.

**Changes to the Start Menu**

The MySQL Installation Wizard creates a new entry in the Windows Start menu under a common MySQL menu heading named after the release series of MySQL that you have installed. For example, if you install MySQL 5.1, the MySQL Installation Wizard creates a MySQL Server 5.1 section in the Start menu.

The following entries are created within the new Start menu section:

- **MySQL Command-Line Client**: This is a shortcut to the mysql command-line client and is configured to connect as the root user. The shortcut prompts for a root user password when you connect.

- **MySQL Server Instance Config Wizard**: This is a shortcut to the MySQL Config Wizard. Use this shortcut to configure a newly installed server, or to reconfigure an existing server.

- **MySQL Documentation**: This is a link to the MySQL server documentation that is stored locally in the MySQL server installation directory. This option is not available when the MySQL server is installed using the Essentials installation package.

**Changes to the File System**

The MySQL Installation Wizard by default installs the MySQL 5.1 server to `C:\Program Files\MySQL\MySQL Server 5.1`, where Program Files is the default location for applications in your system, and 5.1 is the release series of your MySQL server. This is the recommended location for the MySQL server, replacing the former default location `C:\mysql`.
By default, all MySQL applications are stored in a common directory at $C:\Program Files\MySQL$, where Program Files is the default location for applications in your Windows installation. A typical MySQL installation on a developer machine might look like this:

```
C:\Program Files\MySQL\MySQL Server 5.1
C:\Program Files\MySQL\MySQL Workbench 5.1 OSS
```

This approach makes it easier to manage and maintain all MySQL applications installed on a particular system.

In MySQL 5.1.23 and earlier, the default location for the data files used by MySQL is located within the corresponding MySQL Server installation directory. For MySQL 5.1.24 and later, the default location of the data directory is the AppData directory configured for the user that installed the MySQL application.

### 5.4.1.6 MySQL Installation Wizard: Upgrading MySQL

The MySQL Installation Wizard can perform server upgrades automatically using the upgrade capabilities of MSI. That means you do not need to remove a previous installation manually before installing a new release. The installer automatically shuts down and removes the previous MySQL service before installing the new version.

Automatic upgrades are available only when upgrading between installations that have the same major and minor version numbers. For example, you can upgrade automatically from MySQL 5.1.5 to MySQL 5.1.6, but not from MySQL 5.0 to MySQL 5.1.

In MySQL 5.1.32 and later, the EXE version of the MSI installer packages were removed. When upgrading an existing MySQL installation from the old EXE based installer to the MSI based installer, please keep the following notes in mind:

- The MSI installer will not identify an existing installation that was installed using the old EXE installer. This means that the installer will not stop the existing server, or detect that the existing password is required before installing the new version. To work around this:
  1. Stop the current server manually using `net stop` or `mysqladmin shutdown`.
  2. Remove the existing installation manually by using the Add/Remove Programs control panel. This will keep the existing configuration and data files, as these are not removed automatically.
  3. Install the new version of MySQL using the MSI installer. When running the installation, skip updating the security by deselecting the check box on the security screen.
  4. Complete the installation, and then start the server again. You should be able to login with your existing user and password credentials.

- You can only upgrade the version and release using the MSI installer. For example, you can upgrade an open source installation with an open source installer. You cannot upgrade an open source installation using the enterprise installer.

See Section 5.9, “Upgrading MySQL Server on Microsoft Windows”.

### 5.4.2 Automating MySQL Installation on Microsoft Windows Using the MSI Package

The Microsoft Installer (MSI) supports a both a quiet and a passive mode that can be used to install MySQL automatically without requiring intervention. You can use this either in scripts to automatically
Removing MySQL When Installed from the MSI Package

install MySQL or through a terminal connection such as Telnet where you do not have access to the standard Windows user interface. The MSI packages can also be used in combination with Microsoft’s Group Policy system (part of Windows Server 2003 and Windows Server 2008) to install MySQL across multiple machines.

To install MySQL from one of the MSI packages automatically from the command line (or within a script), you need to use the `msiexec.exe` tool. For example, to perform a quiet installation (which shows no dialog boxes or progress):

```
shell> msiexec /i mysql-5.1.73.msi /quiet
```

The `/i` indicates that you want to perform an installation. The `/quiet` option indicates that you want no interactive elements.

To provide a dialog box showing the progress during installation, and the dialog boxes providing information on the installation and registration of MySQL, use `/passive` mode instead of `/quiet`:

```
shell> msiexec /i mysql-5.1.73.msi /passive
```

Regardless of the mode of the installation, installing the package in this manner performs a 'Typical' installation, and installs the default components into the standard location.

You can also use this method to uninstall MySQL by using the `/uninstall` or `/x` options:

```
shell> msiexec /x mysql-5.1.73.msi /uninstall
```

To install MySQL and configure a MySQL instance from the command line, see Section 5.5.13, “MySQL Server Instance Config Wizard: Creating an Instance from the Command Line”.

For information on using MSI packages to install software automatically using Group Policy, see How to use Group Policy to remotely install software in Windows Server 2003.

### 5.4.3 Removing MySQL When Installed from the MSI Package

To uninstall a MySQL where you have used the MSI packages, you must use the `Add/Remove Programs` tool within `Control Panel`. To do this:

1. Right-click the `start` menu and choose `Control Panel`.

2. If the Control Panel is set to category mode (you will see `Pick a category` at the top of the `Control Panel` window), double-click `Add or Remove Programs`. If the Control is set to classic mode, double-click the `Add or Remove Programs` icon.

3. Find MySQL in the list of installed software. MySQL Server is installed against release series numbers (MySQL 5.0, MySQL 5.1, etc.). Select the version that you want to remove and click `Remove`.

4. You will be prompted to confirm the removal. Click `Yes` to remove MySQL.

When MySQL is removed using this method, only the installed components are removed. Any database information (including the tables and data), import or export files, log files, and binary logs produced during execution are kept in their configured location.

If you try to install MySQL again the information will be retained and you will be prompted to enter the password configured with the original installation.
If you want to delete MySQL completely:

- Delete the associated data directory. On Windows XP and Windows Server 2003, before MySQL 5.1.24, the default data directory would be located within the MySQL installation directory. On MySQL 5.1.24 and later, the default data directory is the configured AppData directory, which is **C:\Documents and Settings\All Users\Application Data\MySQL** by default.

- On Windows 7 and Windows Server 2008, the default data directory location is **C:\ProgramData\Mysql**.

  **Note**

  The **C:\ProgramData** directory is hidden by default. You must change your folder options to view the hidden file. Choose **Organize**, Folder and search options, **Show hidden folders**.

### 5.5 Using the MySQL Server Instance Config Wizard

The MySQL Server Instance Config Wizard helps automate the process of configuring your server. It creates a custom MySQL configuration file (**my.ini** or **my.cnf**) by asking you a series of questions and then applying your responses to a template to generate the configuration file that is tuned to your installation.

The complete and essential MSI installation packages include the MySQL Server Instance Config Wizard in the MySQL 5.1 server. The MySQL Server Instance Config Wizard is only available for Windows.

The workflow sequence for using the MySQL Server Instance Config Wizard is shown in the figure below:
Using the MySQL Server Instance Config Wizard

Figure 5.12 MySQL Server Instance Config Wizard Workflow:

- Confirm Removal
- Start Wizard
  - New Installation
  - Detailed Installation
  - Configuration Type
    - Standard Installation
    - InnoDB Settings
    - Connection Settings
    - Networking Options
    - Character Set Options
    - Windows Options
    - Security Options
  - Execute
  - Completed

Flow:
- Remove Instance
- Reconfigure Instance
- Non-transactional Database
5.5.1 Starting the MySQL Server Instance Config Wizard

The MySQL Server Instance Config Wizard is normally started as part of the installation process. You should only need to run the MySQL Server Instance Config Wizard again when you need to change the configuration parameters of your server.

If you chose not to open a port prior to installing MySQL on Windows Vista or newer, you can choose to use the MySQL Server Instance Config Wizard after installation. However, you must open a port in the Windows Firewall. To do this see the instructions given in Section 5.4.1.1, “MySQL Installation Wizard: Downloading and Starting”. Rather than opening a port, you also have the option of adding MySQL as a program that bypasses the Windows Firewall. One or the other option is sufficient—you need not do both. Additionally, when running the MySQL Server Config Wizard on Windows Vista or newer, ensure that you are logged in as a user with administrative rights.

You can launch the MySQL Config Wizard by clicking the MySQL Server Instance Config Wizard entry in the MySQL section of the Windows Start menu.

Alternatively, you can navigate to the bin directory of your MySQL installation and launch the MySQLInstanceConfig.exe file directly.

The MySQL Server Instance Config Wizard places the my.ini file in the installation directory for the MySQL server. This helps associate configuration files with particular server instances.
To ensure that the MySQL server knows where to look for the `my.ini` file, an argument similar to this is passed to the MySQL server as part of the service installation:

```bash
--defaults-file="C:\Program Files\MySQL\MySQL Server 5.1\my.ini"
```

Here, `C:\Program Files\MySQL\MySQL Server 5.1` is replaced with the installation path to the MySQL Server. The `--defaults-file` option instructs the MySQL server to read the specified file for configuration options when it starts.

Apart from making changes to the `my.ini` file by running the MySQL Server Instance Config Wizard again, you can modify it by opening it with a text editor and making any necessary changes. You can also modify the server configuration with the [http://www.mysql.com/products/administrator/](http://www.mysql.com/products/administrator/) utility. For more information about server configuration, see Server Command Options.

MySQL clients and utilities such as the `mysql` and `mysqldump` command-line clients are not able to locate the `my.ini` file located in the server installation directory. To configure the client and utility applications, create a new `my.ini` file in the Windows installation directory (for example, `C:\WINDOWS`).

Under Windows Server 2003, Windows Server 2000, Windows XP, and Windows Vista, MySQL Server Instance Config Wizard will configure MySQL to work as a Windows service. To start and stop MySQL you use the Services application that is supplied as part of the Windows Administrator Tools.

### 5.5.2 MySQL Server Instance Config Wizard: Choosing a Maintenance Option

If the MySQL Server Instance Config Wizard detects an existing configuration file, you have the option of either reconfiguring your existing server, or removing the server instance by deleting the configuration file and stopping and removing the MySQL service.

To reconfigure an existing server, choose the Re-configure Instance option and click the Next button. Any existing configuration file is not overwritten, but renamed (within the same directory) using a timestamp (Windows) or sequential number (Linux). To remove the existing server instance, choose the Remove Instance option and click the Next button.

If you choose the Remove Instance option, you advance to a confirmation window. Click the Execute button. The MySQL Server Config Wizard stops and removes the MySQL service, and then deletes the configuration file. The server installation and its `data` folder are not removed.

If you choose the Re-configure Instance option, you advance to the Configuration Type dialog where you can choose the type of installation that you wish to configure.

### 5.5.3 MySQL Server Instance Config Wizard: Choosing a Configuration Type

When you start the MySQL Server Instance Config Wizard for a new MySQL installation, or choose the Re-configure Instance option for an existing installation, you advance to the Configuration Type dialog.
There are two configuration types available: Detailed Configuration and Standard Configuration. The Standard Configuration option is intended for new users who want to get started with MySQL quickly without having to make many decisions about server configuration. The Detailed Configuration option is intended for advanced users who want more fine-grained control over server configuration.

If you are new to MySQL and need a server configured as a single-user developer machine, the Standard Configuration should suit your needs. Choosing the Standard Configuration option causes the MySQL Config Wizard to set all configuration options automatically with the exception of Service Options and Security Options.

The Standard Configuration sets options that may be incompatible with systems where there are existing MySQL installations. If you have an existing MySQL installation on your system in addition to the installation you wish to configure, the Detailed Configuration option is recommended.

To complete the Standard Configuration, please refer to the sections on Service Options and Security Options in Section 5.5.10, “MySQL Server Instance Config Wizard: The Service Options Dialog”, and Section 5.5.11, “MySQL Server Instance Config Wizard: The Security Options Dialog”, respectively.

### 5.5.4 MySQL Server Instance Config Wizard: The Server Type Dialog

There are three different server types available to choose from. The server type that you choose affects the decisions that the MySQL Server Instance Config Wizard makes with regard to memory, disk, and processor usage.
• Developer Machine: Choose this option for a typical desktop workstation where MySQL is intended only for personal use. It is assumed that many other desktop applications are running. The MySQL server is configured to use minimal system resources.

• Server Machine: Choose this option for a server machine where the MySQL server is running alongside other server applications such as FTP, email, and Web servers. The MySQL server is configured to use a moderate portion of the system resources.

• Dedicated MySQL Server Machine: Choose this option for a server machine that is intended to run only the MySQL server. It is assumed that no other applications are running. The MySQL server is configured to use all available system resources.

Note

By selecting one of the preconfigured configurations, the values and settings of various options in your `my.cnf` or `my.ini` will be altered accordingly. The default values and options as described in the reference manual may therefore be different to the options and values that were created during the execution of the Config Wizard.
5.5.5 MySQL Server Instance Config Wizard: The Database Usage Dialog

The Database Usage dialog enables you to indicate the storage engines that you expect to use when creating MySQL tables. The option you choose determines whether the InnoDB storage engine is available and what percentage of the server resources are available to InnoDB.

- Multifunctional Database: This option enables both the InnoDB and MyISAM storage engines and divides resources evenly between the two. This option is recommended for users who use both storage engines on a regular basis.

- Transactional Database Only: This option enables both the InnoDB and MyISAM storage engines, but dedicates most server resources to the InnoDB storage engine. This option is recommended for users who use InnoDB almost exclusively and make only minimal use of MyISAM.

- Non-Transactional Database Only: This option disables the InnoDB storage engine completely and dedicates all server resources to the MyISAM storage engine. This option is recommended for users who do not use InnoDB.

The Config Wizard uses a template to generate the server configuration file. The Database Usage dialog sets one of the following option strings:

<table>
<thead>
<tr>
<th>Option String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multifunctional Database:</td>
</tr>
<tr>
<td>Transactional Database Only:</td>
</tr>
<tr>
<td>Non-Transactional Database Only:</td>
</tr>
</tbody>
</table>
When these options are processed through the default template (my-template.ini) the result is:

```
Multifunctional Database:
default-storage-engine=InnoDB
_myisam_pct=50

Transactional Database Only:
default-storage-engine=InnoDB
_myisam_pct=5

Non-Transactional Database Only:
default-storage-engine=MyISAM
_myisam_pct=100
skip-innodb
```

The `_myisam_pct` value is used to calculate the percentage of resources dedicated to MyISAM. The remaining resources are allocated to InnoDB.

5.5.6 MySQL Server Instance Config Wizard: The InnoDB Tablespace Dialog

Some users may want to locate the InnoDB tablespace files in a different location than the MySQL server data directory. Placing the tablespace files in a separate location can be desirable if your system has a higher capacity or higher performance storage device available, such as a RAID storage system.
To change the default location for the InnoDB tablespace files, choose a new drive from the drop-down list of drive letters and choose a new path from the drop-down list of paths. To create a custom path, click the ... button.

If you are modifying the configuration of an existing server, you must click the Modify button before you change the path. In this situation you must move the existing tablespace files to the new location manually before starting the server.

### 5.5.7 MySQL Server Instance Config Wizard: The Concurrent Connections Dialog

To prevent the server from running out of resources, it is important to limit the number of concurrent connections to the MySQL server that can be established. The Concurrent Connections dialog enables you to choose the expected usage of your server, and sets the limit for concurrent connections accordingly. It is also possible to set the concurrent connection limit manually.

- **Decision Support (DSS)/OLAP**: Choose this option if your server does not require a large number of concurrent connections. The maximum number of connections is set at 100, with an average of 20 concurrent connections assumed.

- **Online Transaction Processing (OLTP)**: Choose this option if your server requires a large number of concurrent connections. The maximum number of connections is set at 500.

- **Manual Setting**: Please enter the approximate number of concurrent connections. The default is 15.
• Manual Setting: Choose this option to set the maximum number of concurrent connections to the server manually. Choose the number of concurrent connections from the drop-down box provided, or enter the maximum number of connections into the drop-down box if the number you desire is not listed.

5.5.8 MySQL Server Instance Config Wizard: The Networking and Strict Mode Options Dialog

Use the Networking Options dialog to enable or disable TCP/IP networking and to configure the port number that is used to connect to the MySQL server.

TCP/IP networking is enabled by default. To disable TCP/IP networking, uncheck the box next to the Enable TCP/IP Networking option.

Port 3306 is used by default. To change the port used to access MySQL, choose a new port number from the drop-down box or type a new port number directly into the drop-down box. If the port number you choose is in use, you are prompted to confirm your choice of port number.

Set the Server SQL Mode to either enable or disable strict mode. Enabling strict mode (default) makes MySQL behave more like other database management systems. If you run applications that rely on MySQL’s old “forgiving” behavior, make sure to either adapt those applications or to disable strict mode. For more information about strict mode, see Server SQL Modes.
5.5.9 MySQL Server Instance Config Wizard: The Character Set Dialog

The MySQL server supports multiple character sets and it is possible to set a default server character set that is applied to all tables, columns, and databases unless overridden. Use the Character Set dialog to change the default character set of the MySQL server.

- Standard Character Set: Choose this option if you want to use latin1 as the default server character set. latin1 is used for English and many Western European languages.

- Best Support For Multilingualism: Choose this option if you want to use utf8 as the default server character set. This is a Unicode character set that can store characters from many different languages.

- Manual Selected Default Character Set / Collation: Choose this option if you want to pick the server's default character set manually. Choose the desired character set from the provided drop-down list.

5.5.10 MySQL Server Instance Config Wizard: The Service Options Dialog

On Windows platforms, the MySQL server can be installed as a Windows service. When installed this way, the MySQL server can be started automatically during system startup, and even restarted automatically by Windows in the event of a service failure.

The MySQL Server Instance Config Wizard installs the MySQL server as a service by default, using the service name MySQL. If you do not wish to install the service, uncheck the box next to the Install As
Windows Service option. You can change the service name by picking a new service name from the drop-down box provided or by entering a new service name into the drop-down box.

**Note**

Service names can include any legal character except forward (/) or backward (\) slashes, and must be less than 256 characters long.

**Warning**

If you are installing multiple versions of MySQL onto the same machine, you must choose a different service name for each version that you install. If you do not choose a different service for each installed version then the service manager information will be inconsistent and this will cause problems when you try to uninstall a previous version.

If you have already installed multiple versions using the same service name, you must manually edit the contents of the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services` parameters within the Windows registry to update the association of the service name with the correct server version.

Typically, when installing multiple versions you create a service name based on the version information. For example, you might install MySQL 5.x as `mysql5`, or specific versions such as MySQL 5.1.30 as `mysql50130`.

To install the MySQL server as a service but not have it started automatically at startup, uncheck the box next to the Launch the MySQL Server Automatically option.

### 5.5.11 MySQL Server Instance Config Wizard: The Security Options Dialog

The content of the security options portion of the MySQL Server Instance Configuration Wizard will depend on whether this is a new installation, or modifying an existing installation.

- **Setting the root password for a new installation**

  *It is strongly recommended that you set a root password for your MySQL server*, and the MySQL Server Instance Config Wizard requires by default that you do so. If you do not wish to set a root password, uncheck the box next to the Modify Security Settings option.

**Note**

If you have previously installed MySQL, but not deleted the data directory associated with the previous installation, you may be prompted to provide the current root password. The password will be the one configured with your old data directory. If you do not want to use this data, or do not know the root password, you should cancel the installation, delete the previous installation data, and then restart the installation process. For more information on deleting MySQL data on Microsoft Windows, see [Section 5.4.3, “Removing MySQL When Installed from the MSI Package”](#).
• To set the root password, enter the desired password into both the New root password and Confirm boxes.

**Setting the root password for an existing installation**

If you are modifying the configuration of an existing configuration, or you are installing an upgrade and the MySQL Server Instance Configuration Wizard has detected an existing MySQL system, then you must enter the existing password for root before changing the configuration information.
If you want to change the current root password, enter the desired new password into both the New root password and Confirm boxes.

To permit root logins from across the network, check the box next to the Enable root access from remote machines option. This decreases the security of your root account.

To create an anonymous user account, check the box next to the Create An Anonymous Account option. Creating an anonymous account can decrease server security and cause login and permission difficulties. For this reason, it is not recommended.

### 5.5.12 MySQL Server Instance Config Wizard: The Confirmation Dialog

The final dialog in the MySQL Server Instance Config Wizard is the Confirmation Dialog. To start the configuration process, click the Execute button. To return to a previous dialog, click the Back button. To exit the MySQL Server Instance Config Wizard without configuring the server, click the Cancel button.
After you click the Execute button, the MySQL Server Instance Config Wizard performs a series of tasks and displays the progress onscreen as the tasks are performed.

The MySQL Server Instance Config Wizard first determines configuration file options based on your choices using a template prepared by MySQL developers and engineers. This template is named `my-template.ini` and is located in your server installation directory.

The MySQL Config Wizard then writes these options to the corresponding configuration file.

If you chose to create a service for the MySQL server, the MySQL Server Instance Config Wizard creates and starts the service. If you are reconfiguring an existing service, the MySQL Server Instance Config Wizard restarts the service to apply your configuration changes.

If you chose to set a root password, the MySQL Config Wizard connects to the server, sets your new root password, and applies any other security settings you may have selected.

After the MySQL Server Instance Config Wizard has completed its tasks, it displays a summary. Click the Finish button to exit the MySQL Server Config Wizard.

5.5.13 MySQL Server Instance Config Wizard: Creating an Instance from the Command Line

In addition to using the GUI interface to the MySQL Server Instance Config Wizard, you can also create instances automatically from the command line.
MySQL Server Instance Config Wizard: Creating an Instance from the Command Line

To use the MySQL Server Instance Config Wizard on the command line, you need to use the MySQLInstanceConfig.exe command that is installed with MySQL in the bin directory within the installation directory. MySQLInstanceConfig.exe takes a number of command-line arguments that set the properties that would normally be selected through the GUI interface, and then creates a new configuration file (my.ini) by combining these selections with a template configuration file to produce the working configuration file.

The main command line options are provided in the table below. Some of the options are required, while some options are optional.

### Table 5.4 MySQL Server Instance Config Wizard Command Line Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Parameters</td>
<td></td>
</tr>
<tr>
<td>-nPRODUCTNAME</td>
<td>The name of the instance when installed</td>
</tr>
<tr>
<td>-pPATH</td>
<td>Path of the base directory for installation. This is equivalent to the directory when using the basedir configuration parameter</td>
</tr>
<tr>
<td>-vVERSION</td>
<td>The version tag to use for this installation</td>
</tr>
<tr>
<td>Action to Perform</td>
<td></td>
</tr>
<tr>
<td>-i</td>
<td>Install an instance</td>
</tr>
<tr>
<td>-r</td>
<td>Remove an instance</td>
</tr>
<tr>
<td>-s</td>
<td>Stop an existing instance</td>
</tr>
<tr>
<td>-q</td>
<td>Perform the operation quietly</td>
</tr>
<tr>
<td>-lFILENAME</td>
<td>Save the installation progress in a logfile</td>
</tr>
<tr>
<td>Config File to Use</td>
<td></td>
</tr>
<tr>
<td>-tFILENAME</td>
<td>Path to the template config file that will be used to generate the installed configuration file</td>
</tr>
<tr>
<td>-cFILENAME</td>
<td>Path to a config file to be generated</td>
</tr>
</tbody>
</table>

The -t and -c options work together to set the configuration parameters for a new instance. The -t option specifies the template configuration file to use as the basic configuration, which are then merged with the configuration parameters generated by the MySQL Server Instance Config Wizard into the configuration file specified by the -c option.

A sample template file, my-template.ini is provided in the toplevel MySQL installation directory. The file contains elements that are replaced automatically by the MySQL Server Instance Config Wizard during configuration.

If you specify a configuration file that already exists, the existing configuration file will be saved in the file with the original, with the date and time added. For example, the mysql.ini will be copied to mysql 2009-10-27 1646.ini.bak.

The parameters that you can specify on the command line are listed in the table below.

### Table 5.5 MySQL Server Instance Config Wizard Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceName=$</td>
<td>Specify the name of the service to be created</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AddBinToPath={yes</td>
<td>no}</td>
</tr>
<tr>
<td>ServerType={DEVELOPMENT</td>
<td>SERVER</td>
</tr>
<tr>
<td>DatabaseType={MIXED</td>
<td>INNODB</td>
</tr>
<tr>
<td>ConnectionUsage={DSS</td>
<td>OLTP}</td>
</tr>
<tr>
<td>ConnectionCount=#</td>
<td>Specify the number of concurrent connections to support. For more information, see Section 5.5.4, &quot;MySQL Server Instance Config Wizard: The Server Type Dialog&quot;</td>
</tr>
<tr>
<td>SkipNetworking={yes</td>
<td>no}</td>
</tr>
<tr>
<td>Port=#</td>
<td>Specify the network port number to use for network connections. For more information, see Section 5.5.8, &quot;MySQL Server Instance Config Wizard: The Networking and Strict Mode Options Dialog&quot;</td>
</tr>
<tr>
<td>StrictMode={yes</td>
<td>no}</td>
</tr>
<tr>
<td>Charset=$</td>
<td>Specify the default character set. For more information, see Section 5.5.9, &quot;MySQL Server Instance Config Wizard: The Character Set Dialog&quot;</td>
</tr>
<tr>
<td>RootPassword=$</td>
<td>Specify the root password</td>
</tr>
<tr>
<td>RootCurrentPassword=$</td>
<td>Specify the current root password then stopping or reconfiguring an existing service</td>
</tr>
</tbody>
</table>

**Note**

When specifying options on the command line, you can enclose the entire command-line option and the value you are specifying using double quotation marks. This enables you to use spaces in the options. For example, "--C:\mysql.ini".

The following command installs a MySQL Server 5.1 instance from the directory C:\Program Files \MySQL\MySQL Server 5.1 using the service name MySQL51 and setting the root password to 1234.

```shell
MySQLInstanceConfig.exe -i -q "-lC:\mysql_install_log.txt" » 
"--nMySQL Server 5.1" "-pC:\Program Files\MySQL\MySQL Server 5.1" -v5.1.73 » 
"--tmy-template.ini" "-C:\mytest.ini" ServerType=DEVELOPMENT DatabaseType=MIXED » 
ConnectionUsage=DSS Port=3311 ServiceName=MySQL51 RootPassword=1234
```

In the above example, a log file will be generated in mysql_install_log.txt containing the information about the instance creation process. The log file generated by the above example is shown below:

```
Welcome to the MySQL Server Instance Configuration Wizard 1.0.16.0
Date: 2009-10-27 17:07:21
```
Installing service ...

Product Name: MySQL Server 5.1
Version: 5.1.73
Installation Path: C:\Program Files\MySQL\MySQL Server 5.1\

Creating configuration file C:\mytest.ini using template my-template.ini.
Options:
DEVELOPMENT
MIXED
DSS
STRICTMODE

Variables:
port: 3311
default-character-set: latin1
basedir: "C:/Program Files/MySQL/MySQL Server 5.1/"
datadir: "C:/Program Files/MySQL/MySQL Server 5.1/Data/"

Creating Windows service entry.
Service name: "MySQL51"
Parameters: "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld" --defaults-file="C:\mytest.ini" MySQL51.
Windows service MySQL51 installed.

When using the command line, the return values in the following table indicate an error performing the specified option.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Configuration template file cannot be found</td>
</tr>
<tr>
<td>3</td>
<td>The Windows service entry cannot be created</td>
</tr>
<tr>
<td>4</td>
<td>Could not connect to the Service Control Manager</td>
</tr>
<tr>
<td>5</td>
<td>The MySQL service cannot be started</td>
</tr>
<tr>
<td>6</td>
<td>The MySQL service cannot be stopped</td>
</tr>
<tr>
<td>7</td>
<td>The security settings cannot be applied</td>
</tr>
<tr>
<td>8</td>
<td>The configuration file cannot be written</td>
</tr>
<tr>
<td>9</td>
<td>The Windows service entry cannot be removed</td>
</tr>
</tbody>
</table>

You can perform an installation of MySQL automatically using the MSI package. For more information, see Section 5.4.2, “Automating MySQL Installation on Microsoft Windows Using the MSI Package”.

5.6 Installing MySQL on Microsoft Windows Using a noinstall Zip Archive

Users who are installing from the noinstall package can use the instructions in this section to manually install MySQL. The process for installing MySQL from a Zip archive is as follows:

1. Extract the archive to the desired install directory
2. Create an option file
3. Choose a MySQL server type
4. Start the MySQL server
5. Secure the default user accounts

This process is described in the sections that follow.

5.6.1 Extracting the Install Archive

To install MySQL manually, do the following:

1. If you are upgrading from a previous version please refer to Section 5.9, “Upgrading MySQL Server on Microsoft Windows”, before beginning the upgrade process.

2. Make sure that you are logged in as a user with administrator privileges.

3. Choose an installation location. Traditionally, the MySQL server is installed in \texttt{C:\mysql}. The MySQL Installation Wizard installs MySQL under \texttt{C:\Program Files\MySQL}. If you do not install MySQL at \texttt{C:\mysql}, you must specify the path to the install directory during startup or in an option file. See Section 5.6.2, “Creating an Option File”.

4. Extract the install archive to the chosen installation location using your preferred Zip archive tool. Some tools may extract the archive to a folder within your chosen installation location. If this occurs, you can move the contents of the subfolder into the chosen installation location.

5.6.2 Creating an Option File

If you need to specify startup options when you run the server, you can indicate them on the command line or place them in an option file. For options that are used every time the server starts, you may find it most convenient to use an option file to specify your MySQL configuration. This is particularly true under the following circumstances:

- The installation or data directory locations are different from the default locations (\texttt{C:\Program Files\MySQL\MySQL Server 5.1} and \texttt{C:\Program Files\MySQL\MySQL Server 5.1\data}).

- You need to tune the server settings, such as memory, cache, or InnoDB configuration information.

When the MySQL server starts on Windows, it looks for option files in several locations, such as the Windows directory, \texttt{C:\}, and the MySQL installation directory (for the full list of locations, see Using Option Files). The Windows directory typically is named something like \texttt{C:\WINDOWS}. You can determine its exact location from the value of the \texttt{WINDIR} environment variable using the following command:

```
C:\> echo %WINDIR%
```

MySQL looks for options in each location first in the \texttt{my.ini} file, and then in the \texttt{my.cnf} file. However, to avoid confusion, it is best if you use only one file. If your PC uses a boot loader where \texttt{C:} is not the boot drive, your only option is to use the \texttt{my.ini} file. Whichever option file you use, it must be a plain text file.

You can also make use of the example option files included with your MySQL distribution; see Server Configuration Defaults.

An option file can be created and modified with any text editor, such as Notepad. For example, if MySQL is installed in \texttt{E:\mysql} and the data directory is in \texttt{E:\mydata\data}, you can create an option file containing a \texttt{[mysqld]} section to specify values for the \texttt{basedir} and \texttt{datadir} options:

```
[mysqld]
# set basedir to your installation path
```
Selecting a MySQL Server Type

```plaintext
basedir=E:/mysql
# set datadir to the location of your data directory
datadir=E:/mydata/data
```

Microsoft Windows path names are specified in option files using (forward) slashes rather than backslashes. If you do use backslashes, double them:

```plaintext
[mysqld]
# set basedir to your installation path
basedir=E:\mysql
# set datadir to the location of your data directory
datadir=E:\mydata\data
```

The rules for use of backslash in option file values are given in Using Option Files.

In MySQL 5.1.23 and earlier, the MySQL installer places the data directory directly under the directory where you install MySQL. On MySQL 5.1.24 and later, the data directory is located within the AppData directory for the user running MySQL.

If you would like to use a data directory in a different location, you should copy the entire contents of the data directory to the new location. For example, if you want to use `E:\mydata` as the data directory instead, you must do two things:

1. Move the entire `data` directory and all of its contents from the default location (for example `C:\Program Files\MySQL\MySQL Server 5.1\data`) to `E:\mydata`.
2. Use a `--datadir` option to specify the new data directory location each time you start the server.

### 5.6.3 Selecting a MySQL Server Type

The following table shows the available servers for Windows in MySQL 5.1.20 and earlier.

<table>
<thead>
<tr>
<th>Binary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqld-nt</td>
<td>Optimized binary with named-pipe support</td>
</tr>
<tr>
<td>mysqld</td>
<td>Optimized binary without named-pipe support</td>
</tr>
<tr>
<td>mysqld-debug</td>
<td>Like <code>mysqld-nt</code>, but compiled with full debugging and automatic memory allocation checking</td>
</tr>
</tbody>
</table>

The following table shows the available servers for Windows in MySQL 5.1.21 and later.

**Table 5.7 mysqld binary types for Microsoft Windows MySQL 5.1.21 and later**

<table>
<thead>
<tr>
<th>Binary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqld</td>
<td>Optimized binary with named-pipe support</td>
</tr>
<tr>
<td>mysqld-debug</td>
<td>Like <code>mysqld</code>, but compiled with full debugging and automatic memory allocation checking</td>
</tr>
</tbody>
</table>

All of the preceding binaries are optimized for modern Intel processors, but should work on any Intel i386-class or higher processor.

Each of the servers in a distribution support the same set of storage engines. The `SHOW ENGINES` statement displays which engines a given server supports.
All Windows MySQL 5.1 servers have support for symbolic linking of database directories.

MySQL supports TCP/IP on all Windows platforms. MySQL servers on Windows support named pipes as indicated in the following list. However, the default is to use TCP/IP regardless of platform. (Named pipes are slower than TCP/IP in many Windows configurations.)

Use of named pipes is subject to these conditions:

- Named pipes are enabled only if you start the server with the `--enable-named-pipe` option. It is necessary to use this option explicitly because some users have experienced problems with shutting down the MySQL server when named pipes were used.

- For MySQL 5.1.20 and earlier, named-pipe connections are permitted only by the `mysqld-nt` and `mysqld-debug` servers. For MySQL 5.1.21 and later, the `mysqld` and `mysqld-debug` servers both contain support for named-pipe connections.

**Note**

Most of the examples in this manual use `mysqld` as the server name. If you choose to use a different server, such as `mysqld-nt` or `mysqld-debug`, make the appropriate substitutions in the commands that are shown in the examples.

### 5.6.4 Starting MySQL Server on Microsoft Windows for the First Time

This section gives a general overview of starting the MySQL server. The following sections provide more specific information for starting the MySQL server from the command line or as a Windows service.

The information here applies primarily if you installed MySQL using the `Noinstall` version, or if you wish to configure and test MySQL manually rather than with the GUI tools.

**Note**

The MySQL Notifier GUI can be used to start/stop/restart the MySQL server at any time.

The examples in these sections assume that MySQL is installed under the default location of `C:\Program Files\MySQL\MySQL Server 5.1`. Adjust the path names shown in the examples if you have MySQL installed in a different location.

Clients have two options. They can use TCP/IP, or they can use a named pipe if the server supports named-pipe connections.

MySQL for Windows also supports shared-memory connections if the server is started with the `--shared-memory` option. Clients can connect through shared memory by using the `--protocol=MEMORY` option.

For information about which server binary to run, see Section 5.6.3, “Selecting a MySQL Server Type”.

Testing is best done from a command prompt in a console window (or “DOS window”). In this way you can have the server display status messages in the window where they are easy to see. If something is wrong with your configuration, these messages make it easier for you to identify and fix any problems.

To start the server, enter this command:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld" --console
```
For a server that includes InnoDB support, you should see the messages similar to those following as it starts (the path names and sizes may differ):

```
InnoDB: The first specified datafile c:\ibdata\ibdata1 did not exist:
InnoDB: a new database to be created!
InnoDB: Setting file c:\ibdata\ibdata1 size to 209715200
InnoDB: Database physically writes the file full: wait...
InnoDB: Log file c:\iblogs\ib_logfile0 did not exist: new to be created
InnoDB: Setting log file c:\iblogs\ib_logfile0 size to 31457280
InnoDB: Log file c:\iblogs\ib_logfile1 did not exist: new to be created
InnoDB: Setting log file c:\iblogs\ib_logfile1 size to 31457280
InnoDB: Log file c:\iblogs\ib_logfile2 did not exist: new to be created
InnoDB: Setting log file c:\iblogs\ib_logfile2 size to 31457280
InnoDB: Doublewrite buffer not found: creating new
InnoDB: Doublewrite buffer created
InnoDB: creating foreign key constraint system tables
InnoDB: foreign key constraint system tables created
011024 10:58:25  InnoDB: Started
```

When the server finishes its startup sequence, you should see something like this, which indicates that the server is ready to service client connections:

```
mysqld: ready for connections
Version: '5.1.73' socket: '' port: 3306
```

The server continues to write to the console any further diagnostic output it produces. You can open a new console window in which to run client programs.

If you omit the `--console` option, the server writes diagnostic output to the error log in the data directory (C:\Program Files\MySQL\MySQL Server 5.1\data by default). The error log is the file with the .err extension, and may be set using the `--log-error` option.

---

**Note**

The accounts that are listed in the MySQL grant tables initially have no passwords. After starting the server, you should set up passwords for them using the instructions in Section 10.4, "Securing the Initial MySQL Accounts".

---

### 5.6.5 Starting MySQL Server from the Windows Command Line

The MySQL server can be started manually from the command line. This can be done on any version of Windows.

**Note**

The MySQL Notifier GUI can also be used to start/stop/restart the MySQL server.

To start the `mysqld` server from the command line, you should start a console window (or “DOS window”) and enter this command:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld"
```

The path to `mysqld` may vary depending on the install location of MySQL on your system.

You can stop the MySQL server by executing this command:
Customizing the PATH for MySQL Tools

C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqladmin" -u root shutdown

Note
If the MySQL root user account has a password, you need to invoke mysqladmin with the -p option and supply the password when prompted.

This command invokes the MySQL administrative utility mysqladmin to connect to the server and tell it to shut down. The command connects as the MySQL root user, which is the default administrative account in the MySQL grant system.

Note
Users in the MySQL grant system are wholly independent from any login users under Microsoft Windows.

If mysqld doesn't start, check the error log to see whether the server wrote any messages there to indicate the cause of the problem. By default, the error log is located in the C:\Program Files\MySQL\MySQL Server 5.1\data directory. It is the file with a suffix of .err, or may be specified by passing in the --log-error option. Alternatively, you can try to start the server with the --console option; in this case, the server may display some useful information on the screen that will help solve the problem.

The last option is to start mysqld with the --standalone and --debug options. In this case, mysqld writes a log file C:\mysqld.trace that should contain the reason why mysqld doesn't start. See The DBUG Package.

Use mysqld --verbose --help to display all the options that mysqld supports.

5.6.6 Customizing the PATH for MySQL Tools

To make it easier to invoke MySQL programs, you can add the path name of the MySQL bin directory to your Windows system PATH environment variable:

• On the Windows desktop, right-click the My Computer icon, and select Properties.

• Next select the Advanced tab from the System Properties menu that appears, and click the Environment Variables button.

• Under System Variables, select Path, and then click the Edit button. The Edit System Variable dialogue should appear.

• Place your cursor at the end of the text appearing in the space marked Variable Value. (Use the End key to ensure that your cursor is positioned at the very end of the text in this space.) Then enter the complete path name of your MySQL bin directory (for example, C:\Program Files\MySQL\MySQL Server 5.1\bin)

Note
There must be a semicolon separating this path from any values present in this field.

Dismiss this dialogue, and each dialogue in turn, by clicking OK until all of the dialogues that were opened have been dismissed. You should now be able to invoke any MySQL executable program by typing its name at the DOS prompt from any directory on the system, without having to supply the path. This includes the servers, the mysql client, and all MySQL command-line utilities such as mysqladmin and mysqldump.
You should not add the MySQL bin directory to your Windows PATH if you are running multiple MySQL servers on the same machine.

**Warning**

You must exercise great care when editing your system PATH by hand; accidental deletion or modification of any portion of the existing PATH value can leave you with a malfunctioning or even unusable system.

### 5.6.7 Starting MySQL Server as a Microsoft Windows Service

On Windows, the recommended way to run MySQL is to install it as a Windows service, so that MySQL starts and stops automatically when Windows starts and stops, and can be managed using the service manager framework. A MySQL server installed as a service can also be controlled from the command line using NET commands, or with the graphical Services utility. Generally, to install MySQL as a Windows service you should be logged in using an account that has administrator rights.

**Note**

The MySQL Notifier GUI can also be used to monitor the status of the MySQL service.

The Services utility (the Windows Service Control Manager) can be found in the Windows Control Panel (under Administrative Tools on Windows 2000, XP, Vista, and Server 2003). To avoid conflicts, it is advisable to close the Services utility while performing server installation or removal operations from the command line.

### Installing the service

Before installing MySQL as a Windows service, you should first stop the current server if it is running by using the following command:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqladmin" -u root shutdown
```

**Note**

If the MySQL root user account has a password, you need to invoke mysqladmin with the -p option and supply the password when prompted.

This command invokes the MySQL administrative utility mysqladmin to connect to the server and tell it to shut down. The command connects as the MySQL root user, which is the default administrative account in the MySQL grant system.

**Note**

Users in the MySQL grant system are wholly independent from any login users under Windows.

Install the server as a service using this command:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld" --install
```

The service-installation command does not start the server. Instructions for that are given later in this section.
Starting MySQL Server as a Microsoft Windows Service

To make it easier to invoke MySQL programs, you can add the path name of the MySQL bin directory to your Windows system PATH environment variable:

• On the Windows desktop, right-click the My Computer icon, and select Properties.

• Next select the Advanced tab from the System Properties menu that appears, and click the Environment Variables button.

• Under System Variables, select Path, and then click the Edit button. The Edit System Variable dialogue should appear.

• Place your cursor at the end of the text appearing in the space marked Variable Value. (Use the End key to ensure that your cursor is positioned at the very end of the text in this space.) Then enter the complete path name of your MySQL bin directory (for example, C:\Program Files\MySQL\MySQL Server 5.1\bin), and there should be a semicolon separating this path from any values present in this field. Dismiss this dialogue, and each dialogue in turn, by clicking OK until all of the dialogues that were opened have been dismissed. You should now be able to invoke any MySQL executable program by typing its name at the DOS prompt from any directory on the system, without having to supply the path. This includes the servers, the mysql client, and all MySQL command-line utilities such as mysqladmin and mysqldump.

You should not add the MySQL bin directory to your Windows PATH if you are running multiple MySQL servers on the same machine.

Warning

You must exercise great care when editing your system PATH by hand; accidental deletion or modification of any portion of the existing PATH value can leave you with a malfunctioning or even unusable system.

The following additional arguments can be used when installing the service:

• You can specify a service name immediately following the --install option. The default service name is MySQL.

• If a service name is given, it can be followed by a single option. By convention, this should be --defaults-file=file_name to specify the name of an option file from which the server should read options when it starts.

The use of a single option other than --defaults-file is possible but discouraged. --defaults-file is more flexible because it enables you to specify multiple startup options for the server by placing them in the named option file.

• You can also specify a --local-service option following the service name. This causes the server to run using the LocalService Windows account that has limited system privileges. This account is available only for Windows XP or newer. If both --defaults-file and --local-service are given following the service name, they can be in any order.

For a MySQL server that is installed as a Windows service, the following rules determine the service name and option files that the server uses:

• If the service-installation command specifies no service name or the default service name (MySQL) following the --install option, the server uses the a service name of MySQL and reads options from the [mysqld] group in the standard option files.

• If the service-installation command specifies a service name other than MySQL following the --install option, the server uses that service name. It reads options from the [mysqld] group and the group that
Starting MySQL Server as a Microsoft Windows Service

has the same name as the service in the standard option files. This enables you to use the [mysqld] group for options that should be used by all MySQL services, and an option group with the service name for use by the server installed with that service name.

- If the service-installation command specifies a --defaults-file option after the service name, the server reads options the same way as described in the previous item, except that it reads options only from the named file and ignores the standard option files.

As a more complex example, consider the following command:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld"
   --install MySQL --defaults-file=C:\my-opts.cnf
```

Here, the default service name (MySQL) is given after the --install option. If no --defaults-file option had been given, this command would have the effect of causing the server to read the [mysqld] group from the standard option files. However, because the --defaults-file option is present, the server reads options from the [mysqld] option group, and only from the named file.

**Note**

On Windows, if the server is started with the --defaults-file and --install options, --install must be first. Otherwise, mysqld.exe will attempt to start the MySQL server.

You can also specify options as Start parameters in the Windows Services utility before you start the MySQL service.

**Starting the service**

Once a MySQL server has been installed as a service, Windows starts the service automatically whenever Windows starts. The service also can be started immediately from the Services utility, or by using a NET START MySQL command. The NET command is not case sensitive.

When run as a service, mysqld has no access to a console window, so no messages can be seen there. If mysqld does not start, check the error log to see whether the server wrote any messages there to indicate the cause of the problem. The error log is located in the MySQL data directory (for example, C:\Program Files\MySQL\MySQL Server 5.1\data). It is the file with a suffix of .err.

When a MySQL server has been installed as a service, and the service is running, Windows stops the service automatically when Windows shuts down. The server also can be stopped manually by using the Services utility, the NET STOP MySQL command, or the mysqladmin shutdown command.

You also have the choice of installing the server as a manual service if you do not wish for the service to be started automatically during the boot process. To do this, use the --install-manual option rather than the --install option:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld" --install-manual
```

**Removing the service**

To remove a server that is installed as a service, first stop it if it is running by executing NET STOP MySQL. Then use the --remove option to remove it:
5.6.8 Testing The MySQL Server Installation on Microsoft Windows

You can test whether the MySQL server is working by executing any of the following commands:

- `C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqlshow"
- `C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqlshow" -u root mysql
- `C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqladmin" version status proc
- `C:\> "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql" test

**Note**

By default, `mysqlshow` will try to connect using the ODBC user. This user is not created by default. You should specify a valid user, or `root` with the right password to check the operation of the server.

If `mysqld` is slow to respond to TCP/IP connections from client programs, there is probably a problem with your DNS. In this case, start `mysqld` with the `--skip-name-resolve` option and use only `localhost` and IP addresses in the `Host` column of the MySQL grant tables. (Be sure that an account exists that specifies an IP address or you may not be able to connect.)

You can force a MySQL client to use a named-pipe connection rather than TCP/IP by specifying the `--pipe` or `--protocol=PIPE` option, or by specifying . (period) as the host name. Use the `--socket` option to specify the name of the pipe if you do not want to use the default pipe name.

If you have set a password for the `root` account, deleted the anonymous account, or created a new user account, then to connect to the MySQL server you must use the appropriate `-u` and `-p` options with the commands shown previously. See Connecting to the MySQL Server.

For more information about `mysqlshow`, see `mysqlshow — Display Database, Table, and Column Information`.

5.7 Troubleshooting a Microsoft Windows MySQL Server Installation

When installing and running MySQL for the first time, you may encounter certain errors that prevent the MySQL server from starting. This section helps you diagnose and correct some of these errors.

Your first resource when troubleshooting server issues is the **error log**. The MySQL server uses the error log to record information relevant to the error that prevents the server from starting. The error log is located in the **data directory** specified in your `my.ini` file. The default data directory location is `C:\Program Files\MySQL\MySQL Server 5.1\data`, or `C:\ProgramData\MySql` on Windows 7 and Windows Server 2008. The `C:\ProgramData` directory is hidden by default. You need to change your folder options to see the directory and contents. For more information on the error log and understanding the content, see **The Error Log**.

For information regarding possible errors, also consult the console messages displayed when the MySQL service is starting. Use the `NET START MySQL` command from the command line after installing `mysqld`
as a service to see any error messages regarding the starting of the MySQL server as a service. See Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”.

The following examples show other common error messages you might encounter when installing MySQL and starting the server for the first time:

- If the MySQL server cannot find the mysql privileges database or other critical files, it displays these messages:

  ```
  System error 1067 has occurred.
  Fatal error: Can’t open and lock privilege tables:
  Table 'mysql.user' doesn't exist
  ```

  These messages often occur when the MySQL base or data directories are installed in different locations than the default locations (C:\Program Files\MySQL\MySQL Server 5.1 and C:\Program Files\MySQL\MySQL Server 5.1\data, respectively).

  This situation can occur when MySQL is upgraded and installed to a new location, but the configuration file is not updated to reflect the new location. In addition, old and new configuration files might conflict. Be sure to delete or rename any old configuration files when upgrading MySQL.

  If you have installed MySQL to a directory other than C:\Program Files\MySQL\MySQL Server 5.1, ensure that the MySQL server is aware of this through the use of a configuration (my.ini) file. Put the my.ini file in your Windows directory, typically C:\WINDOWS. To determine its exact location from the value of the WINDIR environment variable, issue the following command from the command prompt:

  ```
  C:\> echo %WINDIR%
  ```

  You can create or modify an option file with any text editor, such as Notepad. For example, if MySQL is installed in E:\mysql and the data directory is D:\MySQLdata, you can create the option file and set up a [mysqld] section to specify values for the basedir and datadir options:

  ```
  [mysqld]
  # set basedir to your installation path
  basedir=E:/mysql
  # set datadir to the location of your data directory
  datadir=D:/MySQLdata
  ```

  Microsoft Windows path names are specified in option files using (forward) slashes rather than backslashes. If you do use backslashes, double them:

  ```
  [mysqld]
  # set basedir to your installation path
  basedir=C:\Program Files\MySQL\MySQL Server 5.1
  # set datadir to the location of your data directory
  datadir=D:\MySQLdata
  ```

  The rules for use of backslash in option file values are given in Using Option Files.

  If you change the datadir value in your MySQL configuration file, you must move the contents of the existing MySQL data directory before restarting the MySQL server.

  See Section 5.6.2, “Creating an Option File”.

- If you reinstall or upgrade MySQL without first stopping and removing the existing MySQL service and install MySQL using the MySQL Config Wizard, you may see this error:
Error: Cannot create Windows service for MySql. Error: 0

This occurs when the Configuration Wizard tries to install the service and finds an existing service with the same name.

One solution to this problem is to choose a service name other than mysql when using the configuration wizard. This enables the new service to be installed correctly, but leaves the outdated service in place. Although this is harmless, it is best to remove old services that are no longer in use.

To permanently remove the old mysql service, execute the following command as a user with administrative privileges, on the command line:

```
C:\> sc delete mysql
[SC] DeleteService SUCCESS
```

If the sc utility is not available for your version of Windows, download the delsrv utility from http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/delsrv-o.asp and use the delsrv mysql syntax.

5.8 Windows Postinstallation Procedures

GUI tools exist that perform most of the tasks described in this section, including:

- MySQL Installer: Used to install and upgrade MySQL products.
- MySQL Workbench: Manages the MySQL server and edits SQL queries.
- MySQL Notifier: Starts, stops, or restarts the MySQL server, and monitors its status.
- MySQL for Excel: Edits MySQL data with Microsoft Excel.

On Windows, you need not create the data directory and the grant tables. MySQL Windows distributions include the grant tables with a set of preinitialized accounts in the mysql database under the data directory.

Regarding passwords, if you installed MySQL using the Windows Installation Wizard, you may have already assigned passwords to the accounts. (See Section 5.4.1, “Using the MySQL Installation Wizard for Microsoft Windows”.) Otherwise, use the password-assignment procedure given in Section 10.4, “Securing the Initial MySQL Accounts”.

Before assigning passwords, you might want to try running some client programs to make sure that you can connect to the server and that it is operating properly. Make sure that the server is running (see Section 5.6.4, “Starting MySQL Server on Microsoft Windows for the First Time”). You can also set up a MySQL service that runs automatically when Windows starts (see Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”).

These instructions assume that your current location is the MySQL installation directory and that it has a bin subdirectory containing the MySQL programs used here. If that is not true, adjust the command path names accordingly.

If you installed MySQL using the Windows installation Wizard (see Section 5.4.1, “Using the MySQL Installation Wizard for Microsoft Windows”), the default installation directory is C:\Program Files \MySQL\MySQL Server 5.1:
Windows Postinstallation Procedures

C:\> cd "C:\Program Files\MySQL\MySQL Server 5.1"

A common installation location for installation from a Zip package is C:\mysql:

C:\> cd C:\mysql

Alternatively, add the bin directory to your PATH environment variable setting. That enables your command interpreter to find MySQL programs properly, so that you can run a program by typing only its name, not its path name. See Section 5.6.6, "Customizing the PATH for MySQL Tools".

With the server running, issue the following commands to verify that you can retrieve information from the server. The output should be similar to that shown here.

Use mysqlshow to see what databases exist:

C:\> bin\mysqlshow
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+

The list of installed databases may vary, but will always include the minimum of mysql and information_schema.

The preceding command (and commands for other MySQL programs such as mysql) may not work if the correct MySQL account does not exist. For example, the program may fail with an error, or you may not be able to view all databases. If you installed using the MSI packages and used the MySQL Server Instance Config Wizard, then the root user will have been created automatically with the password you supplied. In this case, you should use the -u root and -p options. (You will also need to use the -u root and -p options if you have already secured the initial MySQL accounts.) With -p, you will be prompted for the root password. For example:

C:\> bin\mysqlshow -u root -p
Enter password: (enter root password here)
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+

If you specify a database name, mysqlshow displays a list of the tables within the database:
Use the `mysql` program to select information from a table in the `mysql` database:

```bash
C:\> bin\mysql -e "SELECT User, Host FROM mysql.user" mysql
+------+-----------+
| User | Host      |
+------+-----------+
| root | localhost |
+------+-----------+
```

For more information about `mysql` and `mysqlshow`, see `mysql — The MySQL Command-Line Tool`, and `mysqlshow — Display Database, Table, and Column Information`.

If you are running a version of Windows that supports services, you can set up the MySQL server to run automatically when Windows starts. See Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”.

## 5.9 Upgrading MySQL Server on Microsoft Windows

To upgrade MySQL on Windows, follow these steps:

1. Review Section 11.1, “Upgrading MySQL”, for additional information on upgrading MySQL that is not specific to Windows.

2. Always back up your current MySQL installation before performing an upgrade. See Database Backup Methods.


4. Before upgrading MySQL, stop the server. If the server is installed as a service, stop the service with the following command from the command prompt:

```bash
C:\> NET STOP MySQL
```

If you are not running the MySQL server as a service, use `mysqladmin` to stop it. For example, before upgrading from MySQL 5.0 to 5.1, use `mysqladmin` from MySQL 5.0 as follows:

```bash
C:\> "C:\Program Files\MySQL\MySQL Server 5.0\bin\mysqladmin" -u root shutdown
```

### Note

If the MySQL `root` user account has a password, invoke `mysqladmin` with the `-p` option and enter the password when prompted.
5. Before upgrading to MySQL 5.1 from a version previous to 4.1.5, or from a version of MySQL installed from a Zip archive to a version of MySQL installed with the MySQL Installation Wizard, you must first manually remove the previous installation and MySQL service (if the server is installed as a service).

To remove the MySQL service, use the following command:

```
C:\> C:\mysql\bin\mysqld --remove
```

**Important**

If you do not remove the existing service, the MySQL Installation Wizard may fail to properly install the new MySQL service.

6. When upgrading from MySQL 5.1.23 to MySQL 5.1.24, the change in the default location of the data directory from a directory within the MySQL installation to the `AppData` folder means that you must manually copy the data files from your old installation to the new location.

7. If you are using the MySQL Installation Wizard, start the wizard as described in Section 5.4.1, “Using the MySQL Installation Wizard for Microsoft Windows”.

8. If you are upgrading MySQL from a Zip archive, extract the archive. You may either overwrite your existing MySQL installation (usually located at `C:\mysql`), or install it into a different directory, such as `C:\mysql5`. Overwriting the existing installation is recommended. However, for upgrades (as opposed to installing for the first time), you must remove the data directory from your existing MySQL installation to avoid replacing your current data files. To do so, follow these steps:

   a. Unzip the Zip archive in some location other than your current MySQL installation
   b. Remove the data directory
   c. Rezip the Zip archive
   d. Unzip the modified Zip archive on top of your existing installation

   Alternatively:
   a. Unzip the Zip archive in some location other than your current MySQL installation
   b. Remove the data directory
   c. Move the data directory from the current MySQL installation to the location of the just-removed data directory
   d. Remove the current MySQL installation
   e. Move the unzipped installation to the location of the just-removed installation

9. If you were running MySQL as a Windows service and you had to remove the service earlier in this procedure, reinstall the service. (See Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”.)

10. Restart the server. For example, use `NET START MySQL` if you run MySQL as a service, or invoke `mysqld` directly otherwise.

11. As Administrator, run `mysql_upgrade` to check your tables, attempt to repair them if necessary, and update your grant tables if they have changed so that you can take advantage of any new capabilities. See `mysql_upgrade — Check and Upgrade MySQL Tables`.
12. If you encounter errors, see Section 5.7, “Troubleshooting a Microsoft Windows MySQL Server Installation”.
Chapter 6 Installing MySQL on OS X

Table of Contents

6.1 General Notes on Installing MySQL on OS X ................................................................. 107
6.2 Installing MySQL on OS X Using Native Packages ..................................................... 109
6.3 Installing a MySQL Launch Daemon ............................................................................. 112
6.4 Installing the MySQL Startup Item .............................................................................. 113
6.5 Installing and Using the MySQL Preference Pane ......................................................... 118
6.6 Using the Bundled MySQL on OS X Server ................................................................. 120

For a list of OS X versions that the MySQL server supports, see http://www.mysql.com/support/supportedplatforms/database.html.

MySQL for OS X is available in a number of different forms:

- Native Package Installer format, which uses the native OS X installer (DMG) to walk you through the installation of MySQL. For more information, see Section 6.2, “Installing MySQL on OS X Using Native Packages”. You can use the package installer with OS X 10.6 and later, and the package is available for both 32-bit and 64-bit architectures. The user you use to perform the installation must have administrator privileges.

- Tar package format, which uses a file packaged using the Unix tar and gzip commands. To use this method, you will need to open a Terminal window. You do not need administrator privileges using this method, as you can install the MySQL server anywhere using this method. For more information on using this method, you can use the generic instructions for using a tarball, Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries. You can use the package installer with OS X 10.6 and later, and available for both 32-bit and 64-bit architectures.

In addition to the core installation, the Package Installer also includes Section 6.4, “Installing the MySQL Startup Item” and Section 6.5, “Installing and Using the MySQL Preference Pane”, both of which simplify the management of your installation.

- OS X server includes a version of MySQL as standard. If you want to use a more recent version than that supplied with the OS X server release, you can make use of the package or tar formats. For more information on using the MySQL bundled with OS X, see Section 6.6, “Using the Bundled MySQL on OS X Server”.

For additional information on using MySQL on OS X, see Section 6.1, “General Notes on Installing MySQL on OS X”.

6.1 General Notes on Installing MySQL on OS X

You should keep the following issues and notes in mind:

- OS X 10.4 deprecated startup items in favor of launchd daemons, and as of OS X 10.10 (Yosemite), startup items do not function. For these reasons, using launchd daemons is preferred over startup items.

- The default location for the MySQL Unix socket is different on OS X and OS X Server depending on the installation type you chose. The following table shows the default locations by installation type.
Table 6.1 MySQL Unix Socket Locations on OS X by Installation Type

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Socket Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Installer from MySQL</td>
<td>/tmp/mysql.sock</td>
</tr>
<tr>
<td>Tarball from MySQL</td>
<td>/tmp/mysql.sock</td>
</tr>
<tr>
<td>MySQL Bundled with OS X Server</td>
<td>/var/mysql/mysql.sock</td>
</tr>
</tbody>
</table>

To prevent issues, you should either change the configuration of the socket used within your application (for example, changing `php.ini`), or you should configure the socket location using a MySQL configuration file and the `socket` option. For more information, see Server Command Options.

- You may need (or want) to create a specific `mysql` user to own the MySQL directory and data. You can do this through the Directory Utility, and the `mysql` user should already exist. For use in single user mode, an entry for `_mysql` (note the underscore prefix) should already exist within the system `/etc/passwd` file.

- If you get an “insecure startup item disabled” error when MySQL launches, use the following procedure. Adjust the pathnames appropriately for your system.

1. Modify the `mysql.script` using this command (enter it on a single line):

   ```shell
   sudo /Applications/TextEdit.app/Contents/MacOS/TextEdit
   /usr/local/mysql/support-files/mysql.server
   ```

2. Locate the option file that defines the `basedir` value and modify it to contain these lines:

   ```
   basedir=/usr/local/mysql
   datadir=/usr/local/mysql/data
   ```

   In the `/Library/StartupItems/MySQLCOM/` directory, make the following group ID changes from `staff` to `wheel`:

   ```shell
   sudo chgrp wheel MySQLCOM StartupParameters.plist
   ```

3. Start the server from System Preferences or Terminal.app.

- Because the MySQL package installer installs the MySQL contents into a version and platform specific directory, you can use this to upgrade and migrate your database between versions. You will need to either copy the `data` directory from the old version to the new version, or alternatively specify an alternative `datadir` value to set location of the data directory.

- You might want to add aliases to your shell’s resource file to make it easier to access commonly used programs such as `mysql` and `mysqladmin` from the command line. The syntax for `bash` is:

  ```
  alias mysql=/usr/local/mysql/bin/mysql
  alias mysqladmin=/usr/local/mysql/bin/mysqladmin
  ```

For `tcsh`, use:

  ```
  alias mysql /usr/local/mysql/bin/mysql
  alias mysqladmin /usr/local/mysql/bin/mysqladmin
  ```

108
Even better, add `/usr/local/mysql/bin` to your `PATH` environment variable. You can do this by modifying the appropriate startup file for your shell. For more information, see Invoking MySQL Programs.

- After you have copied over the MySQL database files from the previous installation and have successfully started the new server, you should consider removing the old installation files to save disk space. Additionally, you should also remove older versions of the Package Receipt directories located in `/Library/Receipts/mysql-VERSION.pkg`.

# 6.2 Installing MySQL on OS X Using Native Packages

The package is located inside a disk image (`.dmg`) file that you first need to mount by double-clicking its icon in the Finder. It should then mount the image and display its contents.

**Note**

Before proceeding with the installation, be sure to stop all running MySQL server instances by using either the MySQL Manager Application (on OS X Server) or `mysqladmin shutdown` on the command line.

When installing from the package version, you should also install the MySQL Preference Pane, which will enable you to control the startup and execution of your MySQL server from System Preferences. For more information, see Section 6.5, “Installing and Using the MySQL Preference Pane”.

When installing using the package installer, the files are installed into a directory within `/usr/local` matching the name of the installation version and platform. For example, the installer file `mysql-5.1-osx10.6-x86_64.dmg` installs MySQL into `/usr/local/mysql-5.1-osx10.6-x86_64/`. The following table shows the layout of the installation directory.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin, scripts</td>
<td><code>mysqld</code> server, client and utility programs</td>
</tr>
<tr>
<td>data</td>
<td>Log files, databases</td>
</tr>
<tr>
<td>docs</td>
<td>Helper documents, like the Release Notes and build information</td>
</tr>
<tr>
<td>include</td>
<td>Include (header) files</td>
</tr>
<tr>
<td>lib</td>
<td>Libraries</td>
</tr>
<tr>
<td>man</td>
<td>Unix manual pages</td>
</tr>
<tr>
<td>mysql-test</td>
<td>MySQL test suite</td>
</tr>
<tr>
<td>share</td>
<td>Miscellaneous support files, including error messages, sample configuration files, SQL for database installation</td>
</tr>
<tr>
<td>sql-bench</td>
<td>Benchmarks</td>
</tr>
<tr>
<td>support-files</td>
<td>Scripts and sample configuration files</td>
</tr>
<tr>
<td>/tmp/mysql.sock</td>
<td>Location of the MySQL Unix socket</td>
</tr>
</tbody>
</table>

During the package installer process, a symbolic link from `/usr/local/mysql` to the version/platform specific directory created during installation will be created automatically.

1. Download and open the MySQL package installer, which is provided on a disk image (`.dmg`) that includes the main MySQL installation package, the `MySQLStartupItem.pkg` installation package, and the `MySQL.prefPane`. Double-click the disk image to open it.
2. Double-click the MySQL installer package. It will be named according to the version of MySQL you have downloaded. For example, if you have downloaded MySQL server 5.1.73, double-click `mysql-5.1.73-osx-10.6-x86_64.pkg`.

3. You will be presented with the opening installer dialog. Click **Continue** to begin installation.
4. A copy of the installation instructions and other important information relevant to this installation are displayed. Click Continue.

5. If you have downloaded the community version of MySQL, you will be shown a copy of the relevant GNU General Public License. Click Continue.

6. Select the drive you want to use to install the MySQL Startup Item. The drive must have a valid, bootable, OS X operating system installed. Click Continue.
7. You will be asked to confirm the details of the installation, including the space required for the installation. To change the drive on which the MySQL server is installed, click either Go Back or Change Install Location.... To install the MySQL server, click Install.

8. Once the installation has been completed successfully, you will be shown an Install Succeeded message.

For convenience, you may also want to install the startup item and preference pane. See Section 6.4, “Installing the MySQL Startup Item”, and Section 6.5, “Installing and Using the MySQL Preference Pane”.

6.3 Installing a MySQL Launch Daemon

OS X uses launch daemons to automatically start, stop, and manage processes and applications such as MySQL. Using launch daemons is recommended over startup items on OS X.

Note

OS X 10.4 deprecated startup items in favor of launchd daemons, and as of OS X 10.10 (Yosemite), startup items do not function. For these reasons, using launchd daemons is preferred over startup items.

Here is an example launchd file that starts MySQL:
Installing the MySQL Startup Item

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>KeepAlive</key>
    <true/>
    <key>Label</key>
    <string>com.mysql.mysqld</string>
    <key>ProgramArguments</key>
    <array>
      <string>/usr/local/mysql/bin/mysqld_safe</string>
      <string>--user=mysql</string>
    </array>
  </dict>
</plist>
```

Note

Some users report that adding a plist DOCTYPE declaration causes the launchd operation to fail, despite it passing the lint check. For this reason, we have removed it from the example above.

Adjust the `ProgramArguments` array according to your system, as for example your path to `mysqld_safe` might be different. After making the proper adjustments, do the following:

- Save the XML as a file named `/Library/LaunchDaemons/com.mysql.mysql.plist`
- Adjust the file permissions using the Apple recommended owner "root", owning group "wheel", and file permissions "644"
  ```shell`
  sudo chown root:wheel /Library/LaunchDaemons/com.mysql.mysql.plist
  sudo chmod 644 /Library/LaunchDaemons/com.mysql.mysql.plist
  ```
- Enable this new MySQL service
  ```shell`
  sudo launchctl load -w /Library/LaunchDaemons/com.mysql.mysql.plist
  ```

The MySQL daemon is now running, and automatically starts when your system is rebooted.

6.4 Installing the MySQL Startup Item

The MySQL Installation Package includes a startup item that can be used to automatically start and stop MySQL.

Important

Startup items are deprecated in favor of launchd daemons. For additional information, see Section 6.3, “Installing a MySQL Launch Daemon”.

To install the MySQL Startup Item:

1. Download and open the MySQL package installer, which is provided on a disk image (.dmg) that includes the main MySQL installation package, the `MySQLStartupItem.pkg` installation package, and the `MySQL.prefPane`. Double-click the disk image to open it.
2. Double-click the MySQLStartItem.pkg file to start the installation process.

3. You will be presented with the Install MySQL Startup Item dialog.
Click **Continue** to continue the installation process.

4. A copy of the installation instructions and other important information relevant to this installation are displayed. Click **Continue**.

5. Select the drive you want to use to install the MySQL Startup Item. The drive must have a valid, bootable, OS X operating system installed. Click **Continue**.
6. You will be asked to confirm the details of the installation. To change the drive on which the startup item is installed, click either Go Back or Change Install Location... To install the startup item, click Install.

7. Once the installation has been completed successfully, you will be shown an Install Succeeded message.
The Startup Item for MySQL is installed into `/Library/StartupItems/MySQLCOM`. The Startup Item installation adds a variable `MYSQLCOM=-YES-` to the system configuration file `/etc/hostconfig`. If you want to disable the automatic startup of MySQL, change this variable to `MYSQLCOM=-NO-`.

Note

Deselecting **Automatically Start MySQL Server on Startup** from the MySQL Preference Pane sets the `MYSQLCOM` variable to `-NO-`.

After the installation, you can start and stop the MySQL server from the MySQL Preference Pane (preferred), or by running the following commands in a terminal window. You must have administrator privileges to perform these tasks, and you may be prompted for your password.

If you have installed the Startup Item, use this command to start the server:

```
shell> sudo /Library/StartupItems/MySQLCOM/MySQLCOM start
```

If you have installed the Startup Item, use this command to stop the server:

```
shell> sudo /Library/StartupItems/MySQLCOM/MySQLCOM stop
```
6.5 Installing and Using the MySQL Preference Pane

The MySQL Package installer disk image also includes a custom MySQL Preference Pane that enables you to start, stop, and control automated startup during boot of your MySQL installation.

To install the MySQL Preference Pane:

1. Download and open the MySQL package installer package, which is provided on a disk image (.dmg) that includes the main MySQL installation package, the MySQLStartupItem.pkg installation package, and the MySQL.prefPane. Double-click the disk image to open it.

![MySQL Package Installer: DMG Contents](image)

2. Double-click the MySQL.prefPane. The MySQL System Preferences will open.

3. If this is the first time you have installed the preference pane, you will be asked to confirm installation and whether you want to install the preference pane for all users, or only the current user. To install the preference pane for all users you will need administrator privileges. If necessary, you will be prompted for the username and password for a user with administrator privileges.

4. If you already have the MySQL Preference Pane installed, you will be asked to confirm whether you want to overwrite the existing MySQL Preference Pane.

   **Note**

   The MySQL Preference Pane only starts and stops MySQL installation installed from the MySQL package installation that have been installed in the default location.

Once the MySQL Preference Pane has been installed, you can control your MySQL server instance using the preference pane. To use the preference pane, open the System Preferences... from the Apple menu. Select the MySQL preference pane by clicking the MySQL logo within the Other section of the preference panes list.
The MySQL Preference Pane shows the current status of the MySQL server, showing **stopped** (in red) if the server is not running and **running** (in green) if the server has already been started. The preference pane also shows the current setting for whether the MySQL server has been set to start automatically.

- **To start the MySQL server using the preference pane:**
  
  Click Start MySQL Server. You may be prompted for the username and password of a user with administrator privileges to start the MySQL server.

- **To stop the MySQL server using the preference pane:**
  
  Click Stop MySQL Server. You may be prompted for the username and password of a user with administrator privileges to stop the MySQL server.

- **To automatically start the MySQL server when the system boots:**
Check the check box next to **Automatically Start MySQL Server on Startup**.

- **To disable automatic MySQL server startup when the system boots:**
  Uncheck the check box next to **Automatically Start MySQL Server on Startup**.

You can close the System Preferences... window once you have completed your settings.

### 6.6 Using the Bundled MySQL on OS X Server

If you are running OS X Server, a version of MySQL should already be installed. The following table shows the versions of MySQL that ship with OS X Server versions.

<table>
<thead>
<tr>
<th>OS X Server Version</th>
<th>MySQL Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2-10.2.2</td>
<td>3.23.51</td>
</tr>
<tr>
<td>10.2.3-10.2.6</td>
<td>3.23.53</td>
</tr>
<tr>
<td>10.3</td>
<td>4.0.14</td>
</tr>
<tr>
<td>10.3.2</td>
<td>4.0.16</td>
</tr>
<tr>
<td>10.4.0</td>
<td>4.1.10a</td>
</tr>
<tr>
<td>10.5.0</td>
<td>5.0.45</td>
</tr>
<tr>
<td>10.6.0</td>
<td>5.0.82</td>
</tr>
</tbody>
</table>

The following table shows the installation layout of MySQL on OS X Server.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/bin</td>
<td>Client programs</td>
</tr>
<tr>
<td>/var/mysql</td>
<td>Log files, databases</td>
</tr>
<tr>
<td>/usr/libexec</td>
<td>The mysqld server</td>
</tr>
<tr>
<td>/usr/share/man</td>
<td>Unix manual pages</td>
</tr>
<tr>
<td>/usr/share/mysql/mysql-test</td>
<td>MySQL test suite</td>
</tr>
<tr>
<td>/usr/share/mysql</td>
<td>Miscellaneous support files, including error messages, character set files, sample configuration files, SQL for database installation</td>
</tr>
<tr>
<td>/var/mysql/mysql.sock</td>
<td>Location of the MySQL Unix socket</td>
</tr>
</tbody>
</table>

### Additional Resources

- For more information on managing the bundled MySQL instance in OS X Server 10.5, see *Mac OS X Server: Web Technologies Administration For Version 10.5 Leopard*.

- For more information on managing the bundled MySQL instance in OS X Server 10.6, see *Mac OS X Server: Web Technologies Administration Version 10.6 Snow Leopard*.

- The MySQL server bundled with OS X Server does not include the MySQL client libraries and header files required to access and use MySQL from a third-party driver, such as Perl DBI or PHP. For more...
information on obtaining and installing MySQL libraries, see OS X Server version 10.5: MySQL libraries available for download. Alternatively, you can ignore the bundled MySQL server and install MySQL from the package or tarball installation.
Chapter 7 Installing MySQL on Linux

Table of Contents

7.1 Installing MySQL on Linux Using RPM Packages .......................................................... 123
7.2 Installing MySQL on Linux Using Native Package Managers ....................................... 128

Linux supports a number of different solutions for installing MySQL. The recommended method is to use one of the distributions from Oracle. If you choose this method, there are several options available:

- Installing from a generic binary package in .tar.gz format. See Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries for more information.
- Extracting and compiling MySQL from a source distribution. For detailed instructions, see Chapter 4, Installing MySQL from Source.
- Installing using a precompiled RPM package. For more information, see Section 7.1, “Installing MySQL on Linux Using RPM Packages”.

As an alternative, you can use the native package manager within your Linux distribution to automatically download and install MySQL for you. Native package installations can take care of the download and dependencies required to run MySQL, but the MySQL version will often be some versions behind the currently available release. You will also normally be unable to install development releases, as these are not usually made available in the native repository. For more information on using the native package installers, see Section 7.2, “Installing MySQL on Linux Using Native Package Managers”.

Note
For many Linux installations, you will want to set up MySQL to be started automatically when your machine starts. Many of the native package installations perform this operation for you, but for source, binary and RPM solutions you may need to set this up separately. The required script, mysql.server, can be found in the support-files directory under the MySQL installation directory or in a MySQL source tree. You can install it as /etc/init.d/mysql for automatic MySQL startup and shutdown. See mysql.server — MySQL Server Startup Script.

7.1 Installing MySQL on Linux Using RPM Packages

The recommended way to install MySQL on RPM-based Linux distributions is by using the RPM packages. The RPMs that we provide to the community should work on all versions of Linux that support RPM packages and use glibc 2.3. To obtain RPM packages, see Section 2.2, “How to Get MySQL”.

For non-RPM Linux distributions, you can install MySQL using a .tar.gz package. See Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

Installations created from our Linux RPM distributions result in files under the system directories shown in the following table.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/bin</td>
<td>Client programs and scripts</td>
</tr>
<tr>
<td>/usr/sbin</td>
<td>The mysqld server</td>
</tr>
</tbody>
</table>
### Directory Contents of Directory

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/lib/mysql</td>
<td>Log files, databases</td>
</tr>
<tr>
<td>/usr/share/info</td>
<td>MySQL manual in Info format</td>
</tr>
<tr>
<td>/usr/share/man</td>
<td>Unix manual pages</td>
</tr>
<tr>
<td>/usr/include/mysql</td>
<td>Include (header) files</td>
</tr>
<tr>
<td>/usr/lib/mysql</td>
<td>Libraries</td>
</tr>
<tr>
<td>/usr/share/mysql</td>
<td>Miscellaneous support files, including error messages, character set files, sample configuration files, SQL for database installation</td>
</tr>
<tr>
<td>/usr/share/sql-bench</td>
<td>Benchmarks</td>
</tr>
</tbody>
</table>

### Note

RPM distributions of MySQL are also provided by other vendors. Be aware that they may differ from those built by Oracle in features, capabilities, and conventions (including communication setup), and that the instructions in this manual do not necessarily apply to installing them. The vendor's instructions should be consulted instead. Because of these differences, RPM packages built by Oracle check whether such RPMs built by other vendors are installed. If so, the RPM does not install and produces a message explaining this.

Conflicts can arise when an RPM from another vendor is already installed, such as when a vendor's conventions about which files belong with the server and which belong with the client library differ from the breakdown used for Oracle packages. In such cases, attempts to install an Oracle RPM with `rpm -i` may result in messages that files in the RPM to be installed conflict with files from an installed package (denoted `mysql-libs` in the following paragraphs).

Each MySQL release provides a `MySQL-shared-compat` package that is meant to replace `mysql-libs` and provides a replacement-compatible client library for older MySQL series. `MySQL-shared-compat` is set up to make `mysql-libs` obsolete, but `rpm` explicitly refuses to replace obsoleted packages when invoked with `-i` (unlike `-U`), which is why installation with `rpm -i` produces a conflict.

`MySQL-shared-compat` can safely be installed alongside `mysql-libs` because libraries are installed to different locations. Therefore, it is possible to install `MySQL-shared-compat` first, then manually remove `mysql-libs` before continuing with the installation. After `mysql-libs` is removed, the dynamic linker stops looking for the client library in the location where `mysql-libs` puts it, and the library provided by the `MySQL-shared-compat` package takes over.

Another alternative is to install packages using `yum`. In a directory containing all RPM packages for a MySQL release, `yum install MySQL*rpm` installs them in the correct order and removes `mysql-libs` in one step without conflicts.

In most cases, you need install only the `MySQL-server` and `MySQL-client` packages to get a functional standard MySQL installation. The other packages are not required for a standard installation.

### RPMs for MySQL Cluster

Beginning with MySQL 5.1.24, standard MySQL server RPMs built by MySQL no longer provide support for the `NDBCLUSTER` storage engine. MySQL Cluster users should check the MySQL Cluster Downloads page at [http://dev.mysql.com/downloads/cluster/](http://dev.mysql.com/downloads/cluster/) for RPMs that should work with most Linux distributions for both of these release series.
Important
When upgrading a MySQL Cluster RPM installation, you must upgrade all installed RPMs, including the Server and Client RPMs.

For more information about installing MySQL Cluster from RPMs, see Installing MySQL Cluster from RPM.

For upgrades, if your installation was originally produced by installing multiple RPM packages, it is best to upgrade all the installed packages, not just some. For example, if you previously installed the server and client RPMs, do not upgrade just the server RPM.

The following list shows the available RPM packages. The names shown here use a suffix of .glibc23.i386.rpm, but particular packages can have different suffixes, described later.

- **MySQL-server-VERSION.glibc23.i386.rpm**
  The MySQL server. You need this unless you only want to connect to a MySQL server running on another machine.

- **MySQL-client-VERSION.glibc23.i386.rpm**
  The standard MySQL client programs. You probably always want to install this package.

- **MySQL-devel-VERSION.glibc23.i386.rpm**
  The libraries and include files needed to compile other MySQL clients, such as the Perl MySQL module. Install this RPM if you intend to compile C API applications.

- **MySQL-debuginfo-VERSION.glibc23.i386.rpm**
  This package contains debugging information. It is specific to Red Hat Enterprise Linux. debuginfo RPMs are never needed to use MySQL software; this is true both for the server and for client programs. However, they contain additional information that might be needed by a debugger to analyze a crash.

- **MySQL-shared-VERSION.glibc23.i386.rpm**
  This package contains the shared libraries (libmysqlclient.so*) that certain languages and applications need to dynamically load and use MySQL. It contains single-threaded and thread-safe libraries. Install this RPM if you intend to compile or run C API applications that depend on the shared client library. If you install this package, do not install the MySQL-shared-compat package.

- **MySQL-shared-compat-VERSION.glibc23.i386.rpm**
  This package includes the shared libraries for older releases, up to the current release. It contains single-threaded and thread-safe libraries. Install this package instead of MySQL-shared if you have applications installed that are dynamically linked against older versions of MySQL but you want to upgrade to the current version without breaking the library dependencies.

- **MySQL-shared-compat-advanced-gpl-VERSION.glibc23.i386.rpm, MySQL-shared-compat-advanced-VERSION.glibc23.i386.rpm**
  These are like the MySQL-shared-compat package, but are for the "MySQL Enterprise Server – Advanced Edition" products. Install these packages rather than the normal MySQL-shared-compat package if you want to included shared client libraries for older MySQL versions.

- **MySQL-embedded-VERSION.glibc23.i386.rpm**
  The embedded MySQL server library.
• MySQL-ndb-management-VERSION.glibc23.i386.rpm, MySQL-ndb-storage-VERSION.glibc23.i386.rpm, MySQL-ndb-tools-VERSION.glibc23.i386.rpm, MySQL-ndb-extra-VERSION.glibc23.i386.rpm

Packages that contain additional files for MySQL Cluster installations.

**Note**

The MySQL-ndb-tools RPM requires a working installation of perl. Prior to MySQL 5.1.18, the DBI and HTML::Template packages were also required. See Chapter 13, Perl Installation Notes, and ndb_size.pl — NDBCLUSTER Size Requirement Estimator, for more information.

• MySQL-test-VERSION.glibc23.i386.rpm

This package includes the MySQL test suite.

• MySQL-VERSION.src.rpm

This contains the source code for all of the previous packages. It can also be used to rebuild the RPMs on other architectures (for example, SPARC).

In RPM package names, the suffix (following the VERSION value) has the following syntax:

```
.PLATFORM.CPU.rpm
```

The PLATFORM and CPU values indicate the type of system for which the package is built. PLATFORM indicates the platform and CPU indicates the processor type or family.

All packages are dynamically linked against glibc 2.3. The PLATFORM value indicates whether the package is platform independent or intended for a specific platform, as shown in the following table.

**Table 7.2 MySQL Linux RPM Package Platforms**

<table>
<thead>
<tr>
<th>PLATFORM Value</th>
<th>Intended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>glibc23</td>
<td>Platform independent, should run on any Linux distribution that supports glibc 2.3</td>
</tr>
<tr>
<td>rhel4, rhel5</td>
<td>Red Hat Enterprise Linux 4 or 5</td>
</tr>
<tr>
<td>sles10, sles11</td>
<td>SuSE Linux Enterprise Server 10 or 11</td>
</tr>
</tbody>
</table>

In MySQL 5.1, only glibc23 packages are available currently.

The CPU value indicates the processor type or family for which the package is built, as shown in the following table.

**Table 7.3 MySQL Linux RPM Package CPU Identifiers**

<table>
<thead>
<tr>
<th>CPU Value</th>
<th>Intended Processor Type or Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>i386, i586, i686</td>
<td>Pentium processor or better, 32 bit</td>
</tr>
<tr>
<td>x86_64</td>
<td>64-bit x86 processor</td>
</tr>
<tr>
<td>ia64</td>
<td>Itanium (IA-64) processor</td>
</tr>
</tbody>
</table>

To see all files in an RPM package (for example, a MySQL-server RPM), run a command like this (modify the platform and CPU identifiers appropriately for your system):
Installing MySQL on Linux Using RPM Packages

shell> rpm -qpl MySQL-server-VERSION.glibc23.i386.rpm

To perform a standard minimal installation, install the server and client RPMs:

shell> rpm -i MySQL-server-VERSION.glibc23.i386.rpm
shell> rpm -i MySQL-client-VERSION.glibc23.i386.rpm

To install only the client programs, install just the client RPM:

shell> rpm -i MySQL-client-VERSION.glibc23.i386.rpm

RPM provides a feature to verify the integrity and authenticity of packages before installing them. To learn more about this feature, see Section 2.3, “Verifying Package Integrity Using MD5 Checksums or GnuPG”.

The server RPM places data under the /var/lib/mysql directory. The RPM also creates a login account for a user named mysql (if one does not exist) to use for running the MySQL server, and creates the appropriate entries in /etc/init.d/ to start the server automatically at boot time. (This means that if you have performed a previous installation and have made changes to its startup script, you may want to make a copy of the script so that you can reinstall it after you install a newer RPM.) See Section 10.5, “Starting and Stopping MySQL Automatically”, for more information on how MySQL can be started automatically on system startup.

In MySQL 5.1.49 and later, during an upgrade installation using RPM packages, if the MySQL server is running when the upgrade occurs, the MySQL server is stopped, the upgrade occurs, and the MySQL server is restarted. If the MySQL server is not already running when the RPM upgrade occurs, the MySQL server is not started at the end of the installation.

If something goes wrong, you can find more information in the binary installation section. See Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

**Note**

The accounts created in the MySQL grant tables for an RPM installation initially have no passwords. After starting the server, you should assign passwords to them using the instructions in Chapter 10, Postinstallation Setup and Testing.

An RPM installation creates a user named mysql and a group named mysql on the system using the useradd, groupadd, and usermod commands. Those commands require appropriate administrative privileges, which is required for locally managed users and groups (as listed in the /etc/passwd and /etc/group files) by the RPM installation process being run by root.

If you log in as the mysql user, you may find that MySQL displays “Invalid (old?) table or database name” errors that mention .mysqlgui, lost+found, .mysqlgui, .bash_history, .fonts.cache-1, .lesshist, .mysql_history, .profile, .viminfo, and similar files created by MySQL or operating system utilities. You can safely ignore these error messages or remove the files or directories that cause them if you do not need them.

For nonlocal user management (LDAP, NIS, and so forth), the administrative tools may require additional authentication (such as a password), and will fail if the installing user does not provide this authentication. Even if they fail, the RPM installation will not abort but succeed, and this is intentional. If they failed, some of the intended transfer of ownership may be missing, and it is recommended that the system administrator then manually ensures some appropriate user and group exists and manually transfers ownership following the actions in the RPM spec file.
7.2 Installing MySQL on Linux Using Native Package Managers

Many Linux distributions include a version of the MySQL server, client tools, and development components into the standard package management system built into distributions such as Fedora, Debian, Ubuntu, and Gentoo. This section provides basic instructions for installing MySQL using these systems.

**Important**

Native package installations can take care of the download and dependencies required to run MySQL, but the MySQL version will often be some way behind the currently available release. You will also normally be unable to install development releases, as these are not usually made available in the native repository.

Distribution specific instructions are shown below:

- **Red Hat Linux, Fedora, CentOS**

For Red Hat and similar distributions, the MySQL distribution is divided into a number of separate packages, `mysql` for the client tools, `mysql-server` for the server and associated tools, and `mysql-libs` for the libraries. The libraries are required if you want to provide connectivity from different languages and environments such as Perl, Python and others.

To install, use the `yum` command to specify the packages that you want to install. For example:

```bash
root-shell> yum install mysql mysql-server mysql-libs mysql-server
```

```
Loaded plugins: presto, refresh-packagekit

Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package mysql.x86_64 0:5.1.48-2.fc13 set to be updated
---> Package mysql-libs.x86_64 0:5.1.48-2.fc13 set to be updated
---> Package mysql-server.x86_64 0:5.1.48-2.fc13 set to be updated
---> Processing Dependency: perl-DBD-MySQL for package: mysql-server-5.1.48-2.fc13.x86_64
---> Running transaction check
---> Package perl-DBD-MySQL.x86_64 0:4.017-1.fc13 set to be updated
---> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package               Arch          Version               Repository      Size
================================================================================
Installing:
  mysql                 x86_64        5.1.48-2.fc13         updates        889 k
  mysql-libs            x86_64        5.1.48-2.fc13         updates        1.2 M
  mysql-server          x86_64        5.1.48-2.fc13         updates        8.1 M
Installing for dependencies:
  perl-DBD-MySQL        x86_64        4.017-1.fc13          updates        136 k

Transaction Summary
================================================================================
Install       4 Package(s)
Upgrade       0 Package(s)
Total download size: 10 M
Installed size: 30 M
Is this ok [y/N]: y

Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
```

Package(s) data still to download: 10 M
```

(1/4): mysql-5.1.48-2.fc13.x86_64.rpm | 889 kB 00:04
(2/4): mysql-libs-5.1.48-2.fc13.x86_64.rpm | 1.2 MB 00:06
(3/4): mysql-server-5.1.48-2.fc13.x86_64.rpm | 8.1 MB 00:40
(4/4): perl-DBD-MySQL-4.017-1.fc13.x86_64.rpm | 136 kB 00:00
```
Installing MySQL on Linux Using Native Package Managers

MySQL and the MySQL server should now be installed. A sample configuration file is installed into /etc/my.cnf. An init script, to start and stop the server, will have been installed into /etc/init.d/mysqld. To start the MySQL server use service:

```bash
root-shell> service mysqld start
```

To enable the server to be started and stopped automatically during boot, use chkconfig:

```bash
root-shell> chkconfig --levels 235 mysqld on
```

Which enables the MySQL server to be started (and stopped) automatically at the specified the run levels.

The database tables will have been automatically created for you, if they do not already exist. You should, however, run `mysql_secure_installation` to set the root passwords on your server.

- **Debian, Ubuntu, Kubuntu**

On Debian and related distributions, there are two packages, `mysql-client` and `mysql-server`, for the client and server components respectively. You should specify an explicit version, for example `mysql-client-5.1`, to ensure that you install the version of MySQL that you want.

To download and install, including any dependencies, use the `apt-get` command, specifying the packages that you want to install.

```bash
root-shell> apt-get install mysql-client-5.1 mysql-server-5.1
```

Before installing, make sure that you update your `apt-get` index files to ensure you are downloading the latest available version.

A sample installation of the MySQL packages might look like this (some sections trimmed for clarity):

```bash
root-shell> apt-get install mysql-client-5.1 mysql-server-5.1
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
linux-headers-2.6.28-11 linux-headers-2.6.28-11-generic
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
    bsd-mailx libdbd-mysql-perl libdbi-perl libhtml-template-perl
    libmysqlclient30 libmysqlclient16 libnet-daemon-perl libplrpc-perl mailx
    mysql-common postfix
complete!
```
Installing MySQL on Linux Using Native Package Managers

Suggested packages:
- dbishell libipc-sharedcache-perl tinyca procmail postfix-mysql postfix-pgsql
- postfix-ldap postfix-pcpe sanl2-bin resolvconf postfix-cdb

The following NEW packages will be installed
- bsd-mailx libdbd-mysql-perl libdbi-perl libhtml-template-perl
- libmysqldclient15off libmysqlclient16 libnet-daemon-perl libplrpc-perl mailx
- mysql-client-5.1 mysql-common mysql-server-5.1 postfix

0 upgraded, 13 newly installed, 0 to remove and 182 not upgraded.

Need to get 1907kB/25.3MB of archives.
After this operation, 59.5MB of additional disk space will be used.
Do you want to continue [Y/n]? Y

Get: 1 http://gb.archive.ubuntu.com jaunty-updates/main mysql-common 5.1.30really5.0.75-0ubuntu10.5 [63.6kB]
Get: 2 http://gb.archive.ubuntu.com jaunty-updates/main libmysqlclient15off 5.1.30really5.0.75-0ubuntu10.5 [1843kB]
Fetched 1907kB in 9s (205kB/s)

Preconfiguring packages ...
Selecting previously deselected package mysql-common.
(Reading database ... 121260 files and directories currently installed.)
...
Processing 1 added doc-base file(s)...
Registering documents with scrollkeeper...
Setting up libnet-daemon-perl (0.43-1) ...
Setting up libplrpc-perl (0.2020-1) ...
Setting up libdbi-perl (1.607-1) ...
Setting up libmysqlclient15off (5.1.30really5.0.75-0ubuntu10.5) ...
Setting up libdbd-mysql-perl (4.008-1) ...
Setting up libmysqlclient16 (5.1.31-1ubuntu2) ...
Setting up mysql-client-5.1 (5.1.31-1ubuntu2) ...
Setting up mysql-server-5.1 (5.1.31-1ubuntu2) ...
  * Stopping MySQL database server mysql
    ...done.
100825 11:46:15  InnoDB: Started; log sequence number 0 46409
100825 11:46:15  InnoDB: Starting shutdown...
100825 11:46:17  InnoDB: Shutdown completed; log sequence number 0 46409
100825 11:46:17  [Warning] Forcing shutdown of 1 plugins
  * Starting MySQL database server mysql
    ...done.
  * Checking for corrupt, not cleanly closed and upgrade needing tables.
...
Processing triggers for libc6 ...
ldconfig deferred processing now taking place

Note

The apt-get command will install a number of packages, including the MySQL server, in order to provide the typical tools and application environment. This can mean that you install a large number of packages in addition to the main MySQL package.

During installation, the initial database will be created, and you will be prompted for the MySQL root password (and confirmation). A configuration file will have been created in /etc/mysql/my.cnf. An init script will have been created in /etc/init.d/mysql.

The server will already be started. You can manually start and stop the server using:

```
root-shell> service mysql [start|stop]
```

The service will automatically be added to the 2, 3 and 4 run levels, with stop scripts in the single, shutdown and restart levels.

- Gentoo Linux
As a source-based distribution, installing MySQL on Gentoo involves downloading the source, patching the Gentoo specifics, and then compiling the MySQL server and installing it. This process is handled automatically by the `emerge` command. Depending on the version of MySQL that you want to install, you may need to unmask the specific version that you want for your chosen platform.

The MySQL server and client tools are provided within a single package, `dev-db/mysql`. You can obtain a list of the versions available to install by looking at the portage directory for the package:

```
root-shell> ls /usr/portage/dev-db/mysql/mysql-5.1*
mysql-5.1.39-r1.ebuild
mysql-5.1.44-r1.ebuild
mysql-5.1.44-r2.ebuild
mysql-5.1.44-r3.ebuild
mysql-5.1.44.ebuild
mysql-5.1.45-r1.ebuild
mysql-5.1.45.ebuild
mysql-5.1.46.ebuild
```

To install a specific MySQL version, you must specify the entire atom. For example:

```
root-shell> emerge =dev-db/mysql-5.1.46
```

A simpler alternative is to use the `virtual/mysql-5.1` package, which will install the latest version:

```
root-shell> emerge =virtual/mysql-5.1
```

If the package is masked (because it is not tested or certified for the current platform), use the `ACCEPT_KEYWORDS` environment variable. For example:

```
root-shell> ACCEPT_KEYWORDS="~x86" emerge =virtual/mysql-5.1
```

After installation, you should create a new database using `mysql_install_db`, and set the password for the root user on MySQL. You can use the configuration interface to set the password and create the initial database:

```
root-shell> emerge --config =dev-db/mysql-5.1.46
```

A sample configuration file will have been created for you in `/etc/mysql/my.cnf`, and an init script will have been created in `/etc/init.d/mysql`.

To enable MySQL to start automatically at the normal (default) run levels, you can use:

```
root-shell> rc-update add mysql default
```
Chapter 8 Installing MySQL on Solaris and OpenSolaris

Table of Contents

8.1 Installing MySQL on Solaris Using a Solaris PKG ............................................................. 134
8.2 Installing MySQL on OpenSolaris Using IPS ................................................................. 135

MySQL on Solaris and OpenSolaris is available in a number of different formats.

• For information on installing using the native Solaris PKG format, see Section 8.1, “Installing MySQL on Solaris Using a Solaris PKG”.

• On OpenSolaris, the standard package repositories include MySQL packages specially built for OpenSolaris that include entries for the Service Management Framework (SMF) to enable control of the installation using the SMF administration commands. For more information, see Section 8.2, “Installing MySQL on OpenSolaris Using IPS”.

• To use a standard tar binary installation, use the notes provided in Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries. Check the notes and hints at the end of this section for Solaris specific notes that you may need before or after installation.

• For information on installing MySQL on Solaris or OpenSolaris using a source distribution, first check the Solaris advice, Section 4.8, “Notes on Installing MySQL on Solaris from Source”. For detailed instructions on installing from source, see Chapter 4, Installing MySQL from Source.

To obtain a binary MySQL distribution for Solaris in tarball or PKG format, http://dev.mysql.com/downloads/mysql/5.1.html.

Additional notes to be aware of when installing and using MySQL on Solaris:

• If you want to use MySQL with the mysql user and group, use the groupadd and useradd commands:

```bash
groupadd mysql
useradd -g mysql -s /bin/false mysql
```

• If you install MySQL using a binary tarball distribution on Solaris, you may run into trouble even before you get the MySQL distribution unpacked, as the Solaris tar cannot handle long file names. This means that you may see errors when you try to unpack MySQL.

If this occurs, you must use GNU tar (gtar) to unpack the distribution. In Solaris 10 and OpenSolaris gtar is normally located in /usr/sfw/bin/gtar, but may not be included in the default path definition.

• When using Solaris 10 for x86_64, you should mount any file systems on which you intend to store InnoDB files with the forcedirectio option. (By default mounting is done without this option.) Failing to do so will cause a significant drop in performance when using the InnoDB storage engine on this platform.

• If you would like MySQL to start automatically, you can copy support-files/mysql.server to /etc/init.d and create a symbolic link to it named /etc/rc3.d/S99mysql.server.

• If too many processes try to connect very rapidly to mysqld, you should see this error in the MySQL log:
Error in accept: Protocol error

You might try starting the server with the --back_log=50 option as a workaround for this.

- To configure the generation of core files on Solaris you should use the coreadm command. Because of the security implications of generating a core on a setuid() application, by default, Solaris does not support core files on setuid() programs. However, you can modify this behavior using coreadm. If you enable setuid() core files for the current user, they will be generated using the mode 600 and owned by the superuser.

8.1 Installing MySQL on Solaris Using a Solaris PKG

You can install MySQL on Solaris and OpenSolaris using a binary package using the native Solaris PKG format instead of the binary tarball distribution.

To use this package, download the corresponding mysql-VERSION-solaris10-PLATFORM.pkg.gz file, then uncompress it. For example:

```
shell> gunzip mysql-5.1.73-solaris10-x86_64.pkg.gz
```

To install a new package, use pkgadd and follow the onscreen prompts. You must have root privileges to perform this operation:

```
shell> pkgadd -d mysql-5.1.73-solaris10-x86_64.pkg
```

The following packages are available:

```
  1 mysql MySQL Community Server (GPL)
(i86pc) 5.1.73
```

Select package(s) you wish to process (or 'all' to process all packages). (default: all) [?,??,q]:

The PKG installer installs all of the files and tools needed, and then initializes your database if one does not exist. To complete the installation, you should set the root password for MySQL as provided in the instructions at the end of the installation. Alternatively, you can run the `mysql_secure_installation` script that comes with the installation.

By default, the PKG package installs MySQL under the root path /opt/mysql. You can change only the installation root path when using pkgadd, which can be used to install MySQL in a different Solaris zone. If you need to install in a specific directory, use a binary tar file distribution.

The pkg installer copies a suitable startup script for MySQL into /etc/init.d/mysql. To enable MySQL to startup and shutdown automatically, you should create a link between this file and the init script directories. For example, to ensure safe startup and shutdown of MySQL you could use the following commands to add the right links:

```
shell> ln /etc/init.d/mysql /etc/rc3.d/S91mysql
shell> ln /etc/init.d/mysql /etc/rc0.d/K02mysql
```

To remove MySQL, the installed package name is mysql. You can use this in combination with the pkgrm command to remove the installation.

To upgrade when using the Solaris package file format, you must remove the existing installation before installing the updated package. Removal of the package does not delete the existing database information, only the server, binaries and support files. The typical upgrade sequence is therefore:

```
shell> mysqladmin shutdown
```
You should check the notes in Chapter 11, Upgrading or Downgrading MySQL before performing any upgrade.

8.2 Installing MySQL on OpenSolaris Using IPS

OpenSolaris includes standard packages for MySQL in the core repository. The MySQL packages are based on a specific release of MySQL and updated periodically. For the latest release you must use either the native Solaris PKG, tar, or source installations. The native OpenSolaris packages include SMF files so that you can easily control your MySQL installation, including automatic startup and recovery, using the native service management tools.

To install MySQL on OpenSolaris, use the pkg command. You will need to be logged in as root, or use the pfexec tool, as shown in the example below:

```
shell> pfexec pkg install SUNWmysql51
```

The package set installs three individual packages, SUNWmysql51lib, which contains the MySQL client libraries; SUNWmysql51r which contains the root components, including SMF and configuration files; and SUNWmysql51u which contains the scripts, binary tools and other files. You can install these packages individually if you only need the corresponding components.

The MySQL files are installed into /usr/mysql which symbolic links for the sub directories (bin, lib, etc.) to a version specific directory. For MySQL 5.1, the full installation is located in /usr/mysql/5.1. The default data directory is /var/mysql/5.1/data. The configuration file is installed in /etc/mysql/5.1/my.cnf. This layout permits multiple versions of MySQL to be installed, without overwriting the data and binaries from other versions.

Once installed, you must run mysql_install_db to initialize the database, and use the mysql_secure_installation to secure your installation.

Using SMF to manage your MySQL installation

Once installed, you can start and stop your MySQL server using the installed SMF configuration. The service name is mysql, or if you have multiple versions installed, you should use the full version name, for example mysql:version_51. To start and enable MySQL to be started at boot time:

```
shell> svcadm enable mysql
```

To disable MySQL from starting during boot time, and shut the MySQL server down if it is running, use:

```
shell> svcadm disable mysql
```

To restart MySQL, for example after a configuration file changes, use the restart option:

```
shell> svcadm restart mysql
```

You can also use SMF to configure the data directory and enable full 64-bit mode. For example, to set the data directory used by MySQL:

```
shell> svcfg
svc:> select mysql:version_51
svc:/application/database/mysql:version_51> setprop mysql/data=/data0/mysql
```
By default, the 32-bit binaries are used. To enable the 64-bit server on 64-bit platforms, set the `enable_64bit` parameter. For example:

```
svc:/application/database/mysql:version_51> setprop mysql/enable_64bit=1
```

You need to refresh the SMF after setting these options:

```
shell> svcadm refresh mysql
```
Chapter 9 Installing MySQL on i5/OS

The i5/OS POWER MySQL package was created in cooperation with IBM. MySQL works within the Portable Application Solution Environment (PASE) on the System i series of hardware and will also provide database services for the Zend Core for i5/OS.

MySQL for i5/OS is provided both as a tar file and as a save file (.savf) package that can be downloaded and installed directly without any additional installation steps required. To install MySQL using the tar file, see Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

MySQL is only supported on i5/OS V5R4 or later releases. The i5/OS PASE must be installed for MySQL to operate. You must be able to login as a user in *SECOFR class.

You should the installation notes and tips for i5/OS before starting installation. See i5/OS Installation Notes [140].

Before Installation:

Note
The installation package will use an existing configuration if you have previously installed MySQL (which is identified by looking for the file /etc/my.cnf). The values for the data directory (DATADIR) and owner of the MySQL files (USRPRF) specified during the installation will be ignored, and the values determined from the /etc/my.cnf will be used instead.

If you want to change these parameters during a new install, you should temporarily rename /etc/my.cnf, install MySQL using the new parameters you want to use, and then merge your previous /etc/my.cnf configuration settings with the new /etc/my.cnf file that is created during installation.

• You must have a user profile with PASE with suitable privileges. The user should be within the *SECOFR class, such as the QSECOFR user ID. You can use the WRKUSRPRF command to check your user profile.

• For network connections to MySQL, you must have TCP/IP enabled. You should also check the following:

  • Ensure that a name has defined for the system. Run the Configure TCP/IP (CFGTCP) command and select option 12 (Change TCP/IP domain information) to display this setting. Make sure that a value is listed in the Host name field.

  • Make sure that the system has a loopback entry which represents the localhost or 127.0.0.1.

  • Ensure that the IP address of the IBM i machine is mapped correctly to the host name.

To install MySQL on i5/OS, follow these steps:

1. On the System i machine, create a save file that will be used to receive the downloaded installation save file. The file should be located within the General Purpose Library (QGPL):

   CRTSAVF FILE(QGPL/MYSQLINST) TESXT('MySQL Save file')

2. Download the MySQL installation save file in 32-bit (mysql-5.1.73-i5os-power-32bit.savf) or 64-bit (mysql-5.1.73-i5os-power-64bit.savf) from MySQL Downloads.

3. You need to FTP the downloaded .savf file directly into the QGPL/MYSQLINST file on the System i server. You can do this through FTP using the following steps after logging in to the System i machine:
ftp> bin
ftp> cd qgpl
ftp> put mysql-5.1.73-i5os-power.savf mysqlinst

4. Log into the System i server using a user in the *SECOFR class, such as the QSECOFR user ID.

5. You need to restore the installation library stored in the .savf save file:

   RSTLIB MYSQLINST DEV(*SAVF) SAVF(QGPL/MYSQLINST) MBROPT(*ALL) ALWOBJDIF(*ALL)

   **Note**
   You can ignore the security changes-type message at the bottom of the installation panel.

6. Once you have finished restoring the MYSQLINST library, check that all the necessary objects for installation are on the system by using the Display Library (DSPLIB) command:

   DSPLIB LIB(MYSQLINST)

7. You need to execute the installation command, MYSQLINST/INSMYSQL. You can specify three parameter settings during installation:

   - **DIR('/QOpenSys/usr/local/mysql')** sets the installation location for the MySQL files. The directory will be created if it does not already exist.
   
   - **DATADIR('/QOpenSys/usr/local/mysql/data')** sets the location of the directory that will be used to store the database files and binary logs. The default setting is /QOpenSys/usr/local/mysql/data. Note that if the installer detects an existing installation (due to the existence of /etc/my.cnf), then the existing setting will be used instead of the default.
   
   - **USRPRF(MYSQL)** sets the user profile that will own the files that are installed. The profile will be created if it does not already exist.

   **Note**
   You should choose an appropriate user for using the MySQL server installation. The user will be used whenever you need to do any administration on the MySQL server.

Once you have set the appropriate parameters, you can begin the installation.

The installation copies all the necessary files into a directory matching the DIR configuration value; sets the ownership on those files, sets up the MySQL environment and creates the MySQL configuration file (in /etc/my.cnf) completing all the steps in a typical binary installation process automatically. If this is a new installation of MySQL, or if the installer detects that this is a new version (because the /etc/my.cnf file does not exist), then the initial core MySQL databases will also be created during installation.

Once the installation has been completed, you will get a notice advising you to set the password for the root user. For more information, Chapter 10, Postinstallation Setup and Testing.

8. Once the installation has completed, you can delete the installation file:

   DLTLIB LIB(MYSQLINST)

**Upgrading an existing MySQL instance**
You need to execute the upgrade command, `MYSQLINST/UPGMYSQL`.

**Note**

You cannot use `MYSQLINST/UPGMYSQL` to upgrade between release series of MySQL (for example from 5.0 to 5.1). For information and advice on migrating between release series you can use the advice provided in Section 11.1.1, “Changes Affecting Upgrades to 5.1”.

You must specify 6 parameters to perform an upgrade:

- **DIR('/QOpenSys/usr/local/')**: Sets the installation location for the MySQL files. The directory will be created if it does not already exist. This is the directory that the MySQL server will be installed into, inside a directory with a name matching the version and release. For example, if installing MySQL 5.1.73 with the DIR set to /QOpenSys/usr/local/ would result in /QOpenSys/usr/local/mysql-5.1.73-i5os-power64 and a symbolic link to this directory will be created in /QOpenSys/usr/local/mysql.

- **DATADIR('/QOpenSys/mysql/data')**: Sets the location of the directory that will be upgraded.

- **USRPRF('MYSQL')**: Sets the user profile that will own the files that are installed. The profile will be created if it does not already exist; if it is created as part of the upgrade process, it will be disabled initially. You may wish to enable this user profile so that it can be used to start the MySQL server later. It is best practice to use the one previously created during the first installation.

- **MYSQLUSR('root user')**: Any user account in the current MySQL server with SUPER privileges.

- **PASSWORD('root user password')**: The password for the above account. This is necessary as the upgrade starts the MySQL server to upgrade the tables and the password is need to be able to shutdown the MySQL server.

- **CURINST('path to previous install')**: The full path to the installation that is being upgraded. For example an installation in /QOpenSys/usr/local/ will be /QOpenSys/usr/local/mysql-5.1.73-i5os-power64. Failure to specify this option may result in corruption of your existing data files.

For example:

```
MYSQLINST/UPGMYSQL DIR('/QOpenSys/usr/local/') DATADIR('/QOpenSys/mysql/data') USRPRF('MYSQL') MYSQLUSR('root') PASSWORD('root') CURINST('/QOpenSys/usr/local/mysql-5.1.73-i5os-power64')
```

You should receive a Program Message indicating `UPGRADE SUCCESSFUL!` upon completion or an error message if there is a problem. You can view the upgrade programs progression and the error in the text file `upgrade.log` in the installation directory.

**To start MySQL:**

1. Log into the System i server using the user profile create or specified during installation. By default, this is `MYSQL`.

**Note**

You should start `mysqld_safe` using a user that in the PASE environment has the id=0 (the equivalent of the standard Unix `root` user). If you do not use a user with this ID then the system will be unable to change the user when executing `mysqld` as set using `--user` option. If this happens, `mysqld` may be unable to read the files located within the MySQL data directory and the execution will fail.
2. Enter the PASE environment using `call qp2term`.

3. Start the MySQL server by changing to the installation directory and running `mysqld_safe`, specifying the user name used to install the server. The installer conveniently installs a symbolic link to the installation directory (`mysql-5.0.42-i5os-power-32bit`) as `/opt/mysql/mysql`:

   ```
   > cd /opt/mysql/mysql
   > bin/mysqld_safe --user=mysql &
   ```

   You should see a message similar to the following:

   ```
   Starting mysqld daemon with databases »
   from /opt/mysql/mysql-enterprise-5.0.42-i5os-power-32bit/data
   ```

   If you are having problems starting MySQL server, see Section 10.2.1, “Troubleshooting Problems Starting the MySQL Server”.

   **To stop MySQL:**

   1. Log into the System i server using the user profile create or specified during installation. By default, this is `MYSQL`.

   2. Enter the PASE environment using `call qp2term`.

   3. Stop the MySQL server by changing into the installation directory and running `mysqladmin`, specifying the user name used to install the server:

   ```
   > cd /opt/mysql/mysql
   > bin/mysqladmin -u root shutdown
   ```

   If the session that you started and stopped MySQL are the same, you may get the log output from `mysqld`:

   ```
   STOPPING server from pid file »
   /opt/mysql/mysql-enterprise-5.0.42-i5os-power-32bit/data/I5DBX.RCHLAND.IBM.COM.pid
   070718 10:34:20  mysqld ended
   ```

   If the sessions used to start and stop MySQL are different, you will not receive any confirmation of the shutdown.

   **Notes and tips**

   • A problem has been identified with the installation process on DBCS systems. If you are having problems install MySQL on a DBCS system, you need to change your job’s coded character set identifier (CSSID) to 37 (EBCDIC) before executing the install command, `INSMYSQL`. To do this, determine your existing CSSID (using DSPJOB and selecting option 2), execute `CHGJOB CSSID(37)`, run `INSMYSQL` to install MySQL and then execute `CHGJOB` again with your original CSSID.

   • If you want to use the Perl scripts that are included with MySQL, you need to download the iSeries Tools for Developers (5799-PTL). See [http://www-03.ibm.com/servers/enable/site/porting/tools/](http://www-03.ibm.com/servers/enable/site/porting/tools/).
Chapter 10 Postinstallation Setup and Testing

Table of Contents

10.1 Initializing the Data Directory ................................................................. 141
    10.1.1 Problems Running mysql_install_db ............................................. 143
10.2 Starting the Server ................................................................................. 145
    10.2.1 Troubleshooting Problems Starting the MySQL Server ..................... 146
10.3 Testing the Server .................................................................................. 148
10.4 Securing the Initial MySQL Accounts .................................................... 151
10.5 Starting and Stopping MySQL Automatically .......................................... 155

This section discusses tasks that you should perform after installing MySQL:

• If necessary, initialize the data directory and create the MySQL grant tables. For some MySQL installation methods, data directory initialization may be done for you automatically:
  • Installation on Windows
  • Installation on Linux using a server RPM distribution.
  • Installation using the native packaging system on many platforms, including Debian Linux, Ubuntu Linux, Gentoo Linux, and others.
  • Installation on OS X using a DMG distribution.

For other platforms and installation types, including installation from generic binary and source distributions, you must initialize the data directory yourself. For instructions, see Section 10.1, “Initializing the Data Directory”.

• For instructions, see Section 10.2, “Starting the Server”, and Section 10.3, “Testing the Server”.

• Assign passwords to any initial accounts in the grant tables, if that was not already done during data directory initialization. Passwords prevent unauthorized access to the MySQL server. You may also wish to restrict access to test databases. For instructions, see Section 10.4, “Securing the Initial MySQL Accounts”.

• Optionally, arrange for the server to start and stop automatically when your system starts and stops. For instructions, see Section 10.5, “Starting and Stopping MySQL Automatically”.

• Optionally, populate time zone tables to enable recognition of named time zones. For instructions, see MySQL Server Time Zone Support.

When you are ready to create additional user accounts, you can find information on the MySQL access control system and account management in The MySQL Access Privilege System, and MySQL User Account Management.

10.1 Initializing the Data Directory

After installing MySQL, you must initialize the data directory, including the tables in the mysql system database. For some MySQL installation methods, data directory initialization may be done automatically,
Initializing the Data Directory

as described in Chapter 10, Postinstallation Setup and Testing. For other installation methods, including installation from generic binary and source distributions, you must initialize the data directory yourself.

This section describes how to initialize the data directory on Unix and Unix-like systems. (For Windows, see Section 5.8, “Windows Postinstallation Procedures”.) For some suggested commands that you can use to test whether the server is accessible and working properly, see Section 10.3, “Testing the Server”.

In the examples shown here, the server runs under the user ID of the mysql login account. This assumes that such an account exists. Either create the account if it does not exist, or substitute the name of a different existing login account that you plan to use for running the server. For information about creating the account, see Creating a mysql System User and Group, in Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries.

1. Change location into the top-level directory of your MySQL installation, represented here by BASEDIR:

```
shell> cd BASEDIR
```

BASEDIR is likely to be something like /usr/local/mysql or /usr/local. The following steps assume that you have changed location to this directory.

You will find several files and subdirectories in the BASEDIR directory. The most important for installation purposes are the bin and scripts subdirectories, which contain the server as well as client and utility programs.

For some distribution types, mysqld is installed in the libexec directory.

2. If necessary, ensure that the distribution contents are accessible to mysql. If you installed the distribution as mysql, no further action is required. If you installed the distribution as root, its contents will be owned by root. Change its ownership to mysql by executing the following commands as root in the installation directory. The first command changes the owner attribute of the files to the mysql user. The second changes the group attribute to the mysql group.

```
shell> chown -R mysql .
shell> chgrp -R mysql .
```

3. If necessary, initialize the data directory, including the mysql database containing the initial MySQL grant tables that determine how users are permitted to connect to the server.

Typically, data directory initialization need be done only the first time you install MySQL. If you are upgrading an existing installation, you should run mysql_upgrade instead (see mysql_upgrade — Check and Upgrade MySQL Tables). However, the command that initializes the data directory does not overwrite any existing privilege tables, so it should be safe to run in any circumstances.

The exact location of mysql_install_db depends on the layout for your given installation. To initialize the grant tables, use one of the following commands, depending on whether mysql_install_db is located in the bin or scripts directory:

```
shell> scripts/mysql_install_db --user=mysql
shell> bin/mysql_install_db --user=mysql
```

It is important to make sure that the database directories and files are owned by the mysql login account so that the server has read and write access to them when you run it later. To ensure this if you run mysql_install_db as root, include the --user option as shown. Otherwise, you should execute the program while logged in as mysql, in which case you can omit the --user option from the command.
Problems Running mysql_install_db

The `mysql_install_db` command creates the server’s data directory. Under the data directory, it creates directories for the `mysql` database that holds the grant tables and the `test` database that you can use to test MySQL. The program also creates privilege table entries for the initial account or accounts. `test_`. For a complete listing and description of the grant tables, see The MySQL Access Privilege System.

It might be necessary to specify other options such as `--basedir` or `--datadir` if `mysql_install_db` does not identify the correct locations for the installation directory or data directory. For example:

```
shell> scripts/mysql_install_db --user=mysql 
   --basedir=/opt/mysql/mysql 
   --datadir=/opt/mysql/mysql/data
```

If you do not want to have the `test` database, you can remove it after starting the server, using the instructions in Section 10.4, “Securing the Initial MySQL Accounts”.

If you have trouble with `mysql_install_db` at this point, see Section 10.1.1, “Problems Running mysql_install_db”.

4. After initializing the data directory, you can establish the final installation ownership settings. To leave the installation owned by `mysql`, no action is required here. Otherwise, most of the MySQL installation can be owned by `root` if you like. The exception is that the data directory must be owned by `mysql`. To accomplish this, run the following commands as `root` in the installation directory. For some distribution types, the data directory might be named `var` rather than `data`; adjust the second command accordingly.

```
shell> chown -R root .
shell> chown -R mysql data
```

If the plugin directory (the directory named by the `plugin_dir` system variable) is writable by the server, it may be possible for a user to write executable code to a file in the directory using `SELECT ... INTO DUMPFILE`. This can be prevented by making the plugin directory read only to the server or by setting the `secure_file_priv` system variable at server startup to a directory where `SELECT` writes can be performed safely.

5. If you installed MySQL using a source distribution, you may want to optionally copy one of the provided configuration files from the `support-files` directory into your `/etc` directory. There are different sample configuration files for different use cases, server types, and CPU and RAM configurations. To use one of these standard files, copy it to `/etc/my.cnf`, or `/etc/mysql/my.cnf` and edit and check the configuration before starting your MySQL server for the first time.

You can also create `my.cnf` yourself and place into it the options the server should use at startup. See Server Configuration Defaults.

If you do not copy one of the standard configuration files or create your own, the MySQL server starts with its default settings.

6. If you want MySQL to start automatically when you boot your machine, see Section 10.5, “Starting and Stopping MySQL Automatically”.

Data directory initialization creates time zone tables in the `mysql` database but does not populate them. To do so, use the instructions in MySQL Server Time Zone Support.

10.1.1 Problems Running mysql_install_db
Problems Running mysql_install_db

The purpose of the `mysql_install_db` program is to initialize the data directory, including the tables in the `mysql` system database. It does not overwrite existing MySQL privilege tables, and it does not affect any other data.

To re-create your privilege tables, first stop the `mysqld` server if it is running. Then rename the `mysql` directory under the data directory to save it, and run `mysql_install_db`. Suppose that your current directory is the MySQL installation directory and that `mysql_install_db` is located in the `bin` directory and the data directory is named `data`. To rename the `mysql` database and re-run `mysql_install_db`, use these commands.

```shell
mv data/mysql data/mysql.old
scripts/mysql_install_db --user=mysql
```

When you run `mysql_install_db`, you might encounter the following problems:

- **`mysql_install_db` fails to install the grant tables**

  You may find that `mysql_install_db` fails to install the grant tables and terminates after displaying the following messages:

  ```
  Starting mysqld daemon with databases from XXXXXX
  mysqld ended
  ```

  In this case, you should examine the error log file very carefully. The log should be located in the directory `XXXXXX` named by the error message and should indicate why `mysqld` did not start. If you do not understand what happened, include the log when you post a bug report. See How to Report Bugs or Problems.

- **There is a `mysqld` process running**

  This indicates that the server is running, in which case the grant tables have probably been created already. If so, there is no need to run `mysql_install_db` at all because it needs to be run only once, when you first install MySQL.

- **Installing a second `mysqld` server does not work when one server is running**

  This can happen when you have an existing MySQL installation, but want to put a new installation in a different location. For example, you might have a production installation, but you want to create a second installation for testing purposes. Generally the problem that occurs when you try to run a second server is that it tries to use a network interface that is in use by the first server. In this case, you should see one of the following error messages:

  ```
  Can't start server: Bind on TCP/IP port: Address already in use
  Can't start server: Bind on unix socket...
  ```

  For instructions on setting up multiple servers, see Running Multiple MySQL Instances on One Machine.

- **You do not have write access to the `/tmp` directory**

  If you do not have write access to create temporary files or a Unix socket file in the default location (the `/tmp` directory) or the `TMPDIR` environment variable, if it has been set, an error occurs when you run `mysql_install_db` or the `mysqld` server.
Starting the Server

You can specify different locations for the temporary directory and Unix socket file by executing these commands prior to starting `mysql_install_db` or `mysqld`, where `some_tmp_dir` is the full path name to some directory for which you have write permission:

```
shell> TMPDIR=/some_tmp_dir/
shell> MYSQL_UNIX_PORT=/some_tmp_dir/mysql.sock
shell> export TMPDIR MYSQL_UNIX_PORT
```

Then you should be able to run `mysql_install_db` and start the server with these commands:

```
shell> scripts/mysql_install_db --user=mysql
shell> bin/mysqld_safe --user=mysql &
```

If `mysql_install_db` is located in the `bin` directory, modify the first command to `bin/mysql_install_db`.

See How to Protect or Change the MySQL Unix Socket File, and Chapter 12, Environment Variables.

There are some alternatives to running the `mysql_install_db` program provided in the MySQL distribution:

- If you want the initial privileges to be different from the standard defaults, use account-management statements such as `CREATE USER`, `GRANT`, and `REVOKE` to change the privileges after the grant tables have been set up. In other words, run `mysql_install_db`, and then use `mysql -u root mysql` to connect to the server as the MySQL root user so that you can issue the necessary statements. (See Account Management Statements.)

To install MySQL on several machines with the same privileges, put the `CREATE USER`, `GRANT`, and `REVOKE` statements in a file and execute the file as a script using `mysql` after running `mysql_install_db`. For example:

```
shell> scripts/mysql_install_db --user=mysql
shell> bin/mysql -u root < your_script_file
```

This enables you to avoid issuing the statements manually on each machine.

- It is possible to re-create the grant tables completely after they have previously been created. You might want to do this if you are just learning how to use `CREATE USER`, `GRANT`, and `REVOKE` and have made so many modifications after running `mysql_install_db` that you want to wipe out the tables and start over.

To re-create the grant tables, stop the server if it is running and remove the `mysql` database directory. Then run `mysql_install_db` again.

### 10.2 Starting the Server

This section describes how start the server on Unix and Unix-like systems. (For Windows, see Section 5.6.4, “Starting MySQL Server on Microsoft Windows for the First Time”.) For some suggested commands that you can use to test whether the server is accessible and working properly, see Section 10.3, “Testing the Server”.

Start the MySQL server like this:

```
shell> bin/mysqld_safe --user=mysql &
```
It is important that the MySQL server be run using an unprivileged (non-root) login account. To ensure this if you run `mysqld_safe` as `root`, include the `--user` option as shown. Otherwise, execute the program while logged in as `mysql`, in which case you can omit the `--user` option from the command.

For further instructions for running MySQL as an unprivileged user, see How to Run MySQL as a Normal User.

If the command fails immediately and prints `mysqld ended`, look for information in the error log (which by default is the `host_name.err` file in the data directory).

If the server is unable to access the data directory it starts or read the grant tables in the `mysql` database, it writes a message to its error log. Such problems can occur if you neglected to create the grant tables by initializing the data directory before proceeding to this step, or if you ran the command that initializes the data directory without the `--user` option. Remove the `data` directory and run the command with the `--user` option.

If you have other problems starting the server, see Section 10.2.1, “Troubleshooting Problems Starting the MySQL Server”. For more information about `mysqld_safe`, see `mysqld_safe — MySQL Server Startup Script`.

You can set up new accounts using the `bin/mysql_setpermission` script if you install the DBI and DBD::mysql Perl modules. See `mysql_setpermission — Interactively Set Permissions in Grant Tables`. For Perl module installation instructions, see Chapter 13, Perl Installation Notes.

If you would like to use `mysqlaccess` and have the MySQL distribution in some nonstandard location, you must change the location where `mysqlaccess` expects to find the `mysql` client. Edit the `bin/mysqlaccess` script at approximately line 18. Search for a line that looks like this:

```
$MYSQL     = '/usr/local/bin/mysql';    # path to mysql executable
```

Change the path to reflect the location where `mysql` actually is stored on your system. If you do not do this, a Broken pipe error will occur when you run `mysqlaccess`.

### 10.2.1 Troubleshooting Problems Starting the MySQL Server

This section provides troubleshooting suggestions for problems starting the server. For additional suggestions for Windows systems, see Section 5.7, “Troubleshooting a Microsoft Windows MySQL Server Installation”.

If you have problems starting the server, here are some things to try:

- Check the error log to see why the server does not start.
- Specify any special options needed by the storage engines you are using.
- Make sure that the server knows where to find the data directory.
- Make sure that the server can access the data directory. The ownership and permissions of the data directory and its contents must be set such that the server can read and modify them.
- Verify that the network interfaces the server wants to use are available.

Some storage engines have options that control their behavior. You can create a `my.cnf` file and specify startup options for the engines that you plan to use. If you are going to use storage engines that support transactional tables (InnoDB, NDB), be sure that you have them configured the way you want before starting the server.
If you are using InnoDB tables, see InnoDB Configuration.

If you are using MySQL Cluster, see Configuration of MySQL Cluster NDB 6.1-7.1.

Storage engines will use default option values if you specify none, but it is recommended that you review the available options and specify explicit values for those for which the defaults are not appropriate for your installation.

When the mysql server starts, it changes location to the data directory. This is where it expects to find databases and where it expects to write log files. The server also writes the pid (process ID) file in the data directory.

The data directory location is hardwired in when the server is compiled. This is where the server looks for the data directory by default. If the data directory is located somewhere else on your system, the server will not work properly. You can determine what the default path settings are by invokes mysql with the --verbose and --help options.

If the default locations do not match the MySQL installation layout on your system, you can override them by specifying options to mysql or mysqld_safe on the command line or in an option file.

To specify the location of the data directory explicitly, use the --datadir option. However, normally you can tell mysql the location of the base directory under which MySQL is installed and it looks for the data directory there. You can do this with the --basedir option.

To check the effect of specifying path options, invoke mysql with those options followed by the --verbose and --help options. For example, if you change location into the directory where mysql is installed and then run the following command, it shows the effect of starting the server with a base directory of /usr/local:

```
shell> ./mysqld --basedir=/usr/local --verbose --help
```

You can specify other options such as --datadir as well, but --verbose and --help must be the last options.

Once you determine the path settings you want, start the server without --verbose and --help.

If mysqld is currently running, you can find out what path settings it is using by executing this command:

```
shell> mysqladmin variables
```

Or:

```
shell> mysqladmin -h host_name variables
```

**host_name** is the name of the MySQL server host.

If you get **Errcode 13** (which means Permission denied) when starting mysql, this means that the privileges of the data directory or its contents do not permit server access. In this case, you change the permissions for the involved files and directories so that the server has the right to use them. You can also start the server as root, but this raises security issues and should be avoided.

Change location into the data directory and check the ownership of the data directory and its contents to make sure the server has access. For example, if the data directory is /usr/local/mysql/var, use this command:

```
shell> ls -la /usr/local/mysql/var
```
If the data directory or its files or subdirectories are not owned by the login account that you use for running the server, change their ownership to that account. If the account is named `mysql`, use these commands:

```
shell> chown -R mysql /usr/local/mysql/var
shell> chgrp -R mysql /usr/local/mysql/var
```

Even with correct ownership, MySQL might fail to start up if there is other security software running on your system that manages application access to various parts of the file system. In this case, reconfigure that software to enable `mysqld` to access the directories it uses during normal operation.

If the server fails to start up correctly, check the error log. Log files are located in the data directory (typically `C:\Program Files\MySQL\MySQL Server 5.1\data` on Windows, `/usr/local/mysql/data` for a Unix/Linux binary distribution, and `/usr/local/var` for a Unix/Linux source distribution). Look in the data directory for files with names of the form `host_name.err` and `host_name.log`, where `host_name` is the name of your server host. Then examine the last few lines of these files. You can use `tail` to display them:

```
shell> tail host_name.err
shell> tail host_name.log
```

The error log should contain information that indicates why the server could not start.

If either of the following errors occur, it means that some other program (perhaps another `mysqld` server) is using the TCP/IP port or Unix socket file that `mysqld` is trying to use:

- Can't start server: Bind on TCP/IP port: Address already in use
- Can't start server: Bind on unix socket...

Use `ps` to determine whether you have another `mysqld` server running. If so, shut down the server before starting `mysqld` again. (If another server is running, and you really want to run multiple servers, you can find information about how to do so in Running Multiple MySQL Instances on One Machine.)

If no other server is running, try to execute the command `telnet your_host_name tcp_ip_port_number`. (The default MySQL port number is 3306.) Then press Enter a couple of times. If you do not get an error message like `telnet: Unable to connect to remote host: Connection refused`, some other program is using the TCP/IP port that `mysqld` is trying to use. You will need to track down what program this is and disable it, or else tell `mysqld` to listen to a different port with the `--port` option. In this case, you will also need to specify the port number for client programs when connecting to the server using TCP/IP.

Another reason the port might be inaccessible is that you have a firewall running that blocks connections to it. If so, modify the firewall settings to permit access to the port.

If the server starts but you cannot connect to it, you should make sure that you have an entry in `/etc/hosts` that looks like this:

```
127.0.0.1 localhost
```

If you cannot get `mysqld` to start, you can try to make a trace file to find the problem by using the `--debug` option. See The DBUG Package.

## 10.3 Testing the Server

After the data directory is initialized and you have started the server, perform some simple tests to make sure that it works satisfactorily. This section assumes that your current location is the MySQL installation
Testing the Server

directory and that it has a bin subdirectory containing the MySQL programs used here. If that is not true, adjust the command path names accordingly.

Alternatively, add the bin directory to your PATH environment variable setting. That enables your shell (command interpreter) to find MySQL programs properly, so that you can run a program by typing only its name, not its path name. See Setting Environment Variables.

Use mysqladmin to verify that the server is running. The following commands provide simple tests to check whether the server is up and responding to connections:

shell> bin/mysqladmin version
shell> bin/mysqladmin variables

If you cannot connect to the server, specify a -u root option to connect as root. If you have assigned a password for the root account already, you'll also need to specify -p on the command line and enter the password when prompted. For example:

shell> bin/mysqladmin -u root -p version
Enter password: (enter root password here)

The output from mysqladmin version varies slightly depending on your platform and version of MySQL, but should be similar to that shown here:

shell> bin/mysqladmin version
mysqladmin Ver 14.12 Distrib 5.1.73, for pc-linux-gnu on i686 ...
Server version 5.1.73
Protocol version 10
Connection Localhost via UNIX socket
UNIX socket /var/lib/mysql/mysql.sock
Uptime: 14 days 5 hours 5 min 21 sec
Threads: 1 Questions: 366 Slow queries: 0
Opens: 0 Flush tables: 1 Open tables: 19
Queries per second avg: 0.000

To see what else you can do with mysqladmin, invoke it with the --help option.

Verify that you can shut down the server (include a -p option if the root account has a password already):

shell> bin/mysqladmin -u root shutdown

Verify that you can start the server again. Do this by using mysql_safe or by invoking mysqld directly. For example:

shell> bin/mysqld_safe --user=mysql &

If mysqld_safe fails, see Section 10.2.1, “Troubleshooting Problems Starting the MySQL Server”.

Run some simple tests to verify that you can retrieve information from the server. The output should be similar to that shown here.

Use mysqlshow to see what databases exist:

shell> bin/mysqlshow
+-------------------+
Testing the Server

<table>
<thead>
<tr>
<th>Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>information_schema</td>
</tr>
<tr>
<td>mysql</td>
</tr>
<tr>
<td>test</td>
</tr>
<tr>
<td>----------------</td>
</tr>
</tbody>
</table>

The list of installed databases may vary, but will always include the minimum of mysql and information_schema.

If you specify a database name, mysqlshow displays a list of the tables within the database:

```
shell> bin/mysqlshow mysql
Database: mysql
+---------------------------+
|          Tables           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| servers                   |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
```

Use the mysql program to select information from a table in the mysql database:

```
shell> bin/mysql -e "SELECT User, Host FROM mysql.user" mysql
+------+-----------+
| User | Host      |
+------+-----------+
| root | localhost |
+------+-----------+
```

There is a benchmark suite in the sql-bench directory (under the MySQL installation directory) that you can use to compare how MySQL performs on different platforms. The benchmark suite is written in Perl. It requires the Perl DBI module that provides a database-independent interface to the various databases, and some other additional Perl modules:

```
DBI
DBD::mysql
Data::Dumper
Data::ShowTable
```

These modules can be obtained from CPAN (http://www.cpan.org/). See also Section 13.1, “Installing Perl on Unix”.
The `sql-bench/Results` directory contains the results from many runs against different databases and platforms. To run all tests, execute these commands:

```shell
shell> cd sql-bench
shell> perl run-all-tests
```

If you do not have the `sql-bench` directory, you probably installed MySQL using RPM files other than the source RPM. (The source RPM includes the `sql-bench` benchmark directory.) In this case, you must first install the benchmark suite before you can use it. There are separate benchmark RPM files named `mysql-bench-VERSION.i386.rpm` that contain benchmark code and data.

If you have a source distribution, there are also tests in its `tests` subdirectory that you can run. For example, to run `auto_increment.tst`, execute this command from the top-level directory of your source distribution:

```shell
shell> mysql -vvf test < ./tests/auto_increment.tst
```

The expected result of the test can be found in the `./tests/auto_increment.res` file.

At this point, your server is running and you can access it. To tighten security if you have not yet assigned passwords to the initial account or accounts, follow the instructions in Section 10.4, “Securing the Initial MySQL Accounts”.

For more information about `mysql`, `mysqladmin`, and `mysqlshow`, see `mysql — The MySQL Command-Line Tool`, `mysqladmin — Client for Administering a MySQL Server`, and `mysqlshow — Display Database, Table, and Column Information`.

### 10.4 Securing the Initial MySQL Accounts

The MySQL installation process involves initializing the data directory, including the `mysql` database containing the grant tables that define MySQL accounts. For details, see Chapter 10, "Postinstallation Setup and Testing".

This section describes how to assign passwords to the initial accounts created during the MySQL installation procedure, if you have not already done so.

The `mysql.user` grant table defines the initial MySQL user accounts and their access privileges:

- Some accounts have the user name `root`. These are superuser accounts that have all privileges and can do anything. If these `root` accounts have empty passwords, anyone can connect to the MySQL server as `root without a password` and be granted all privileges.

- On Windows, `root` accounts are created that permit connections from the local host only. Connections can be made by specifying the host name `localhost` or the IP address `127.0.0.1`. If the user selects the `Enable root access from remote machines` option during installation, the Windows installer creates another `root` account that permits connections from any host.

- On Unix, each `root` account permits connections from the local host. Connections can be made by specifying the host name `localhost`, the IP address `127.0.0.1`, or the actual host name or IP address.

An attempt to connect to the host `127.0.0.1` normally resolves to the `localhost` account. However, this fails if the server is run with the `--skip-name-resolve` option, so the `127.0.0.1` account is useful in that case.
If accounts for anonymous users were created, these have an empty user name. The anonymous accounts have no password, so anyone can use them to connect to the MySQL server.

On Windows, there is one anonymous account that permits connections from the local host. Connections can be made by specifying a host name of `localhost`. The account has no global privileges. (Before MySQL 5.1.16, it has all global privileges, just like the `root` accounts.)

On Unix, each anonymous account permits connections from the local host. Connections can be made by specifying a host name of `localhost` for one of the accounts, or the actual host name or IP address for the other.

To display which accounts exist in the `mysql.user` table and check whether their passwords are empty, use the following statement:

```
mysql> SELECT User, Host, Password FROM mysql.user;
+------+--------------------+----------+
| User | Host               | Password |
+------+--------------------+----------+
| root | localhost          |          |
| root | myhost.example.com |          |
| root | 127.0.0.1          |          |
|      | localhost          |          |
|      | myhost.example.com |          |
+------+--------------------+----------+
```

This output indicates that there are several `root` and anonymous-user accounts, none of which have passwords. The output might differ on your system, but the presence of accounts with empty passwords means that your MySQL installation is unprotected until you do something about it:

- Assign a password to each MySQL `root` account that does not have one.

- To prevent clients from connecting as anonymous users without a password, either assign a password to each anonymous account or remove the accounts.

In addition, the `mysql.db` table contains rows that permit all accounts to access the `test` database and other databases with names that start with `test_`. This is true even for accounts that otherwise have no special privileges such as the default anonymous accounts. This is convenient for testing but inadvisable on production servers. Administrators who want database access restricted only to accounts that have permissions granted explicitly for that purpose should remove these `mysql.db` table rows.

The following instructions describe how to set up passwords for the initial MySQL accounts, first for the `root` accounts, then for the anonymous accounts. The instructions also cover how to remove anonymous accounts, should you prefer not to permit anonymous access at all, and describe how to remove permissive access to test databases. Replace `new_password` in the examples with the password that you want to use. Replace `host_name` with the name of the server host. You can determine this name from the output of the preceding `SELECT` statement. For the output shown, `host_name` is `myhost.example.com`.

```
Note
```
For additional information about setting passwords, see Assigning Account Passwords. If you forget your `root` password after setting it, see How to Reset the Root Password.

To set up additional accounts, see Adding User Accounts.

You might want to defer setting the passwords until later, to avoid the need to specify them while you perform additional setup or testing. However, be sure to set them before using your installation for production purposes.
Assigning root Account Passwords

A root account password can be set several ways. The following discussion demonstrates three methods:

• Use the SET PASSWORD statement

• Use the UPDATE statement

• Use the mysqladmin command-line client program

To assign passwords using SET PASSWORD, connect to the server as root and issue a SET PASSWORD statement for each root account listed in the mysql.user table.

For Windows, do this:

```shell
shell> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'%' = PASSWORD('new_password');
```

The last statement is unnecessary if the mysql.user table has no root account with a host value of %.

For Unix, do this:

```shell
shell> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('new_password');
```

You can also use a single statement that assigns a password to all root accounts by using UPDATE to modify the mysql.user table directly. This method works on any platform:

```shell
shell> mysql -u root
mysql> UPDATE mysql.user SET Password = PASSWORD('new_password')
    -> WHERE User = 'root';
mysql> FLUSH PRIVILEGES;
```

The FLUSH statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until you restart it.

To assign passwords to the root accounts using mysqladmin, execute the following commands:

```shell
shell> mysqladmin -u root password "new_password"
shell> mysqladmin -u root -h host_name password "new_password"
```

Those commands apply both to Windows and to Unix. The double quotation marks around the password are not always necessary, but you should use them if the password contains spaces or other characters that are special to your command interpreter.
Assigning Anonymous Account Passwords

The `mysqladmin` method of setting the `root` account passwords does not work for the
"root'@'127.0.0.1' account. Use the `SET PASSWORD` method shown earlier.

After the `root` passwords have been set, you must supply the appropriate password whenever you
connect as `root` to the server. For example, to shut down the server with `mysqladmin`, use this
command:

```
shell> mysqladmin -u root -p shutdown
Enter password: (enter root password here)
```

The `mysql` commands in the following instructions include a `-p` option based on the assumption that
you have assigned the `root` account passwords using the preceding instructions and must specify that
password when connecting to the server.

**Assigning Anonymous Account Passwords**

To assign passwords to the anonymous accounts, connect to the server as `root`, then use either `SET
PASSWORD` or `UPDATE`.

To use `SET PASSWORD` on Windows, do this:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('new_password');
```

To use `SET PASSWORD` on Unix, do this:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('new_password');
mysql> SET PASSWORD FOR ''@'host_name' = PASSWORD('new_password');
```

To set the anonymous-user account passwords with a single `UPDATE` statement, do this (on any platform):

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> UPDATE mysql.user SET Password = PASSWORD('new_password')
    -> WHERE User = '';
mysql> FLUSH PRIVILEGES;
```

The `FLUSH` statement causes the server to reread the grant tables. Without it, the password change
remains unnoticed by the server until you restart it.

**Removing Anonymous Accounts**

If you prefer to remove any anonymous accounts rather than assigning them passwords, do so as follows
on Windows:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> DROP USER ''@'localhost';
```

On Unix, remove the anonymous accounts like this:

```
shell> mysql -u root -p
```
Securing Test Databases

By default, the `mysql.db` table contains rows that permit access by any user to the `test` database and other databases with names that start with `test_`. (These rows have an empty `User` column value, which for access-checking purposes matches any user name.) This means that such databases can be used even by accounts that otherwise possess no privileges. If you want to remove any-user access to test databases, do so as follows:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> DELETE FROM mysql.db WHERE Db LIKE 'test%';
mysql> FLUSH PRIVILEGES;
```

The `FLUSH` statement causes the server to reread the grant tables. Without it, the privilege change remains unnoticed by the server until you restart it.

With the preceding change, only users who have global database privileges or privileges granted explicitly for the `test` database can use it. However, if you prefer that the database not exist at all, drop it:

```
mysql> DROP DATABASE test;
```

10.5 Starting and Stopping MySQL Automatically

This section discusses methods for starting and stopping the MySQL server.

Generally, you start the `mysqld` server in one of these ways:

- Invoke `mysqld` directly. This works on any platform.
- On Windows, you can set up a MySQL service that runs automatically when Windows starts. See Section 5.6.7, “Starting MySQL Server as a Microsoft Windows Service”.
- On Unix and Unix-like systems, you can invoke `mysqld_safe`, which tries to determine the proper options for `mysqld` and then runs it with those options. See `mysqld_safe — MySQL Server Startup Script`.
- On systems that use System V-style run directories (that is, `/etc/init.d` and run-level specific directories), invoke `mysql.server`. This script is used primarily at system startup and shutdown. It usually is installed under the name `mysql`. The `mysql.server` script starts the server by invoking `mysqld_safe`. See `mysql.server — MySQL Server Startup Script`.
- On OS X, install a launchd daemon to enable automatic MySQL startup at system startup. The daemon starts the server by invoking `mysqld_safe`. For details, see Section 6.3, “Installing a MySQL Launch Daemon”. A MySQL Preference Pane also provides control for starting and stopping MySQL through the System Preferences. See Section 6.5, “Installing and Using the MySQL Preference Pane”.
- On Solaris/OpenSolaris, use the service management framework (SMF) system to initiate and control MySQL startup. For more information, see Section 8.2, “Installing MySQL on OpenSolaris Using IPS”.

The `mysqld_safe` and `mysql.server` scripts, Solaris/OpenSolaris SMF, and the OS X Startup Item (or MySQL Preference Pane) can be used to start the server manually, or automatically at system startup time. `mysql.server` and the Startup Item also can be used to stop the server.
The following table shows which option groups the server and startup scripts read from option files.

### Table 10.1 MySQL Startup Scripts and Supported Server Option Groups

<table>
<thead>
<tr>
<th>Script</th>
<th>Option Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqlld</td>
<td>[mysqlld], [server], [mysqlld-major_version]</td>
</tr>
<tr>
<td>mysqlld_safe</td>
<td>[mysqlld], [server], [mysqlld_safe]</td>
</tr>
<tr>
<td>mysql.server</td>
<td>[mysqlld], [mysql.server], [server]</td>
</tr>
</tbody>
</table>

[mysqlld-major_version] means that groups with names like [mysqlld-5.0] and [mysqlld-5.1] are read by servers having versions 5.0.x, 5.1.x, and so forth. This feature can be used to specify options that can be read only by servers within a given release series.

For backward compatibility, mysql.server also reads the [mysql_server] group and mysqlld_safe also reads the [safe_mysqld] group. However, you should update your option files to use the [mysql.server] and [mysql_safe] groups instead.

For more information on MySQL configuration files and their structure and contents, see Using Option Files.
Chapter 11 Upgrading or Downgrading MySQL

Table of Contents

11.1 Upgrading MySQL .............................................................. 157
  11.1.1 Changes Affecting Upgrades to 5.1 ................................. 161
11.2 Downgrading MySQL .......................................................... 172
  11.2.1 Changes Affecting Downgrades from MySQL 5.1 .......... 175
  11.2.2 Downgrading from MySQL Enterprise Edition to MySQL Community Server ...... 176
11.3 Checking Whether Tables or Indexes Must Be Rebuilt ............... 177
11.4 Rebuilding or Repairing Tables or Indexes ............................. 179
11.5 Copying MySQL Databases to Another Machine .................... 180

This section describes the steps to upgrade or downgrade a MySQL installation.

Upgrading is a common procedure, as you pick up bug fixes within the same MySQL release series or significant features between major MySQL releases. You perform this procedure first on some test systems to make sure everything works smoothly, and then on the production systems.

Downgrading is less common. Typically, you undo an upgrade because of some compatibility or performance issue that occurs on a production system, and was not uncovered during initial upgrade verification on the test systems. As with the upgrade procedure, perform and verify the downgrade procedure on some test systems first, before using it on a production system.

11.1 Upgrading MySQL

This section describes how to upgrade to a new MySQL version.

- Supported Upgrade Methods
- Supported Upgrade Paths
- Before You Begin
- Performing an In-place Upgrade
- Performing a Logical Upgrade
- Upgrade Troubleshooting

Supported Upgrade Methods

- In-place Upgrade: Involves shutting down the old MySQL version, replacing the old MySQL binaries or packages with the new ones, restarting MySQL on the existing data directory, and running mysql_upgrade.

- Logical Upgrade: Involves exporting existing data from the old MySQL version using mysqldump, installing the new MySQL version, loading the dump file into the new MySQL version, and running mysql_upgrade.

Note

MySQL recommends a mysqldump upgrade when upgrading from a previous release. For example, use this method when upgrading from 5.0 to 5.1.
Supported Upgrade Paths

Unless otherwise documented, the following upgrade paths are supported:

- Upgrading from a release series version to a newer release series version is supported. For example, upgrading from 5.1.71 to 5.1.72 is supported. Skipping release series versions is also supported. For example, upgrading from 5.1.70 to 5.1.72 is supported.

- Upgrading one release level is supported. For example, upgrading from 5.0 to 5.1 is supported. Upgrading to the latest release series version is recommended before upgrading to the next release level. For example, upgrade to the latest 5.0 release before upgrading to 5.1.

- Upgrading more than one release level is supported, but only if you upgrade one release level at a time. For example, if you currently are running MySQL 4.1 and wish to upgrade to a newer series, upgrade to MySQL 5.0 first before upgrading to MySQL 5.1, and so forth. For information on upgrading to MySQL 5.0 see the MySQL 5.0 Reference Manual.

- Direct upgrades that skip a release level (for example, upgrading directly from MySQL 4.1 to 5.1) are not recommended or supported.

The following conditions apply to all upgrade paths:

- Upgrades between General Availability (GA) status releases are supported.

- Upgrades between milestone releases (or from a milestone release to a GA release) are not supported.

- For upgrades between versions of a MySQL release series that has reached GA status, you can move the MySQL format files and data files between different versions on systems with the same architecture. This is not necessarily true for upgrades between milestone releases. Use of milestone releases is at your own risk.

Before You Begin

Before upgrading, review the following information and perform the recommended steps:

- Before upgrading, protect your data by creating a backup of your current databases and log files. The backup should include the mysql database, which contains the MySQL system tables. See Database Backup Methods.

- Review the Release Notes which provide information about features that are new in the MySQL 5.1 or differ from those found in earlier MySQL releases. Some of these changes may result in incompatibilities.

- Review Section 11.1.1, “Changes Affecting Upgrades to 5.1”. This section describes changes that may require action before or after upgrading.

- Check Section 11.3, “Checking Whether Tables or Indexes Must Be Rebuilt”, to see whether changes to table formats or to character sets or collations were made between your current version of MySQL and the version to which you are upgrading. If such changes have resulted in an incompatibility between MySQL versions, you will need to upgrade the affected tables using the instructions in Section 11.4, “Rebuilding or Repairing Tables or Indexes”.
Performing an In-place Upgrade

This section describes how to perform an in-place upgrade. Review Before you Begin before proceeding.

Note

If you upgrade an installation originally produced by installing multiple RPM packages, upgrade all the packages, not just some. For example, if you previously installed the server and client RPMs, do not upgrade just the server RPM.

To perform an in-place upgrade:

1. Review the changes described in Section 11.1.1, “Changes Affecting Upgrades to 5.1” for steps to be performed before upgrading.

2. If you use InnoDB, configure MySQL to perform a slow shutdown by setting innodb_fast_shutdown to 0. For example:

   shell> bin/mysql -u root -p --execute="set global innodb_fast_shutdown=0"

   With a slow shutdown, InnoDB performs a full purge and change buffer merge before shutting down, which ensures that data files are fully prepared in case of file format differences between releases.

3. Shut down the old MySQL server. For example:

   shell> bin/mysqladmin -u root -p shutdown

4. Upgrade the MySQL binaries or packages in place (replace the old binaries with the new ones).

5. Start the MySQL 5.1 server, using the existing data directory. For example:

   shell> bin/mysqld_safe --user=mysql --datadir=/path/to/existing-datadir

6. Run mysql_upgrade. For example:

   shell> bin/mysql_upgrade -u root -p

   mysql_upgrade examines all tables in all databases for incompatibilities with the current version of MySQL. mysql_upgrade also upgrades the system tables so that you can take advantage of new privileges or capabilities.
Performing a Logical Upgrade

This section describes how to perform a logical upgrade. Review Before you Begin before proceeding.

To perform a logical upgrade:

1. Review the changes described in Section 11.1.1, “Changes Affecting Upgrades to 5.1” for steps to be performed before upgrading.

2. Export your existing data from the previous MySQL version:

   ```
   shell> mysqldump --add-drop-table --routines --events --add-drop-table
            -> --all-databases --force > data-for-upgrade.sql
   ```

   **Note**
   
   Use the `--routines` and `--events` options with `mysqldump` (as shown above) if your databases include stored programs. The `--all-databases` option includes all databases in the dump, including the `mysql` database that holds the system tables.

3. Shut down the old MySQL server. For example:

   ```
   shell> bin/mysqladmin -u root -p password shutdown
   ```

4. Install MySQL 5.1. For installation instructions, see Chapter 1, Installing and Upgrading MySQL.

5. Initialize a new data directory:

   ```
   shell> scripts/mysql_install_db --user=mysql --datadir=/path/to/5.1-datadir
   ```

6. Start the MySQL 5.1 server, using the new data directory. For example:

   ```
   shell> bin/mysqld_safe --user=mysql --datadir=/path/to/5.1-datadir
   ```

7. Load the previously created dump file into the new MySQL server. For example:

   ```
   shell> bin/mysql -u root -p password --execute="source data-for-upgrade.sql" --force
   ```

8. Run `mysql_upgrade`. For example:

   ```
   shell> bin/mysql_upgrade -u root -p password
   ```

   `mysql_upgrade` examines all tables in all databases for incompatibilities with the current version of MySQL. `mysql_upgrade` also upgrades the system tables so that you can take advantage of new privileges or capabilities.

   **Note**
   
   `mysql_upgrade` does not upgrade the contents of the help tables. For upgrade instructions, see Server-Side Help.
Upgrade Troubleshooting

9. If you use InnoDB, configure MySQL to perform a slow shutdown by setting `innodb_fast_shutdown` to 0. For example:

```
shell> bin/mysql -u root -p password --execute="set global innodb_fast_shutdown=0"
```

10. Shut down and restart the MySQL server to ensure a clean shutdown and startup. For example:

```
shell> bin/mysqld_safe --user=mysql --datadir=/path/to/5.1-datadir
```

Upgrade Troubleshooting

• If problems occur, such as that the new mysqld server does not start or that you cannot connect without a password, verify that you do not have an old my.cnf file from your previous installation. You can check this with the `--print-defaults` option (for example, `mysqld --print-defaults`). If this command displays anything other than the program name, you have an active my.cnf file that affects server or client operation.

• If, after an upgrade, you experience problems with compiled client programs, such as Commands out of sync or unexpected core dumps, you probably have used old header or library files when compiling your programs. In this case, check the date for your mysql.h file and libmysqlclient.a library to verify that they are from the new MySQL distribution. If not, recompile your programs with the new headers and libraries. Re compilation might also be necessary for programs compiled against the shared client library if the library major version number has changed (for example from libmysqlclient.so.15 to libmysqlclient.so.16).

• If you have created a user-defined function (UDF) with a given name and upgrade MySQL to a version that implements a new built-in function with the same name, the UDF becomes inaccessible. To correct this, use DROP FUNCTION to drop the UDF, and then use CREATE FUNCTION to re-create the UDF with a different nonconflicting name. The same is true if the new version of MySQL implements a built-in function with the same name as an existing stored function. See Function Name Parsing and Resolution, for the rules describing how the server interprets references to different kinds of functions.

11.1.1 Changes Affecting Upgrades to 5.1

Before upgrading to MySQL 5.1, review the changes described in this section to identify upgrade issues that apply to your current MySQL installation and applications.

Note

In addition to the changes outlined in this section, review the Release Notes and other important information outlined in Before You Begin.

Changes marked as either Known issue or Incompatible change are incompatibilities with earlier versions of MySQL, and may require your attention before you upgrade. Our aim is to avoid these changes, but occasionally they are necessary to correct problems that would be worse than an incompatibility between releases. If any upgrade issue applicable to your installation involves an incompatibility that requires special handling, follow the instructions given in the incompatibility description. Sometimes this involves dumping and reloading tables, or use of a statement such as CHECK TABLE or REPAIR TABLE.

For dump and reload instructions, see Section 11.4, “Rebuilding or Repairing Tables or Indexes”. Any procedure that involves REPAIR TABLE with the USE_FRM option must be done before upgrading. Use of this statement with a version of MySQL different from the one used to create the table (that is, using it after upgrading) may damage the table. See REPAIR TABLE Syntax.

• Configuration Changes
Changes Affecting Upgrades to 5.1

- **System Table Changes**
- **Server Changes**
- **SQL Changes**
- **C API Changes**

**Configuration Changes**

- Before MySQL 5.1.11, to build MySQL from source with SSL support enabled, you would invoke `configure` with either the `--with-openssl` or `--with-yassl` option. In MySQL 5.1.11, those options both have been replaced by the `--with-ssl` option. By default, `--with-ssl` causes the bundled yaSSL library to be used. To select OpenSSL instead, give the option as `--with-ssl=path`, where `path` is the directory where the OpenSSL header files and libraries are located.

**System Table Changes**

- **After upgrading a 5.0 installation to 5.0.10 or above**, it is necessary to upgrade your grant tables using `mysql_upgrade`. Otherwise, creating stored procedures and functions might not work. Running `mysql_upgrade` is part of the the in-place and logical upgrade procedures.

**Server Changes**

- As of MySQL 5.1.9, the `mysqld-max` server is included in binary distributions. There is no separate MySQL-Max distribution. As of MySQL 5.1.12, there is no `mysqld-max` server at all in binary distributions. They contain a server that includes the features previously included in `mysqld-max`.

- **Known issue:** As of MySQL 5.1.70, for new installations, the `url` columns in the `mysql` database help tables are now created as type `TEXT` to accommodate longer URLs. For upgrades, `mysql_upgrade` does not update the columns. Modify them manually using these statements:

  ```sql
  ALTER TABLE mysql.help_category MODIFY url TEXT NOT NULL;
  ALTER TABLE mysql.help_topic MODIFY url TEXT NOT NULL;
  ```

- **Known issue:** `mysql_upgrade` attempts to upgrade tables that are incompatible with the current version of MySQL. (It invokes `mysqlcheck` to check tables and, if necessary, repair them.) However this can fail for storage engines that do not support `REPAIR TABLE`, such as InnoDB, and leave tables in a nonupgradable state.

  To work around this problem, use `ALTER TABLE tbl_name ENGINE=InnoDB` to perform a “null” alter operation that rebuilds the table.

- **Known issue:** After a binary upgrade to MySQL 5.1 from a MySQL 5.0 installation that contains `ARCHIVE` tables:
  
  - Before MySQL 5.1.42, accessing those tables will cause the server to crash, even if you have run `mysql_upgrade` or `CHECK TABLE ... FOR UPGRADE`.
  
  - As of MySQL 5.1.42, the server will not open 5.0 `ARCHIVE` tables at all.

  In either case, the solution is to use `mysqldump` to dump all 5.0 `ARCHIVE` tables before upgrading, and reload them into MySQL 5.1 after upgrading. This problem is fixed in MySQL 5.6.4: The server can open `ARCHIVE` tables created in MySQL 5.0. However, it remains the recommended upgrade procedure to dump 5.0 `ARCHIVE` tables before upgrading and reload them after upgrading.

- **Known issue:** The fix for Bug #23491 introduced a problem with `SHOW CREATE VIEW`, which is used by `mysqldump`. This causes an incompatibility when upgrading from versions affected by that bug fix.
Changes Affecting Upgrades to 5.1

(MySQL 5.0.40 through 5.0.43, MySQL 5.1.18 through 5.1.19): If you use `mysqldump` before upgrading from an affected version and reload the data after upgrading to a higher version, you must drop and recreate your views.

- **Known issue:** Dumps performed by using `mysqldump` to generate a dump file before the upgrade and reloading the file after upgrading are subject to the following problem:

Before MySQL 5.0.40, `mysqldump` displays SPATIAL index definitions using prefix lengths for the indexed columns. These prefix lengths are accepted in MySQL 5.0, but not as of MySQL 5.1. If you use `mysqldump` from versions of MySQL older than 5.0.40, any table containing SPATIAL indexes will cause an error when the dump file is reloaded into MySQL 5.1 or higher.

For example, a table definition might look like this when dumped in MySQL 5.0:

```sql
CREATE TABLE `t` (
  `g` geometry NOT NULL,
  SPATIAL KEY `g` (`g`(32))
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

The SPATIAL index definition will not be accepted in MySQL 5.1. To work around this, edit the dump file to remove the prefix:

```sql
CREATE TABLE `t` (
  `g` geometry NOT NULL,
  SPATIAL KEY `g` (`g`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Dump files can be large, so it may be preferable to dump table definitions and data separately to make it easier to edit the definitions:

```
shell> mysqldump --no-data other_args > definitions.sql
shell> mysqldump --no-create-info other_args > data.sql
```

Then edit `definitions.sql` before reloading `definitions.sql` and `data.sql`, in that order.

If you upgrade to a version of MySQL 5.0 higher than 5.0.40 before upgrading to MySQL 5.1, this problem does not occur.

- **Known issue:** Before MySQL 5.1.30, the `CHECK TABLE ... FOR UPGRADE` statement did not check for incompatible collation changes made in MySQL 5.1.24. (This also affects `mysqlcheck` and `mysql_upgrade`, which cause that statement to be executed.)

Prior to the fix made in 5.1.30, a binary upgrade (performed without dumping tables with `mysqldump` before the upgrade and reloading the dump file after the upgrade) would corrupt tables. After the fix, `CHECK TABLE ... FOR UPGRADE` properly detects the problem and warns about tables that need repair.

However, the fix is not backward compatible and can result in a downgrading problem under these circumstances:

1. Perform a binary upgrade to a version of MySQL that includes the fix.
2. Run `CHECK TABLE ... FOR UPGRADE` (or `mysqlcheck` or `mysql_upgrade`) to upgrade tables.
3. Perform a binary downgrade to a version of MySQL that does not include the fix.
Changes Affecting Upgrades to 5.1

The solution is to dump tables with `mysqldump` before the downgrade and reload the dump file after the downgrade. Alternatively, drop and recreate affected indexes.

- **Known issue**: MySQL introduces encoding for table names that have non-ASCII characters (see [Mapping of Identifiers to File Names](#)). After a binary upgrade from MySQL 5.0 to 5.1 or higher, the server recognizes names that have non-ASCII characters and adds a `mysql50` prefix to them.

As of MySQL 5.1.31, `mysql_upgrade` encodes these names by executing the following command:

```
mysqlcheck --all-databases --check-upgrade --fix-db-names --fix-table-names
```

Prior to MySQL 5.1.31, `mysql_upgrade` does not execute this command, so you should execute it manually if you have database or table names that contain nonalphanumeric characters.

Prior to MySQL 5.1.23, the `mysqlcheck` command does not perform the name encoding for views. To work around this problem, drop each affected view and recreate it.

`mysqlcheck` cannot fix names that contain literal instances of the `@` character that is used for encoding special characters. If you have databases or tables that contain this character, use `mysqldump` to dump them before upgrading to MySQL 5.1, and then reload the dump file after upgrading.

- **Known issue**: When upgrading from MySQL 5.0 to versions of 5.1 prior to 5.1.23, running `mysqlcheck` (or `mysql_upgrade`, which runs `mysqlcheck`) to upgrade tables fails for names that must be written as quoted identifiers. To work around this problem, rename each affected table to a name that does not require quoting:

```sql
RENAME TABLE `tab`\'le_a`' TO table_a;
RENAME TABLE `table b` TO table_b;
```

After renaming the tables, run the `mysql_upgrade` program. Then rename the tables back to their original names:

```sql
RENAME TABLE table_a TO `tab`\'le_a`; 
RENAME TABLE table_b TO `table b`;
```

- **Known issue**: In connection with view creation, the server created `arc` directories inside database directories and maintained useless copies of `.frm` files there. Creation and renaming procedures of those copies as well as creation of `arc` directories has been discontinued in MySQL 5.1.29.

This change does cause a problem when downgrading to older server versions which manifests itself under these circumstances:

1. Create a view `v_orig` in MySQL 5.1.29 or higher.
2. Rename the view to `v_new` and then back to `v_orig`.
3. Downgrade to an older 5.1.x server and run `mysql_upgrade`.
4. Try to rename `v_orig` to `v_new` again. This operation fails.

As a workaround to avoid this problem, use either of these approaches:

- Dump your data using `mysqldump` before downgrading and reload the dump file after downgrading.
- Instead of renaming a view after the downgrade, drop it and recreate it.
• **Incompatible change**: Character set or collation changes may require table indexes to be rebuilt. In MySQL 5.1, these occurred in versions 5.1.21, 5.1.23, and 5.1.24. For details, see Section 11.3, “Checking Whether Tables or Indexes Must Be Rebuilt”.

• **Incompatible change**: Prior to MySQL 5.1.51, if you flushed the logs using `FLUSH_logs` or `mysqladmin flush-logs` and `mysqld` was writing the error log to a file (for example, if it was started with the `--log-error` option), it renamed the current log file with the suffix `-old`, then created a new empty log file. This had the problem that a second log-flushing operation thus caused the original error log file to be lost unless you saved it under a different name. For example, you could use the following commands to save the file:

```
shell> mysqladmin flush-logs
shell> mv host_name.err-old backup-directory
```

To avoid the preceding file-loss problem, no renaming occurs as of MySQL 5.1.51; the server merely closes and reopens the log file. To rename the file, you can do so manually before flushing. Then flushing the logs reopens a new file with the original file name. For example, you can rename the file and create a new one using the following commands:

```
shell> mv host_name.err host_name.err-old
shell> mysqladmin flush-logs
shell> mv host_name.err-old backup-directory
```

• **Incompatible change**: MySQL 5.1 implements support for a plugin API that enables the loading and unloading of components at runtime, without restarting the server. The MySQL Plugin API. The plugin API requires the `mysql.plugin` table. After upgrading from an older version of MySQL, you should run the `mysql_upgrade` command to create this table. See `mysql_upgrade` — Check and Upgrade MySQL Tables.

Plugins are installed in the directory named by the `plugin_dir` system variable. This variable also controls the location from which the server loads user-defined functions (UDFs), which is a change from earlier versions of MySQL. That is, all UDF library files now must be installed in the plugin directory. When upgrading from an older version of MySQL, you must migrate your UDF files to the plugin directory.

• **Incompatible change**: The `table_cache` system variable has been renamed to `table_open_cache`. Any scripts that refer to `table_cache` must be updated to use the new name.

• **Incompatible change**: In MySQL 5.1.36, options for loading plugins such as pluggable storage engines were changed from boolean to tristate format. The implementations overlap, but if you previously used options of the form `--plugin_name=0` or `--plugin_name=1`, you should instead use `--plugin_name=OFF` or `--plugin_name=ON`, respectively. For details, see Installing and Uninstalling Plugins.

• **Incompatible change**: From MySQL 5.1.24 to 5.1.31, the `UPDATE` statement was changed such that assigning `NULL` to a `NOT NULL` column caused an error even when strict SQL mode was not enabled. The original behavior before MySQL 5.1.24 was that such assignments caused an error only in strict SQL mode, and otherwise set the column to the implicit default value for the column data type and generated a warning. (For information about implicit default values, see Data Type Default Values.)

The change caused compatibility problems for applications that relied on the original behavior. It also caused replication problems between servers that had the original behavior and those that did not, for applications that assigned `NULL` to `NOT NULL` columns in `UPDATE` statements without strict SQL mode enabled. The change was reverted in MySQL 5.1.32 so that `UPDATE` again had the original behavior.
Problems can still occur if you replicate between servers that have the modified UPDATE behavior and those that do not.

- **Incompatible change:** As of MySQL 5.1.29, the default binary logging mode has been changed from MIXED to STATEMENT for compatibility with MySQL 5.0.

- **Incompatible change:** In MySQL 5.1.25, a change was made to the way that the server handles prepared statements. This affects prepared statements processed at the SQL level (using the PREPARE statement) and those processed using the binary client/server protocol (using the mysql_stmt_prepare() C API function).

Previously, changes to metadata of tables or views referred to in a prepared statement could cause a server crash when the statement was next executed, or perhaps an error at execute time with a crash occurring later. For example, this could happen after dropping a table and recreating it with a different definition.

Now metadata changes to tables or views referred to by prepared statements are detected and cause automatic repreparation of the statement when it is next executed. Metadata changes occur for DDL statements such as those that create, drop, alter, rename, or truncate tables, or that analyze, optimize, or repair tables. Repreparation also occurs after referenced tables or views are flushed from the table definition cache, either implicitly to make room for new entries in the cache, or explicitly due to FLUSH TABLES.

Repreparation is automatic, but to the extent that it occurs, performance of prepared statements is diminished.

Table content changes (for example, with INSERT or UPDATE) do not cause repreparation, nor do SELECT statements.

An incompatibility with previous versions of MySQL is that a prepared statement may now return a different set of columns or different column types from one execution to the next. For example, if the prepared statement is `SELECT * FROM t1`, altering `t1` to contain a different number of columns causes the next execution to return a number of columns different from the previous execution.

Older versions of the client library cannot handle this change in behavior. For applications that use prepared statements with the new server, an upgrade to the new client library is strongly recommended.

Along with this change to statement repreparation, the default value of the table_definition_cache system variable has been increased from 128 to 256. The purpose of this increase is to lessen the chance that prepared statements will need repreparation due to referred-to tables/views having been flushed from the cache to make room for new entries.

A new status variable, Com_stmt_reprepare, has been introduced to track the number of repreparations.

- **Incompatible change:** The -, *, and / operators and the functions POW() and EXP() could misbehave when used with floating-point numbers. Previously they might return +INF, -INF, or NaN in cases of numeric overflow (including that caused by division by zero) or when invalid arguments were used. As of MySQL 5.1.24, NULL is returned in all such cases.

- **Incompatible change:** As of MySQL 5.1.23, within a stored routine, it is no longer permissible to declare a cursor for a SHOW or DESCRIBE statement. This happened to work in some instances, but is no longer supported. In many cases, a workaround for this change is to use the cursor with a SELECT query to read from an INFORMATION_SCHEMA table that produces the same information as the SHOW statement.

- **Incompatible change:** SHOW CREATE VIEW displays view definitions using an AS alias_name clause for each column. If a column is created from an expression, the default alias is the expression text, which
Changes Affecting Upgrades to 5.1

can be quite long. As of MySQL 5.1.23, aliases for column names in CREATE VIEW statements are checked against the maximum column length of 64 characters (not the maximum alias length of 256 characters). As a result, views created from the output of SHOW CREATE VIEW fail if any column alias exceeds 64 characters. This can cause problems for replication or loading dump files. For additional information and workarounds, see Restrictions on Views.

• Incompatible change: Several issues were identified for stored programs (stored procedures and functions, triggers, and events) and views containing non-ASCII symbols. These issues involved conversion errors due to incomplete character set information when translating these objects to and from stored format.

To address these problems, the representation for these objects was changed in MySQL 5.1.21. However, the fixes affect all stored programs and views. (For example, you will see warnings about “no creation context.”) To avoid warnings from the server about the use of old definitions from any release prior to 5.1.21, you should dump stored programs and views with mysqldump after upgrading to 5.1.21 or higher, and then reload them to recreate them with new definitions. Invoke mysqldump with a --default-character-set option that names the non-ASCII character set that was used for the definitions when the objects were originally created, and the --routines, --events, and --triggers options to dump stored program definitions.

Upgrading for triggers in particular must be handled carefully, for two reasons:

• The output from mysqldump does not contain a DROP TRIGGER statement preceding each CREATE TRIGGER statement, so reloading the dump file will fail to re-create the triggers unless you manually drop them after generating the dump file and before reloading it.

• If you are upgrading from a very old version of MySQL 5.0 (before 5.0.10), the trigger upgrade procedure is different because triggers for those versions were created using a different namespace (trigger names had to be unique per table, rather than per schema as is true now).

Assuming that you are upgrading from MySQL 5.0.10 to 5.1.20 to MySQL 5.1.21 or later, use the following procedure to upgrade your triggers:

• Use mysqldump to generate a dump file that contains the trigger definitions:

```
mysqldump --triggers --no-create-db --no-data
   --no-create-info --all-databases > triggers.sql
```

You might need to add options to specify connection parameters, such as --user or --password. Also, if you are updating from a version of MySQL 5.1 older than 5.1.21, you may need to include a --default-character-set option that specifies the non-ASCII character set that was used for the definitions when the triggers were originally created.

Otherwise, invoke mysqldump with exactly the preceding options to avoid generating a dump file that will not have the intended effect when reloaded. For example, if you omit the --no-create-db option, your databases will be removed and recreated with no contents when you reload the dump file.

• Drop existing triggers. To see which triggers exist, use this statement:

```
SELECT TRIGGER_SCHEMA, EVENT_OBJECT_TABLE, TRIGGER_NAME
FROM INFORMATION_SCHEMA.TRIGGERS;
```

To generate DROP TRIGGERS statements for the triggers, use this statement:

```
SELECT CONCAT('DROP TRIGGER ', TRIGGER_SCHEMA, '.', TRIGGER_NAME, ';')
```
FROM INFORMATION_SCHEMA.TRIGGERS
INTO OUTFILE '/tmp/drop_triggers.sql';

The statement uses INTO OUTFILE, so you must have the FILE privilege. The file will be created on
the server host. Use a different file name if you like. To be 100% safe, inspect the trigger definitions in
the drop_triggers.sql file, and perhaps make a backup of the file. Then execute the statements
in the file:

mysql --force < /tmp/drop_triggers.sql

• Recreate the triggers by reloading the dump file created earlier:

mysql --force < triggers.sql

For upgrades from a version of MySQL older than 5.0.10 to MySQL 5.1.21 or later, use the following
procedure to upgrade your triggers:

1. Upgrade to MySQL 5.1 to be able to access trigger information in the
   INFORMATION_SCHEMA.TRIGGERS table. (This should work even for pre-5.0.10 triggers.)

2. Dump all trigger definitions using the following SELECT statement:

```
SELECT CONCAT('CREATE TRIGGER ', t.TRIGGER_SCHEMA, '.', t.TRIGGER_NAME,
   ' ', t.ACTION_TIMING, ' ', t.EVENT_MANIPULATION, ' ON ',
   t.EVENT_OBJECT_SCHEMA, '.', t.EVENT_OBJECT_TABLE,
   ' FOR EACH ROW ', t.ACTION_STATEMENT, '//')
INTO OUTFILE '/tmp/triggers.sql'
FROM INFORMATION_SCHEMA.TRIGGERS AS t;
```

The statement uses INTO OUTFILE, so you must have the FILE privilege. The file will be created
on the server host. Use a different file name if you like. To be 100% safe, inspect the trigger
definitions in the triggers.sql file, and perhaps make a backup of the file.

3. Stop the server and drop all triggers by removing all .TRG files in your database directories. Change
   location to your data directory and issue this command:

```
shell> rm */*.TRG
```

4. Start the server and re-create all triggers using the triggers.sql file:

```
mysql> delimiter // ;
mysql> source /tmp/triggers.sql //
```

5. Use the SHOW TRIGGERS statement to check that all triggers were created successfully.

   • Incompatible change: As of MySQL 5.1.20, mysql_safe supports error logging to syslog on
     systems that support the logger command. The new --syslog and --skip-syslog options can be
     used instead of the --log-error option to control logging behavior, as described in mysql_safe —
     MySQL Server Startup Script.

   In 5.1.21 and up, the default is --skip-syslog, which is compatible with the default behavior of writing
   an error log file for releases prior to 5.1.20.
In 5.1.20 only, the following conditions apply: 1) The default is to use syslog, which is not compatible with releases prior to 5.1.20. 2) Logging to syslog may fail to operate correctly in some cases. For these reasons, avoid using MySQL 5.1.20.

• Incompatible change: As of MySQL 5.1.18, the plugin interface and its handling of system variables was changed. Command-line options such as --skip-innodb now cause an error if InnoDB is not built-in or plugin-loaded. You should use --loose-skip-innodb if you do not want any error even if InnoDB is not available. The --loose prefix modifier should be used for all command-line options where you are uncertain whether the plugin exists and when you want the operation to proceed even if the option is necessarily ignored due to the absence of the plugin. (For a description of how --loose works, see Using Options on the Command Line.)

• Incompatible change: As of MySQL 5.1.15, InnoDB rolls back only the last statement on a transaction timeout. A new option, --innodb_rollback_on_timeout, causes InnoDB to abort and roll back the entire transaction if a transaction timeout occurs (the same behavior as in MySQL 4.1).

• Incompatible change: As of MySQL 5.1.15, the following conditions apply to enabling the read_only system variable:
  • If you attempt to enable read_only while you have any explicit locks (acquired with LOCK TABLES or have a pending transaction, an error will occur.
  • If other clients hold explicit table locks or have pending transactions, the attempt to enable read_only blocks until the locks are released and the transactions end. While the attempt to enable read_only is pending, requests by other clients for table locks or to begin transactions also block until read_only has been set.
  • read_only can be enabled while you hold a global read lock (acquired with FLUSH TABLES WITH READ LOCK) because that does not involve table locks.

Previously, the attempt to enable read_only would return immediately even if explicit locks or transactions were pending, so some data changes could occur for statements executing in the server at the same time.

• Incompatible change: The number of function names affected by IGNORE_SPACE was reduced significantly in MySQL 5.1.13, from about 200 to about 30. (For details about IGNORE_SPACE, see Function Name Parsing and Resolution.) This change improves the consistency of parser operation. However, it also introduces the possibility of incompatibility for old SQL code that relies on the following conditions:
  • IGNORE_SPACE is disabled.
  • The presence or absence of whitespace following a function name is used to distinguish between a built-in function and stored function that have the same name (for example, PI() versus PI ()).

For functions that are no longer affected by IGNORE_SPACE as of MySQL 5.1.13, that strategy no longer works. Either of the following approaches can be used if you have code that is subject to the preceding incompatibility:
  • If a stored function has a name that conflicts with a built-in function, refer to the stored function with a schema name qualifier, regardless of whether whitespace is present. For example, write schema_name.PI() or schema_name.PI ()..
  • Alternatively, rename the stored function to use a nonconflicting name and change invocations of the function to use the new name.
Changes Affecting Upgrades to 5.1

- **Incompatible change**: For utf8 columns, the full-text parser incorrectly considered several nonword punctuation and whitespace characters as word characters, causing some searches to return incorrect results. The fix involves a change to the full-text parser in MySQL 5.1.12, so as of 5.1.12, any tables that have FULLTEXT indexes on utf8 columns must be repaired with REPAIR TABLE:

  ```sql
  REPAIR TABLE tbl_name QUICK;
  ```

- **Incompatible change**: Storage engines can be pluggable at runtime, so the distinction between disabled and invalid storage engines no longer applies. As of MySQL 5.1.12, this affects the NO_ENGINE_SUBSTITUTION SQL mode, as described in Server SQL Modes.

- **Incompatible change**: The structure of FULLTEXT indexes has been changed in MySQL 5.1.6. After upgrading to MySQL 5.1.6 or greater, any tables that have FULLTEXT indexes must be repaired with REPAIR TABLE:

  ```sql
  REPAIR TABLE tbl_name QUICK;
  ```

- **Incompatible change**: In MySQL 5.1.6, when log tables were implemented, the default log destination for the general query and slow query log was TABLE. As of MySQL 5.1.21, this default has been changed to FILE, which is compatible with MySQL 5.0, but incompatible with earlier releases of MySQL 5.1. If you are upgrading from MySQL 5.0 to 5.1.21 or higher, no logging option changes should be necessary. However, if you are upgrading from 5.1.6 through 5.1.20 to 5.1.21 or higher and were using TABLE logging, use the `--log-output=TABLE` option explicitly to preserve your server's table-logging behavior.

- **Incompatible change**: In very old versions of MySQL (prior to 4.1), the TIMESTAMP data type supported a display width, which was silently ignored beginning with MySQL 4.1. This is deprecated in MySQL 5.1, and removed altogether in MySQL 5.5. These changes in behavior can lead to two problem scenarios when trying to use TIMESTAMP(N) columns with a MySQL 5.5 or later server:

  - When importing a dump file (for example, one created using `mysqldump`) created in a MySQL 5.0 or earlier server into a server from a newer release series, a `CREATE TABLE` or `ALTER TABLE` statement containing `TIMESTAMP(N)` causes the import to fail with a syntax error. To fix this problem, edit the dump file in a text editor to replace any instances of `TIMESTAMP(N)` with `TIMESTAMP` prior to importing the file. Be sure to use a plain text editor for this, and not a word processor; otherwise, the result is almost certain to be unusable for importing into the MySQL server.

  You should try to handle potential issues of these types proactively by updating with `ALTER TABLE` any `TIMESTAMP(N)` columns in your databases so that they use `TIMESTAMP` instead, before performing any upgrades.

- **Incompatible change**: For ENUM columns that had enumeration values containing commas, the commas were mapped to 0xff internally. However, this rendered the commas indistinguishable from true 0xff characters in the values. This no longer occurs. However, the fix requires that you dump and reload any tables that have ENUM columns containing true 0xff in their values: Dump the tables using `mysqldump` with the current server before upgrading from a version of MySQL 5.1 older than 5.1.15 to version 5.1.15 or newer.

- As of MySQL 5.1.12, the lc_time_names system variable specifies the locale that controls the language used to display day and month names and abbreviations. This variable affects the output from the `DATE_FORMAT()`, `DAYNAME()` and `MONTHNAME()` functions. See MySQL Server Locale Support.

- As of MySQL 5.1.9, `mysqld_safe` no longer implicitly invokes `mysqld-max` if it exists. Instead, it invokes `mysqld` unless a `--mysqld` or `--mysqld-version` option is given to specify another
server explicitly. If you previously relied on the implicit invocation of `mysqld-max`, you should use an appropriate option now. As of MySQL 5.1.12, there is no longer any separate `mysqld-max` server, so no change should be necessary.

**SQL Changes**

- **Known issue:** Prior to MySQL 5.1.17, the parser accepted invalid code in SQL condition handlers, leading to server crashes or unexpected execution behavior in stored programs. Specifically, the parser permitted a condition handler to refer to labels for blocks that enclose the handler declaration. This was incorrect because block label scope does not include the code for handlers declared within the labeled block.

  As of 5.1.17, the parser rejects this invalid construct, but if you perform a binary upgrade (without dumping and reloading your databases), existing handlers that contain the construct still are invalid and should be rewritten even if they appear to function as you expect.

  To find affected handlers, use `mysqldump` to dump all stored procedures and functions, triggers, and events. Then attempt to reload them into an upgraded server. Handlers that contain illegal label references will be rejected.

  For more information about condition handlers and writing them to avoid invalid jumps, see DECLARE ... HANDLER Syntax.

- **Incompatible change:** The parser accepted statements that contained `/* ... */` that were not properly closed with `*/`, such as `SELECT 1 /* + 2`. As of MySQL 5.1.23, statements that contain unclosed `/*-comments now are rejected with a syntax error.

  This fix has the potential to cause incompatibilities. Because of Bug #26302, which caused the trailing `*/` to be truncated from comments in views, stored routines, triggers, and events, it is possible that objects of those types may have been stored with definitions that now will be rejected as syntactically invalid. Such objects should be dropped and re-created so that their definitions do not contain truncated comments.

- **Incompatible change:** Multiple-table `DELETE` statements containing ambiguous aliases could have unintended side effects such as deleting rows from the wrong table. Examples:

  ```
  DELETE FROM t1 AS a2 USING t1 AS a1 INNER JOIN t2 AS a2;
  DELETE t1 AS a2 FROM t1 AS a1 INNER JOIN t2 AS a2;
  ```

  To avoid ambiguity, declaration of aliases other than in the `table_references` part of the statement should be avoided:

  ```
  DELETE FROM t1 USING t1 AS a1 INNER JOIN t2 AS a2;
  DELETE t1 FROM t1 AS a1 INNER JOIN t2 AS a2;
  ```

  As of MySQL 5.1.23, alias declarations outside the `table_references` part of the statement are disallowed for the `USING` variant of multiple-table `DELETE` syntax. (In MySQL 5.5, alias declarations outside `table_references` are disallowed for all multiple-table `DELETE` statements.) Statements containing aliases that are no longer permitted must be rewritten.

- **Incompatible change:** As of MySQL 5.1.8, `TYPE = engine_name` is still accepted as a synonym for the `ENGINE = engine_name` table option but generates a warning. You should note that this option is not available in MySQL 5.1.7, and is removed altogether in MySQL 5.5 and produces a syntax error. `TYPE` has been deprecated since MySQL 4.0.
• **Incompatible change**: MySQL 5.1.6 introduces the `TRIGGER` privilege. Previously, the `SUPER` privilege was needed to create or drop triggers. Now those operations require the `TRIGGER` privilege. This is a security improvement because you no longer need to grant users the `SUPER` privilege to enable them to create triggers. However, the requirement that the account named in a trigger’s `DEFINER` clause must have the `SUPER` privilege has changed to a requirement for the `TRIGGER` privilege. When upgrading from a previous version of MySQL 5.0 or 5.1 to MySQL 5.1.6 or newer, be sure to update your grant tables by running `mysql_upgrade`. This will assign the `TRIGGER` privilege to all accounts that had the `SUPER` privilege. If you fail to update the grant tables, triggers may fail when activated. After updating the grant tables, you can revoke the `SUPER` privilege from those accounts that no longer otherwise require it.

• Some keywords may be reserved in MySQL 5.1 that were not reserved in MySQL 5.0. See Keywords and Reserved Words.

• The `BACKUP TABLE`, and `RESTORE TABLE` statements are deprecated. `mysqldump` or `mysqlhotcopy` can be used as alternatives.

• The `LOAD DATA FROM MASTER` and `LOAD TABLE FROM MASTER` statements are deprecated. See `LOAD DATA FROM MASTER Syntax`, for recommended alternatives.

• The `INSTALL PLUGIN` and `UNINSTALL PLUGIN` statements that are used for the plugin API are new. So is the `WITH PARSER` clause for `FULLTEXT` index creation that associates a parser plugin with a full-text index. The MySQL Plugin API.

### SC API Changes

• **Incompatible change**: As of MySQL 5.1.7, the `mysql_stmt_attr_get()` C API function returns a boolean rather than an unsigned int for `STMT_ATTR_UPDATE_MAX_LENGTH`. (Bug #16144)

### 11.2 Downgrading MySQL

This section describes how to downgrade to an older MySQL version.

• Supported Downgrade Methods

• Supported Downgrade Paths

• Before You Begin

• Performing an In-place Downgrade

• Performing a Logical Downgrade

• Downgrade Troubleshooting

#### Supported Downgrade Methods

Supported downgrade methods include:

• **In-place Downgrade**: Involves shutting down the new MySQL version, replacing the new MySQL binaries or packages with the old ones, and restarting the old MySQL version on the new data files. In-place downgrades are supported for downgrades between GA versions within the same release series. For example, in-place downgrades are supported for downgrades from 5.1.73 to 5.1.72.

• **Logical Downgrade**: Involves using `mysqldump` to dump all tables from the new MySQL version, and then loading the dump file into the old MySQL version. Logical downgrades are supported for downgrades between GA versions within the same release series and for downgrades between release
levels. For example, logical downgrades are supported for downgrades from 5.1.73 to 5.1.72 and for downgrades from 5.1 to 5.0.

**Supported Downgrade Paths**

Unless otherwise documented, the following downgrade paths are supported:

- Downgrading from a release series version to an older release series version is supported using all down **grade methods**. For example, downgrading from 5.1.73 to 5.1.72 is supported. Skipping release series versions is also supported. For example, downgrading from 5.1.73 to 5.1.71 is supported.

- Downgrading one release level is supported using the **logical downgrade** method. For example, downgrading from 5.1 to 5.0 is supported.

- Downgrading more than one release level is supported using the logical downgrade method, but only if you downgrade one release level at a time. For example, you can downgrade from 5.1 to 5.0, and then to 4.1.

The following conditions apply to all downgrade paths:

- Downgrades between General Availability (GA) status releases are supported.

- Downgrades between milestone releases (or from a GA release to a milestone release) are not supported.

**Before You Begin**

Before downgrading, the following steps are recommended:

- Review the Release Notes for the MySQL version you are downgrading from to ensure that there are no features or fixes that you really need.

- Review Section 11.2.1, “Changes Affecting Downgrades from MySQL 5.1”. This section describes changes that may require action before or after downgrading.

  **Note**

  The downgrade procedures described in the following sections assume you are downgrading with data files created or modified by the newer MySQL version. However, if you did not modify your data after upgrading, downgrading using backups taken before upgrading to the new MySQL version is recommended. Many of the changes described in Section 11.2.1, “Changes Affecting Downgrades from MySQL 5.1” that require action before or after downgrading are not applicable when downgrading using backups taken before upgrading to the new MySQL version.

- Always back up your current databases and log files before downgrading. The backup should include the mysql database, which contains the MySQL system tables. See Database Backup Methods.

- Use of new features, new configuration options, or new configuration option values that are not supported by a previous release may cause downgrade errors or failures. Before downgrading, it is recommended that you reverse changes resulting from the use of new features and remove configuration settings that are not supported by the release you are downgrading to.

- Check Section 11.3, “Checking Whether Tables or Indexes Must Be Rebuilt”, to see whether changes to table formats or to character sets or collations were made between your current version of MySQL and the version to which you are downgrading. If such changes have resulted in an incompatibility between
Performing an In-place Downgrade

In-place downgrades are supported for downgrades between GA status releases within the same release series. Review Before you Begin before proceeding.

To perform an in-place downgrade:

1. Review the changes described in Section 11.2.1, “Changes Affecting Downgrades from MySQL 5.1” for steps to be performed before downgrading.

2. If you use InnoDB, configure MySQL to perform a slow shutdown by setting `innodb_fast_shutdown` to 0. For example:

```
shell> bin/mysql -u root -p password --execute="set global innodb_fast_shutdown=0"
```

   With a slow shutdown, InnoDB performs a full purge and change buffer merge before shutting down, which ensures that data files are fully prepared in case of file format differences between releases.

3. Shut down the newer MySQL server. For example:

```
shell> bin/mysqladmin -u root -p password shutdown
```

4. Downgrade the MySQL binaries or packages in-place by replacing the newer binaries or packages with the older ones.

5. Start the older (downgraded) MySQL server, using the existing data directory. For example:

```
shell> bin/mysqld_safe --user=mysql --datadir=/path/to/existing-datadir
```

6. Run `mysql_upgrade`. For example:

```
shell> bin/mysql_upgrade -u root -p password
```

Performing a Logical Downgrade

Logical downgrades are supported for downgrades between releases within the same release series and for downgrades to the previous release level. Only downgrades between General Availability (GA) status releases are supported. Review Before you Begin before proceeding.

To perform a logical downgrade:

1. Review the changes described in Section 11.2.1, “Changes Affecting Downgrades from MySQL 5.1” for steps to be performed before downgrading.

2. Dump all databases. For example:

```
shell> bin/mysqldump --add-drop-table --skip-events -u root -p password --all-databases --force > all_5_1_databases_dump.sql
```

3. Shut down the newer MySQL server. For example:

```
shell> bin/mysqladmin -u root -p password shutdown
```
4. Initialize an older MySQL instance, with a new data directory. For example:

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

5. Start the older MySQL server, using the new data directory. For example:

```shell
bin/mysqld_safe --user=mysql --datadir=/path/to/new-datadir
```

6. Load the dump file into the older MySQL server. For example:

```shell
bin/mysql -u root -p
password
--execute="source all_5_1_databases_dump.sql" --force
```

7. Run `mysql_upgrade`. For example:

```shell
bin/mysql_upgrade -u root -p
```

8. If you use InnoDB, configure MySQL to perform a slow shutdown by setting `innodb_fast_shutdown` to 0. For example:

```shell
bin/mysql -u root -p
password
--execute="set global innodb_fast_shutdown=0"
```

9. Shut down and restart the MySQL server to ensure a clean shutdown and startup. For example:

```shell
bin/mysqladmin -u root -p
password
shutdown
bin/mysqld_safe --user=mysql --datadir=/path/to/new-datadir
```

**Downgrade Troubleshooting**

If you downgrade from one release series to another, there may be incompatibilities in table storage formats. In this case, use `mysqldump` to dump your tables before downgrading. After downgrading, reload the dump file using `mysql` or `mysqlimport` to re-create your tables. For examples, see Section 11.5, “Copying MySQL Databases to Another Machine”.

A typical symptom of a downward-incompatible table format change when you downgrade is that you cannot open tables. In that case, use the following procedure:

1. Stop the older MySQL server that you are downgrading to.
2. Restart the newer MySQL server you are downgrading from.
3. Dump any tables that were inaccessible to the older server by using `mysqldump` to create a dump file.
4. Stop the newer MySQL server and restart the older one.
5. Reload the dump file into the older server. Your tables should be accessible.

**11.2.1 Changes Affecting Downgrades from MySQL 5.1**

Before downgrading from MySQL 5.1, review the changes described in this section. Some changes may require action before or after downgrading.

- **Partitioning**. MySQL 5.0 does not support user-defined partitioning. If a table was created as a partitioned table in 5.1 (or if an table created in a previous version of MySQL was altered to include partitions after an upgrade to 5.1), the table is accessible after downgrade only if you do one of the following:
  - Export the table using `mysqldump` and then drop it in MySQL 5.1; import the table again following the downgrade to MySQL 5.0.
• Prior to the downgrade, remove the table’s partitioning using `ALTER TABLE table_name REMOVE PARTITIONING`.

**Event Scheduler.** MySQL 5.0 does not support scheduled events. If your databases contain scheduled event definitions, you should prevent them from being dumped when you use `mysqldump` by using the `--skip-events` option. (See `mysqldump — A Database Backup Program`.)

**Stored routines.** MySQL 5.1.21 added a number of new columns to the `mysql.proc` table in which stored routine definitions are stored. If you are downgrading from MySQL 5.1.21 or later to MySQL 5.0, you cannot import the MySQL 5.1 routine definitions into MySQL 5.0.46 or earlier using the dump of `mysql.proc` created by `mysqldump` (such as when using the `--all-databases` option). Instead, you should run `mysqldump --routines` prior to performing the downgrade and run the stored routines DDL statements following the downgrade.

See Bug #11986, Bug #30029, and Bug #30660, for more information.

**Triggers.** Trigger creation requires the `TRIGGER` privilege as of MySQL 5.1. In MySQL 5.0, there is no `TRIGGER` privilege and `SUPER` is required instead. If you downgrade from MySQL 5.1 to 5.0, you will need to give the `SUPER` privilege to those accounts that had the `TRIGGER` privilege in 5.1.

### 11.2.2 Downgrading from MySQL Enterprise Edition to MySQL Community Server

This section describes the steps required to downgrade from MySQL Enterprise Edition to MySQL Community Edition. This can be done at any time, and is required at the expiration of a MySQL Enterprise Edition subscription if you wish to continue using MySQL Server.

When you perform such a downgrade, all commercially licensed components of the MySQL Enterprise Edition subscription must be uninstalled. These components and related considerations are described in the rest of this section.

**Note**

The issues described in this section are in addition to any that may be encountered as a result of any upgrade or downgrade of the MySQL Server version (such as between MySQL 5.1 and 5.0). Information about upgrading and downgrading between MySQL release series can be found elsewhere in this chapter; see Section 11.1, “Upgrading MySQL”, and Section 11.2, “Downgrading MySQL”.

**MySQL Enterprise Database Server.** All commercial versions of MySQL Database Server must be uninstalled.

**Commercially licensed extensions.** All commercially licensed MySQL Enterprise Database Server extensions must be uninstalled, including the MySQL High Availability extension. Following uninstallation of this extension, automated failover is no longer available.

**MySQL Enterprise Backup.** MySQL Enterprise Backup must be uninstalled. Uninstalling this application has the effects listed here:

• Automated backup scripts no longer work.

• Existing backup images taken with MySQL Enterprise Backup can no longer be used for recovery.

• Third-party integration with multimedia systems such as NetBackup, Tivoli, and Oracle Secure Backup no longer works.
MySQL Enterprise Monitor, MySQL Query Analyzer, agents. MySQL Enterprise Monitor, MySQL Query Analyzer, and all server-side agents must be uninstalled. Uninstalling these applications and agents has the following effects:

- Automated SNMP and SMTP alerts no longer work.
- All historical MySQL, OS monitoring, query, and performance metrics as well as all trending data are lost.
- All environment-specific monitoring templates, custom advisors, graphs and scripts are also lost.

### 11.3 Checking Whether Tables or Indexes Must Be Rebuilt

A binary upgrade or downgrade is one that installs one version of MySQL "in place" over an existing version, without dumping and reloading tables:

1. Stop the server for the existing version if it is running.
2. Install a different version of MySQL. This is an upgrade if the new version is higher than the original version, a downgrade if the version is lower.
3. Start the server for the new version.

In many cases, the tables from the previous version of MySQL can be used without problem by the new version. However, sometimes changes occur that require tables or table indexes to be rebuilt, as described in this section. If you have tables that are affected by any of the issues described here, rebuild the tables or indexes as necessary using the instructions given in Section 11.4, “Rebuilding or Repairing Tables or Indexes”.

#### Table Incompatibilities

After a binary upgrade to MySQL 5.1 from a MySQL 5.0 installation that contains ARCHIVE tables:

- Before MySQL 5.1.42, accessing those tables will cause the server to crash, even if you have run `mysql_upgrade` or `CHECK TABLE ... FOR UPGRADE`.
- As of MySQL 5.1.42, the server will not open 5.0 ARCHIVE tables at all.

In either case, the solution is to use `mysqldump` to dump all 5.0 ARCHIVE tables before upgrading, and reload them into MySQL 5.1 after upgrading. This problem is fixed in MySQL 5.6.4: The server can open ARCHIVE tables created in MySQL 5.0. However, it remains the recommended upgrade procedure to dump 5.0 ARCHIVE tables before upgrading and reload them after upgrading.

#### Index Incompatibilities

If you perform a binary upgrade without dumping and reloading tables, you cannot upgrade directly from MySQL 4.1 to 5.1 or higher. This occurs due to an incompatible change in the MyISAM table index format in MySQL 5.0. Upgrade from MySQL 4.1 to 5.0 and repair all MyISAM tables. Then upgrade from MySQL 5.0 to 5.1 and check and repair your tables.

Modifications to the handling of character sets or collations might change the character sort order, which causes the ordering of entries in any index that uses an affected character set or collation to be incorrect. Such changes result in several possible problems:

- Comparison results that differ from previous results
- Inability to find some index values due to misordered index entries
• Misordered ORDER BY results

• Tables that CHECK TABLE reports as being in need of repair

The solution to these problems is to rebuild any indexes that use an affected character set or collation, either by dropping and re-creating the indexes, or by dumping and reloading the entire table. In some cases, it is possible to alter affected columns to use a different collation. For information about rebuilding indexes, see Section 11.4, “Rebuilding or Repairing Tables or Indexes”.

To check whether a table has indexes that must be rebuilt, consult the following list. It indicates which versions of MySQL introduced character set or collation changes that require indexes to be rebuilt. Each entry indicates the version in which the change occurred and the character sets or collations that the change affects. If the change is associated with a particular bug report, the bug number is given.

The list applies both for binary upgrades and downgrades. For example, Bug #27877 was fixed in MySQL 5.1.24, so it applies to upgrades from versions older than 5.1.24 to 5.1.24 or newer, and to downgrades from 5.1.24 or newer to versions older than 5.1.24.

In many cases, you can use CHECK TABLE ... FOR UPGRADE to identify tables for which index rebuilding is required. It will report this message:

```
Table upgrade required.
Please do "REPAIR TABLE `tbl_name`" or dump/reload to fix it!
```

In these cases, you can also use mysqlcheck --check-upgrade or mysql_upgrade, which execute CHECK TABLE. However, the use of CHECK TABLE applies only after upgrades, not downgrades. Also, CHECK TABLE is not applicable to all storage engines. For details about which storage engines CHECK TABLE supports, see CHECK TABLE Syntax.

These changes cause index rebuilding to be necessary:

• MySQL 5.1.24 (Bug #27877)

  Affects indexes that use the utf8_general_ci or ucs2_general_ci collation for columns that contain ‘ß’ LATIN SMALL LETTER SHARP S (German). The bug fix corrected an error in the original collations but introduced an incompatibility such that ‘ß’ compares equal to characters with which it previously compared different.

  Affected tables can be detected by CHECK TABLE ... FOR UPGRADE as of MySQL 5.1.30 (see Bug #40053).

  A workaround for this issue is implemented as of MySQL 5.1.62, 5.5.21, and 5.6.5. The workaround involves altering affected columns to use the utf8_general_mysql500_ci and ucs2_general_mysql500_ci collations, which preserve the original pre-5.1.24 ordering of utf8_general_ci and ucs2_general_ci.

• MySQL 5.0.48, 5.1.23 (Bug #27562)

  Affects indexes that use the ascii_general_ci collation for columns that contain any of these characters: ‘ ’ GRAVE ACCENT, ‘[‘ LEFT SQUARE BRACKET, ‘\’ REVERSE SOLIDUS, ‘]’ RIGHT SQUARE BRACKET, ‘~’ TILDE

  Affected tables can be detected by CHECK TABLE ... FOR UPGRADE as of MySQL 5.1.29 (see Bug #39585).

• MySQL 5.0.48, 5.1.21 (Bug #29461)
Rebuilding or Repairing Tables or Indexes

Affects indexes for columns that use any of these character sets: eucjpms, euc_kr, gb2312, latin7, macce, ujis

Affected tables can be detected by `CHECK TABLE ... FOR UPGRADE` as of MySQL 5.1.29 (see Bug #39585).

11.4 Rebuilding or Repairing Tables or Indexes

This section describes how to rebuild a table. This can be necessitated by changes to MySQL such as how data types are handled or changes to character set handling. For example, an error in a collation might have been corrected, necessitating a table rebuild to update the indexes for character columns that use the collation. (For examples, see Section 11.3, “Checking Whether Tables or Indexes Must Be Rebuilt”.) It might also be that a table repair or upgrade should be done as indicated by a table check operation such as that performed by `CHECK TABLE`, `mysqlcheck`, or `mysql_upgrade`.

Methods for rebuilding a table include dumping and reloading it, or using `ALTER TABLE` or `REPAIR TABLE`.

### Note

If you are rebuilding tables because a different version of MySQL will not handle them after a binary (in-place) upgrade or downgrade, you must use the dump-and-reload method. Dump the tables before upgrading or downgrading using your original version of MySQL. Then reload the tables after upgrading or downgrading.

If you use the dump-and-reload method of rebuilding tables only for the purpose of rebuilding indexes, you can perform the dump either before or after upgrading or downgrading. Reloading still must be done afterward.

To rebuild a table by dumping and reloading it, use `mysqldump` to create a dump file and `mysql` to reload the file:

```shell
shell> mysqldump db_name t1 > dump.sql
shell> mysql db_name < dump.sql
```

To rebuild all the tables in a single database, specify the database name without any following table name:

```shell
shell> mysqldump db_name > dump.sql
shell> mysql db_name < dump.sql
```

To rebuild all tables in all databases, use the `--all-databases` option:

```shell
shell> mysqldump --all-databases > dump.sql
shell> mysql < dump.sql
```

To rebuild a table with `ALTER TABLE`, use a “null” alteration; that is, an `ALTER TABLE` statement that “changes” the table to use the storage engine that it already has. For example, if `t1` is a `MyISAM` table, use this statement:

```shell
mysql> ALTER TABLE t1 ENGINE = MyISAM;
```

If you are not sure which storage engine to specify in the `ALTER TABLE` statement, use `SHOW CREATE TABLE` to display the table definition.
If you must rebuild a table because a table checking operation indicates that the table is corrupt or needs an upgrade, you can use `REPAIR TABLE` if that statement supports the table's storage engine. For example, to repair a `MyISAM` table, use this statement:

```
mysql> REPAIR TABLE t1;
```

For storage engines such as `InnoDB` that `REPAIR TABLE` does not support, use `mysqldump` to create a dump file and `mysql` to reload the file, as described earlier.

For specifics about which storage engines `REPAIR TABLE` supports, see `REPAIR TABLE Syntax`.

`mysqlcheck --repair` provides command-line access to the `REPAIR TABLE` statement. This can be a more convenient means of repairing tables because you can use the `--databases` or `--all-databases` option to repair all tables in specific databases or all databases, respectively:

```
shell> mysqlcheck --repair --databases db_name ...
shell> mysqlcheck --repair --all-databases
```

For incompatibilities introduced in MySQL 5.1.24 by the fix for Bug #27877 that corrected the `utf8_general_ci` and `ucs2_general_ci` collations, a workaround is implemented as of MySQL 5.1.62, 5.5.21, and 5.6.5. Upgrade to one of those versions, then convert each affected table using one of the following methods. In each case, the workaround altering affected columns to use the `utf8_general_mysql500_ci` and `ucs2_general_mysql500_ci` collations, which preserve the original pre-5.1.24 ordering of `utf8_general_ci` and `ucs2_general_ci`.

- To convert an affected table after a binary upgrade that leaves the table files in place, alter the table to use the new collation. Suppose that the table `t1` contains one or more problematic `utf8` columns. To convert the table at the table level, use a statement like this:

  ```
  ALTER TABLE t1 CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_mysql500_ci;
  ```

  To apply the change on a column-specific basis, use a statement like this (be sure to repeat the column definition as originally specified except for the `COLLATE` clause):

  ```
  ALTER TABLE t1 MODIFY c1 CHAR(N) CHARACTER SET utf8 COLLATE utf8_general_mysql500_ci;
  ```

- To upgrade the table using a dump and reload procedure, dump the table using `mysqldump`, modify the `CREATE TABLE` statement in the dump file to use the new collation, and reload the table.

After making the appropriate changes, `CHECK TABLE` should report no error.

### 11.5 Copying MySQL Databases to Another Machine

You can copy the `.frm`, `.MYI`, and `.MYD` files for `MyISAM` tables between different architectures that support the same floating-point format. (MySQL takes care of any byte-swapping issues.) See `The MyISAM Storage Engine`.

In cases where you need to transfer databases between different architectures, you can use `mysqldump` to create a file containing SQL statements. You can then transfer the file to the other machine and feed it as input to the `mysql` client.

Use `mysqldump --help` to see what options are available.
Copying MySQL Databases to Another Machine

The easiest (although not the fastest) way to move a database between two machines is to run the following commands on the machine on which the database is located:

```
shell> mysqladmin -h 'other_hostname' create db_name
shell> mysqldump db_name | mysql -h 'other_hostname' db_name
```

If you want to copy a database from a remote machine over a slow network, you can use these commands:

```
shell> mysqladmin create db_name
shell> mysqldump -h 'other_hostname' --compress db_name | mysql db_name
```

You can also store the dump in a file, transfer the file to the target machine, and then load the file into the database there. For example, you can dump a database to a compressed file on the source machine like this:

```
shell> mysqldump --quick db_name | gzip > db_name.gz
```

Transfer the file containing the database contents to the target machine and run these commands there:

```
shell> mysqladmin create db_name
shell> gunzip < db_name.gz | mysql db_name
```

You can also use `mysqldump` and `mysqlimport` to transfer the database. For large tables, this is much faster than simply using `mysqldump`. In the following commands, `DUMPDIR` represents the full path name of the directory you use to store the output from `mysqldump`.

First, create the directory for the output files and dump the database:

```
shell> mkdir DUMPDIR
shell> mysqldump --tab=DUMPDIR db_name
```

Then transfer the files in the `DUMPDIR` directory to some corresponding directory on the target machine and load the files into MySQL there:

```
shell> mysqladmin create db_name # create database
shell> cat DUMPDIR/*.sql | mysql db_name # create tables in database
shell> mysqlimport db_name DUMPDIR/*.txt # load data into tables
```

Do not forget to copy the `mysql` database because that is where the grant tables are stored. You might have to run commands as the MySQL `root` user on the new machine until you have the `mysql` database in place.

After you import the `mysql` database on the new machine, execute `mysqladmin flush-privileges` so that the server reloads the grant table information.
Chapter 12 Environment Variables

This section lists environment variables that are used directly or indirectly by MySQL. Most of these can also be found in other places in this manual.

Options on the command line take precedence over values specified in option files and environment variables, and values in option files take precedence over values in environment variables. In many cases, it is preferable to use an option file instead of environment variables to modify the behavior of MySQL. See Using Option Files.

To set an environment variable within sh, ksh, bash, or zsh shells you can use the following commands:

```
shell> ENVVAR=value
shell> export ENVVAR
```

The `export` is required if you want other sub-commands to inherit that value during execution. On more recent shells you can combine the export and setting processes, like this:

```
shell> export set ENVVAR=value
```

For the csh, or tcsh shells, use the `setenv` command:

```
shell> setenv ENVVAR value
```

Within the Windows Command Prompt use the `set` command to specify the value of a variable:

```
shell> set ENVVAR=value
```

On Unix, Linux, OS X, and Windows, the `set` command without parameters will display the list of currently set variables.

Table 12.1 MySQL Related Environment Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXX</td>
<td>The name of your C++ compiler (for running <code>configure</code>).</td>
</tr>
<tr>
<td>CC</td>
<td>The name of your C compiler (for running <code>configure</code>).</td>
</tr>
<tr>
<td>CFLAGS</td>
<td>Flags for your C compiler (for running <code>configure</code>).</td>
</tr>
<tr>
<td>CXXFLAGS</td>
<td>Flags for your C++ compiler (for running <code>configure</code>).</td>
</tr>
<tr>
<td>DBI_USER</td>
<td>The default user name for Perl DBI.</td>
</tr>
<tr>
<td>DBI_TRACE</td>
<td>Trace options for Perl DBI.</td>
</tr>
<tr>
<td>HOME</td>
<td>The default path for the <code>mysql</code> history file is <code>$HOME/.mysql_history</code>.</td>
</tr>
<tr>
<td>LD_RUN_PATH</td>
<td>Used to specify the location of <code>libmysqlclient.so</code>.</td>
</tr>
<tr>
<td>MYSQL_DEBUG</td>
<td>Debug trace options when debugging.</td>
</tr>
<tr>
<td>MYSQL_GROUP_SUFFIX</td>
<td>Option group suffix value (like specifying <code>--defaults-group-suffix</code>).</td>
</tr>
<tr>
<td>MYSQL_HISTFILE</td>
<td>The path to the <code>mysql</code> history file. If this variable is set, its value overrides the default for <code>$HOME/.mysql_history</code>.</td>
</tr>
<tr>
<td>MYSQL_HOME</td>
<td>The path to the directory in which the server-specific <code>my.cnf</code> file resides.</td>
</tr>
<tr>
<td>MYSQL_HOST</td>
<td>The default host name used by the <code>mysql</code> command-line client.</td>
</tr>
<tr>
<td>MYSQL_PS1</td>
<td>The command prompt to use in the <code>mysql</code> command-line client.</td>
</tr>
<tr>
<td>MYSQL_PWD</td>
<td>The default password when connecting to <code>mysql</code>. Using this is insecure. See End-User Guidelines for Password Security.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MYSQL_TCP_PORT</td>
<td>The default TCP/IP port number.</td>
</tr>
<tr>
<td>MYSQL_UNIX_PORT</td>
<td>The default Unix socket file name; used for connections to localhost.</td>
</tr>
<tr>
<td>PATH</td>
<td>Used by the shell to find MySQL programs.</td>
</tr>
<tr>
<td>TMPDIR</td>
<td>The directory in which temporary files are created.</td>
</tr>
<tr>
<td>TZ</td>
<td>This should be set to your local time zone. See Time Zone Problems.</td>
</tr>
<tr>
<td>UMASK</td>
<td>The user-file creation mode when creating files. See note following table.</td>
</tr>
<tr>
<td>UMASK_DIR</td>
<td>The user-directory creation mode when creating directories. See note</td>
</tr>
<tr>
<td></td>
<td>following table.</td>
</tr>
<tr>
<td>USER</td>
<td>The default user name on Windows and NetWare when connecting to mysql.</td>
</tr>
</tbody>
</table>

For information about the mysql history file, see mysql Logging.

The default UMASK and UMASK_DIR values are 0660 and 0700, respectively. MySQL assumes that the value for UMASK or UMASK_DIR is in octal if it starts with a zero. For example, setting UMASK=0600 is equivalent to UMASK=384 because 0600 octal is 384 decimal.

The UMASK and UMASK_DIR variables, despite their names, are used as modes, not masks:

- If UMASK is set, mysql uses ($UMASK | 0600) as the mode for file creation, so that newly created files have a mode in the range from 0600 to 0666 (all values octal).

- If UMASK_DIR is set, mysql uses ($UMASK_DIR | 0700) as the base mode for directory creation, which then is AND-ed with (~($UMASK & 0666)), so that newly created directories have a mode in the range from 0700 to 0777 (all values octal). The AND operation may remove read and write permissions from the directory mode, but not execute permissions.
Chapter 13 Perl Installation Notes

Table of Contents

13.1 Installing Perl on Unix ................................................................. 185
13.2 Installing ActiveState Perl on Windows ....................................... 186
13.3 Problems Using the Perl DBI/DBD Interface ................................... 187

The Perl DBI module provides a generic interface for database access. You can write a DBI script that works with many different database engines without change. To use DBI, you must install the DBI module, as well as a DataBase Driver (DBD) module for each type of database server you want to access. For MySQL, this driver is the DBD::mysql module.

Perl, and the DBD::MySQL module for DBI must be installed if you want to run the MySQL benchmark scripts; see The MySQL Benchmark Suite. They are also required for the MySQL Cluster ndb_size.pl utility; see ndb_size.pl — NDBCLUSTER Size Requirement Estimator.

Note
Perl support is not included with MySQL distributions. You can obtain the necessary modules from http://search.cpan.org for Unix, or by using the ActiveState ppm program on Windows. The following sections describe how to do this.

The DBI/DBD interface requires Perl 5.6.0, and 5.6.1 or later is preferred. DBI does not work if you have an older version of Perl. You should use DBD::mysql 4.009 or higher. Although earlier versions are available, they do not support the full functionality of MySQL 5.1.

13.1 Installing Perl on Unix

MySQL Perl support requires that you have installed MySQL client programming support (libraries and header files). Most installation methods install the necessary files. If you install MySQL from RPM files on Linux, be sure to install the developer RPM as well. The client programs are in the client RPM, but client programming support is in the developer RPM.

The files you need for Perl support can be obtained from the CPAN (Comprehensive Perl Archive Network) at http://search.cpan.org.

The easiest way to install Perl modules on Unix is to use the CPAN module. For example:

shell> perl -MCPAN -e shell
cpan> install DBI
cpan> install DBD::mysql

The DBD::mysql installation runs a number of tests. These tests attempt to connect to the local MySQL server using the default user name and password. (The default user name is your login name on Unix, and ODBC on Windows. The default password is "no password.") If you cannot connect to the server with those values (for example, if your account has a password), the tests fail. You can use force install DBD::mysql to ignore the failed tests.

DBI requires the Data::Dumper module. It may be installed; if not, you should install it before installing DBI.
It is also possible to download the module distributions in the form of compressed tar archives and build the modules manually. For example, to unpack and build a DBI distribution, use a procedure such as this:

1. Unpack the distribution into the current directory:

   ```shell
   gunzip < DBI-VERSION.tar.gz | tar xvf -
   ```

   This command creates a directory named `DBI-VERSION`.

2. Change location into the top-level directory of the unpacked distribution:

   ```shell
   cd DBI-VERSION
   ```

3. Build the distribution and compile everything:

   ```shell
   perl Makefile.PL
   make
   make test
   make install
   ```

   The `make test` command is important because it verifies that the module is working. Note that when you run that command during the `DBD::mysql` installation to exercise the interface code, the MySQL server must be running or the test fails.

   It is a good idea to rebuild and reinstall the `DBD::mysql` distribution whenever you install a new release of MySQL. This ensures that the latest versions of the MySQL client libraries are installed correctly.

   If you do not have access rights to install Perl modules in the system directory or if you want to install local Perl modules, the following reference may be useful: [http://learn.perl.org/faq/perlfaq8.html#How-do-I-keep-my-own-module-library-directory](http://learn.perl.org/faq/perlfaq8.html#How-do-I-keep-my-own-module-library-directory)

### 13.2 Installing ActiveState Perl on Windows

On Windows, you should do the following to install the MySQL DBD module with ActiveState Perl:

1. Get ActiveState Perl from [http://www.activestate.com/Products/ActivePerl/](http://www.activestate.com/Products/ActivePerl/) and install it.

2. Open a console window.

3. If necessary, set the `HTTP_proxy` variable. For example, you might try a setting like this:

   ```cmd
   set HTTP_proxy=my.proxy.com:3128
   ```

4. Start the PPM program:

   ```cmd
   C:\> ppm.pl
   ```

5. If you have not previously done so, install `DBI`:

   ```ppm
   install DBI
   ```

6. If this succeeds, run the following command:

   ```ppm
   install DBD-mysql
   ```
This procedure should work with ActiveState Perl 5.6 or newer.

If you cannot get the procedure to work, you should install the ODBC driver instead and connect to the MySQL server through ODBC:

```
use DBI;
$dbh= DBI->connect("DBI:ODBC:$dsn",$user,$password) ||
    die "Got error $DBI::errstr when connecting to $dsn\n";
```

### 13.3 Problems Using the Perl DBI/DBD Interface

If Perl reports that it cannot find the `../mysql/mysql.so` module, the problem is probably that Perl cannot locate the `libmysqlclient.so` shared library. You should be able to fix this problem by one of the following methods:

- Copy `libmysqlclient.so` to the directory where your other shared libraries are located (probably `/usr/lib` or `/lib`).
- Modify the `-L` options used to compile `DBD::mysql` to reflect the actual location of `libmysqlclient.so`.
- On Linux, you can add the path name of the directory where `libmysqlclient.so` is located to the `/etc/ld.so.conf` file.
- Add the path name of the directory where `libmysqlclient.so` is located to the `LD_RUN_PATH` environment variable. Some systems use `LD_LIBRARY_PATH` instead.

Note that you may also need to modify the `-L` options if there are other libraries that the linker fails to find. For example, if the linker cannot find `libc` because it is in `/lib` and the link command specifies `-L/usr/lib`, change the `-L` option to `-L/lib` or add `-L/lib` to the existing link command.

If you get the following errors from `DBD::mysql`, you are probably using `gcc` (or using an old binary compiled with `gcc`):

```
/usr/bin/perl: can't resolve symbol '__moddi3'
/usr/bin/perl: can't resolve symbol '__divdi3'
```

Add `-L/usr/lib/gcc-lib/... -lgcc` to the link command when the `mysql.so` library gets built (check the output from `make` for `mysql.so` when you compile the Perl client). The `-L` option should specify the path name of the directory where `libgcc.a` is located on your system.

Another cause of this problem may be that Perl and MySQL are not both compiled with `gcc`. In this case, you can solve the mismatch by compiling both with `gcc`.

You may see the following error from `DBD::mysql` when you run the tests:

```
t/00base............install_driver(mysql) failed:
Can't load '../blib/arch/auto/DBD/mysql/mysql.so' for module DBD::mysql: ...
```

This means that you need to include the `-lz` compression library on the link line. That can be done by changing the following line in the file `lib/DBD/mysql/Install.pm`:

```
$sysliblist .= " -lm";
```
Change that line to:

```
$sysliblist .= " -lm -lz";
```

After this, you must run `make realclean` and then proceed with the installation from the beginning.
Appendix A Licenses for Third-Party Components

Table of Contents

A.1 ANTLR 3 License ................................................................. 191
A.2 dtoa.c License .................................................................... 191
A.3 Editline Library (libedit) License ........................................ 192
A.4 FindGTest.cmake License ...................................................... 194
A.5 Fred Fish’s Debug Library License ........................................ 195
A.6 getarg License ................................................................. 196
A.7 GNU General Public License Version 2.0, June 1991 .................. 196
A.8 GNU Lesser General Public License Version 2.1, February 1999 ... 202
A.9 GNU Libtool License .......................................................... 210
A.10 GNU Readline License ....................................................... 210
A.11 Google Controlling Master Thread I/O Rate Patch License ......... 211
A.12 Google Perftools (TCMalloc utility) License ......................... 211
A.13 Google SMP Patch License ................................................. 212
A.14 lib_sql.cc License ........................................................... 213
A.15 libevent License .............................................................. 213
A.16 Linux-PAM License .......................................................... 215
A.17 md5 (Message-Digest Algorithm 5) License ......................... 216
A.18 memcached License ......................................................... 216
A.19 nt_servc (Windows NT Service class library) License ............. 217
A.20 OpenPAM License ............................................................ 217
A.21 Paramiko License ........................................................... 217
A.22 Percona Multiple I/O Threads Patch License ......................... 218
A.23 RegEX-Spencer Library License ........................................... 218
A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License ....... 219
A.25 Richard A. O’Keefe String Library License ......................... 219
A.26 SHA-1 in C License .......................................................... 220
A.27 zlib License ....................................................................... 220

The following is a list of the libraries we have included with the MySQL Server source and components used to test MySQL. We are thankful to all individuals that have created these. Some of the components require that their licensing terms be included in the documentation of products that include them. Cross references to these licensing terms are given with the applicable items in the list.

- Bjorn Benson
  For his safe_malloc (memory checker) package which is used in when you build MySQL using one of the BUILD/compile-*debug scripts or by manually setting the -DSAFEMALLOC flag.

- GroupLens Research Project
  The MySQL Quality Assurance team would like to acknowledge the use of the MovieLens Data Sets (10 million ratings and 100,000 tags for 10681 movies by 71567 users) to help test MySQL products and to thank the GroupLens Research Project at the University of Minnesota for making the data sets available.

MySQL 5.1

- Section A.2, “dtoa.c License”
- Section A.3, “Editline Library (libedit) License”
MySQL Cluster 7.1

- Section A.1, “ANTLR 3 License”
- Section A.2, “dtoa.c License”
- Section A.3, “Editline Library (libedit) License”
- Section A.4, “FindGTest.cmake License”
- Section A.5, “Fred Fish’s Dbug Library License”
- Section A.6, “getarg License”
- Section A.7, “GNU General Public License Version 2.0, June 1991”
- Section A.8, “GNU Lesser General Public License Version 2.1, February 1999”
- Section A.9, “GNU Libtool License”
- Section A.10, “GNU Readline License”
- Section A.11, “Google Controlling Master Thread I/O Rate Patch License”
- Section A.12, “Google Perftools (TCMalloc utility) License”
- Section A.13, “Google SMP Patch License”
- Section A.14, “lib_sql.cc License”
- Section A.15, “libevent License”
- Section A.16, “md5 (Message-Digest Algorithm 5) License”
- Section A.17, “nt_servc (Windows NT Service class library) License”
- Section A.18, “Percona Multiple I/O Threads Patch License”
- Section A.19, “RegEX-Spencer Library License”
- Section A.20, “RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License”
- Section A.21, “Richard A. O'Keefe String Library License”
- Section A.22, “Richard A. O'Keefe String Library License”
- Section A.23, “zlib License”
A.1 ANTLR 3 License

The following software may be included in this product:

ANTLR 3

ANTLR 3 License
[The BSD License]
Copyright (c) 2003-2007, Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2 dtoa.c License

The following software may be included in this product:
The author of this software is David M. Gay.

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

A.3 Editline Library (libedit) License

The following software may be included in this product:

Editline Library (libedit)

Some files are:

Copyright (c) 1992, 1993
The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some files are:
Copyright (c) 2001 The NetBSD Foundation, Inc.
All rights reserved.

This code is derived from software contributed to The NetBSD Foundation
by Anthony Mallet.

Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain the
   above copyright notice, this list of conditions
   and the following disclaimer.
2. Redistributions in binary form must reproduce the
   above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or
   other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC.
AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

Some files are:

Copyright (c) 1997 The NetBSD Foundation, Inc.
All rights reserved.

This code is derived from software contributed to The NetBSD Foundation
by Jaromir Dolecek.

Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain the
   above copyright notice, this list of conditions
   and the following disclaimer.
2. Redistributions in binary form must reproduce
   the above copyright notice, this list of conditions
   and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC.
AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

Some files are:

Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

Permission to use, copy, modify, and distribute this
software for any purpose with or without fee is hereby
granted, provided that the above copyright notice and
this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER
DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE
FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

A.4 FindGTest.cmake License

The following software may be included in this product:

FindGTest.cmake helper script (part of CMake)

Copyright 2009 Kitware, Inc.
Copyright 2009 Philip Lowman
Copyright 2009 Daniel Blezek

Distributed under the OSI-approved BSD License (the "License");
see accompanying file Copyright.txt for details.

This software is distributed WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the License for more information.

(To distributed this file outside of CMake, substitute the full
License text for the above reference.)

Thanks to Daniel Blezek for the GTEST_ADD_TESTS code

Text of Copyright.txt mentioned above:

CMake - Cross Platform Makefile Generator
Copyright 2000-2009 Kitware, Inc., Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.5 Fred Fish's Dbug Library License

The following software may be included in this product:

Fred Fish's Dbug Library

NOTICE

Copyright Abandoned, 1987, Fred Fish

This previously copyrighted work has been placed into the public domain by the author and may be freely used for any purpose, private or commercial.

Because of the number of inquiries I was receiving about the use of this product in commercially developed works I have decided to simply make it public domain to further its unrestricted use. I specifically would be most happy to see this material become a part of the standard Unix distributions by AT&T and the Berkeley Computer Science Research Group, and a standard part of the GNU system from the Free Software Foundation.

I would appreciate it, as a courtesy, if this notice is left in all copies and derivative works. Thank you.

The author makes no warranty of any kind with respect to this
A.6 getarg License

The following software may be included in this product:

getarg Function (getarg.h, getarg.c files)

Copyright (c) 1997 – 2000 Kungliga Tekniska Högskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Redistribution and use in source and binary forms, with
or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.
2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials
provided with the distribution.
3. Neither the name of the Institute nor the names of its
contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

A.7 GNU General Public License Version 2.0, June 1991

The following applies to all products licensed under the GNU General
Public License, Version 2.0: You may not use the identified files
except in compliance with the GNU General Public License, Version
2.0 (the "License.") You may obtain a copy of the License at
http://www.gnu.org/licenses/gpl-2.0.txt. A copy of the license is
also reproduced below. Unless required by applicable law or agreed
to in writing, software distributed under the License is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied. See the License for the specific language
governing permissions and limitations under the License.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS
WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it
does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version
2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
    type 'show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type 'show c'
for details.

The hypothetical commands 'show w' and 'show c' should show the
appropriate parts of the General Public License. Of course, the
commands you use may be called something other than 'show w' and
'show c'; they could even be mouse-clicks or menu items--whatever
suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program 'Gnomovision' (which makes passes at compilers) written
by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library,
you may consider it more useful to permit linking proprietary
applications with the library. If this is what you want to do, use
the GNU Lesser General Public License instead of this License.

A.8 GNU Lesser General Public License Version 2.1, February 1999

The following applies to all products licensed under the
GNU Lesser General Public License, Version 2.1: You may
not use the identified files except in compliance with
the GNU Lesser General Public License, Version 2.1 (the
"License"). You may obtain a copy of the License at
http://www.gnu.org/licenses/lgpl-2.1.html. A copy of the
license is also reproduced below. Unless required by
applicable law or agreed to in writing, software distributed
under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied. See the License for the specific language governing
permissions and limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.
Although the Lesser General Public License is less protective of the users’ freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices...
stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no
charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility
is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is
instance-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
   
   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY
15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
That's all there is to it!

A.9 GNU Libtool License

The following software may be included in this product:

GNU Libtool (The GNU Portable Library Tool)

If you are receiving a copy of the Oracle software in source code, you are also receiving a copy of two files (ltmain.sh and ltdl.h) generated by the GNU Libtool in source code. If you received the Oracle software under a license other than a commercial (non-GPL) license, then the terms of the Oracle license do NOT apply to these files from GNU Libtool; they are licensed under the following licenses, separately from the Oracle programs you receive.

Oracle elects to use GNU General Public License version 2 (GPL) for any software where a choice of GPL or GNU Lesser/Library General Public License (LGPL) license versions are made available with the language indicating that GPL/LGPL or any later version may be used, or where a choice of which version of the GPL/LGPL is applied is unspecified.

From GNU Libtool:

ltmain.sh - Provide generalized library-building support services.
NOTE: Changing this file will not affect anything until you rerun configure.
Originally by Gordon Matzigkeit, 1996

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

As a special exception to the GNU General Public License, if you distribute this file as part of a program that contains a configuration script generated by Autoconf, you may include it under the same distribution terms that you use for the rest of that program.

This component is licensed under Section A.7, “GNU General Public License Version 2.0, June 1991”

A.10 GNU Readline License

The following software may be included in this product:
GNU Readline Library

With respect to MySQL Server/Cluster software licensed under GNU General Public License, you are receiving a copy of the GNU Readline Library in source code. The terms of any Oracle license that might accompany the Oracle programs do NOT apply to the GNU Readline Library; it is licensed under the following license, separately from the Oracle programs you receive. Oracle elects to use GNU General Public License version 2 (GPL) for any software where a choice of GPL license versions are made available with the language indicating that GPLv2 or any later version may be used, or where a choice of which version of the GPL is applied is unspecified.

This component is licensed under Section A.7, "GNU General Public License Version 2.0, June 1991"

A.11 Google Controlling Master Thread I/O Rate Patch License

The following software may be included in this product:

Google Controlling master thread I/O rate patch

Copyright (c) 2009, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.12 Google Perftools (TCMalloc utility) License

The following software may be included in this product:

Google Perftools (TCMalloc utility)

Copyright (c) 1998-2006, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.13 Google SMP Patch License

The following software may be included in this product:

Google SMP Patch

Google SMP patch

Copyright (c) 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
A.14 lib_sql.cc License

The following software may be included in this product:

lib_sql.cc

Copyright (c) 2000
SWsoft company

This material is provided "as is", with absolutely no warranty
expressed or implied. Any use is at your own risk.

Permission to use or copy this software for any purpose is hereby
granted without fee, provided the above notices are retained on
all copies. Permission to modify the code and to distribute modified
code is granted, provided the above notices are retained, and a
notice that the code was modified is included with the above copyright
notice.

This code was modified by the MySQL team.

A.15 libevent License

The following software may be included in this product:

libevent

Copyright (c) 2000–2007 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR `\'AS IS\' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

--
Parts developed by Adam Langley
--
--
log.c
Based on err.c, which was adapted from OpenBSD libc *err*warncode.

Copyright (c) 2005 Nick Mathewson
Copyright (c) 2000 Dug Song
Copyright (c) 1993 The Regents of the University of California.
libevent License

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==
==
min_heap.h

Copyright (c) 2006 Maxim Yegorushkin
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==
==
win32.c

Copyright 2000-2002 Niels Provos
Copyright 2003 Michael A. Davis
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.16 Linux-PAM License

The following software may be included in this product:

Linux-PAM (pam-devel, Pluggable authentication modules for Linux)

Copyright Theodore Ts'o, 1996. All rights reserved.

(For the avoidance of doubt, Oracle uses and distributes this component under the terms below and elects not to do so under the GPL even though the GPL is referenced as an option below.)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Public License, in which case the provisions of the GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential bad interaction between the GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
md5 (Message-Digest Algorithm 5) License

A.17 md5 (Message-Digest Algorithm 5) License

The following software may be included in this product:

md5 (Message-Digest Algorithm 5)

This code implements the MD5 message-digest algorithm. The algorithm is due to Ron Rivest. This code was written by Colin Plumb in 1993, no copyright is claimed. This code is in the public domain; do with it what you wish.

Equivalent code is available from RSA Data Security, Inc. This code has been tested against that, and is equivalent, except that you don't need to include two pages of legalese with every copy.

The code has been modified by Mikael Ronstroem to handle calculating a hash value of a key that is always a multiple of 4 bytes long. Word 0 of the calculated 4-word hash value is returned as the hash value.

A.18 memcached License

The following software may be included in this product:

memcached

Copyright (c) 2003, Danga Interactive, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Danga Interactive nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
A.19 nt_servc (Windows NT Service class library) License

The following software may be included in this product:

nt_servc (Windows NT Service class library)

Windows NT Service class library
Copyright Abandoned 1998 Irena Pancirov - Irnet Snc
This file is public domain and comes with NO WARRANTY of any kind

A.20 OpenPAM License

The following software may be included in this product:

OpenPAM

Copyright (c) 2002-2003 Networks Associates Technology, Inc.
Copyright (c) 2004-2007 Dag-Erling Smørgrav
All rights reserved.

This software was developed for the FreeBSD Project by
ThinkSec AS and Network Associates Laboratories, the
Security Research Division of Network Associates, Inc.
under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"),
as part of the DARPA CHATS research program.

Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain the above
   copyright notice, this list of conditions and the
   following disclaimer.
2. Redistributions in binary form must reproduce the
   above copyright notice, this list of conditions and
   the following disclaimer in the documentation and/or
   other materials provided with the distribution.
3. The name of the author may not be used to endorse or
   promote products derived from this software without
   specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.21 Paramiko License

The following software may be included in this product:
Paramiko

You are receiving a copy of Paramiko in both source and object code. The terms of the Oracle license do NOT apply to the Paramiko program; it is licensed under the following license, separately from the Oracle programs you receive. If you do not wish to install this program, you may delete the Paramiko folder and all its contents.

This component is licensed under Section A.8, “GNU Lesser General Public License Version 2.1, February 1999”.

A.22 Percona Multiple I/O Threads Patch License

The following software may be included in this product:

Percona Multiple I/O threads patch

Copyright (c) 2008, 2009 Percona Inc
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Percona Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of Percona Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.23 RegEX-Spencer Library License

The following software may be included in this product: Henry Spencer's Regular-Expression Library (RegEX-Spencer)

Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License

The following software may be included in this product:

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement
Funding for the RFC Editor function is currently provided by the Internet Society.

A.25 Richard A. O'Keefe String Library License

The following software may be included in this product:

Richard A. O'Keefe String Library

The Richard O'Keefe String Library is subject to the following notice:

These files are in the public domain. This includes getopt.c, which is the work of Henry Spencer, University of Toronto Zoology, who says of it "None of this software is derived from Bell software. I had no access to the source for Bell's versions at the time I wrote it. This software is hereby explicitly placed in the public domain.
It may be used for any purpose on any machine by anyone." I would
greatly prefer it if "my" material received no military use.

The t ctype.h file is subject to the following notice:

Copyright (C) 1998, 1999 by Pruet Boonma, all rights reserved.
Copyright (C) 1998 by Theppitak Karoonboonyanan, all rights reserved.

Permission to use, copy, modify, distribute and sell this software and its
documentation for any purpose is hereby granted without fee, provided that the above
copyright notice appear in all copies.

Smaphan Raruenrom and Pruet Boonma makes no representations about
the suitability of this software for any purpose. It is provided
"as is" without express or implied warranty.

A.26 SHA-1 in C License

The following software may be included in this product:

SHA-1 in C

By Steve Reid <steve@edmweb.com>
100% Public Domain

A.27 zlib License

The following software may be included in this product:

zlib

Oracle gratefully acknowledges the contributions of Jean-loup Gailly and Mark Adler in creating the zlib
general purpose compression library which is used in this product.

zlib.h -- interface of the 'zlib' general purpose compression library
Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.5, April 19th, 2010
Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty.
In no event will the authors be held liable for any damages arising from the
use of this software. Permission is granted to anyone to use this software
for any purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would
   be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not
   be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
Mark Adler madler@alumni.caltech.edu