MySQL Installation Guide
Abstract

This is the MySQL Installation Guide from the MySQL 5.5 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: 2016-01-06 (revision: 46015)
# Table of Contents

Preface and Legal Notices ................................................................. vii
1 Installing and Upgrading MySQL ...................................................... 1
2 General Installation Guidance .......................................................... 3
   2.1 Which MySQL Version and Distribution to Install .............................. 3
   2.2 How to Get MySQL ...................................................................... 4
   2.3 Verifying Package Integrity Using MD5 Checksums or GnuPG ................. 5
      2.3.1 Verifying the MD5 checksum .................................................. 5
      2.3.2 Signature Checking Using GnuPG .......................................... 6
      2.3.3 Signature Checking Using Gpg4win for Windows ......................... 9
      2.3.4 Signature Checking Using RPM ............................................. 14
   2.4 Installation Layouts .................................................................... 15
   2.5 Compiler-Specific Build Characteristics ......................................... 15
3 Installing MySQL on Unix/Linux Using Generic Binaries ......................... 17
4 Installing MySQL from Source ......................................................... 21
   4.1 MySQL Layout for Source Installation ........................................... 23
   4.2 Installing MySQL Using a Standard Source Distribution ..................... 23
   4.3 Installing MySQL Using a Development Source Tree ....................... 27
   4.4 MySQL Source-Configuration Options .......................................... 29
   4.5 Dealing with Problems Compiling MySQL ...................................... 41
   4.6 MySQL Configuration and Third-Party Tools ................................... 42
5 Installing MySQL on Microsoft Windows .......................................... 43
   5.1 MySQL Installation Layout on Microsoft Windows ............................ 46
   5.2 Choosing An Installation Package ............................................... 46
   5.3 Installing MySQL on Microsoft Windows Using MySQL Installer .......... 47
      5.3.1 MySQL Installer GUI ......................................................... 49
      5.3.2 MySQL Installer Console ................................................... 73
   5.4 MySQL Notifier ........................................................................ 76
      5.4.1 Remote monitoring set up and installation instructions .................. 83
   5.5 Installing MySQL on Microsoft Windows Using an MSI Package .......... 88
      5.5.1 Using the MySQL Installation Wizard ..................................... 90
      5.5.2 Automating MySQL Installation on Microsoft Windows Using the MSI Package .................................................................................. 93
      5.5.3 Removing MySQL When Installed from the MSI Package .......... 94
   5.6 MySQL Server Instance Configuration Wizard ................................ 95
      5.6.1 Starting the MySQL Server Instance Configuration Wizard .......... 95
      5.6.2 Choosing a Maintenance Option ............................................ 96
      5.6.3 Choosing a Configuration Type ............................................. 96
      5.6.4 The Server Type Dialog ...................................................... 97
      5.6.5 The Database Usage Dialog ................................................. 99
      5.6.6 The InnoDB Tablespace Dialog .......................................... 100
      5.6.7 The Concurrent Connections Dialog .................................... 101
      5.6.8 The Networking and Strict Mode Options Dialog ....................... 102
      5.6.9 The Character Set Dialog .................................................. 103
      5.6.10 The Service Options Dialog ............................................. 103
      5.6.11 The Security Options Dialog ............................................ 104
      5.6.12 The Confirmation Dialog .................................................. 106
      5.6.13 MySQL Server Instance Config Wizard: Creating an Instance from the Command Line ........................................................................................................... 107
   5.7 Installing MySQL on Microsoft Windows Using a noinstall Zip Archive . . . 110
      5.7.1 Extracting the Install Archive ............................................... 110
      5.7.2 Creating an Option File ..................................................... 111
      5.7.3 Selecting a MySQL Server Type .......................................... 112
A.12 Google Perftools (TCMalloc utility) License .................................................. 224
A.13 Google SMP Patch License ............................................................................. 224
A.14 lib_sql.cc License ......................................................................................... 225
A.15 Libaio License ............................................................................................... 225
A.16 Libevent License ............................................................................................ 225
A.17 Linux-PAM License ......................................................................................... 228
A.18 md5 (Message-Digest Algorithm 5) License .................................................. 228
A.19 memcached License ....................................................................................... 229
A.20 nt_servc (Windows NT Service class library) License ................................... 229
A.21 OpenPAM License ......................................................................................... 229
A.22 Paramiko License .......................................................................................... 230
A.23 Percona Multiple I/O Threads Patch License .................................................. 230
A.24 Python License ............................................................................................... 231
A.25 Red HAT RPM Spec File License .................................................................... 241
A.26 RegEX-Spencer Library License ..................................................................... 242
A.27 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License ............................ 242
A.28 Richard A. O'Keefe String Library License .................................................... 243
A.29 SHA-1 in C License ....................................................................................... 243
A.30 zlib License .................................................................................................... 243
Preface and Legal Notices

This is the MySQL Installation Guide from the MySQL 5.5 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.
Chapter 1 Installing and Upgrading MySQL

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 10.1, “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 5.5 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 10.2.2, “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, “Which MySQL Version and 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 9, Postinstallation Setup and Testing for information about making sure the MySQL server is working properly. Also refer to the information provided in Section 9.4, “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 12, 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.
- FreeBSD, see Installing MySQL on FreeBSD.

**Microsoft Windows**

For instructions on installing MySQL on Microsoft Windows, using either the MySQL Installer standalone MSI, or Zipped binary, see Chapter 5, Installing MySQL on Microsoft Windows.

For information about managing MySQL instances, see Section 5.4, “MySQL Notifier”.

For details and instructions on building MySQL from source code using Microsoft Visual Studio, see Chapter 4, Installing MySQL from Source.

- **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 an OS X Launch Daemon to automatically start and stop MySQL, see Section 6.3, “Installing a MySQL Launch Daemon”.

  For information on the MySQL Preference Pane, see Section 6.4, “Installing and Using the MySQL Preference Pane”.
Chapter 2 General Installation Guidance

Table of Contents

2.1 Which MySQL Version and Distribution to Install ................................................................. 3
2.2 How to Get MySQL ..................................................................................................................... 4
2.3 Verifying Package Integrity Using MD5 Checksums or GnuPG ................................................. 5
  2.3.1 Verifying the MD5 Checksum .......................................................................................... 5
  2.3.2 Signature Checking Using GnuPG ................................................................................... 6
  2.3.3 Signature Checking Using Gpg4win for Windows ........................................................... 9
  2.3.4 Signature Checking Using RPM ....................................................................................... 14
2.4 Installation Layouts .................................................................................................................... 15
2.5 Compiler-Specific Build Characteristics .................................................................................. 15

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 Which MySQL Version and 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 on the MySQL Web site.

When preparing to install MySQL, you should decide which version to use, and which distribution format (binary or source) to use for the installation.

First, decide if you want to install a development release or a GA release. Development releases have the newest features, but are not recommended for production use. GA (General Availability) releases, also called production or stable releases, are meant for production use. We recommend to use the most recent GA release.

The naming scheme in MySQL 5.5 uses release names that consist of three numbers and a suffix; for example, mysql-5.6.1-m1. 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 (6) is the release level. Taken together, the major version and release level constitute the release series number.

• The third number (1) 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 can 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:
How to Get MySQL

- 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.

- mN (for example, m1, m2, m3, ...) indicate a milestone number. MySQL development uses a milestone model, in which each milestone proceeds through a small number of versions with a tight focus on a small subset of thoroughly tested features. Following the releases for one milestone, development proceeds with another small number of releases that focuses on the next small set of features, also thoroughly tested. Features within milestone releases may be considered to be of pre-production quality.

- 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.

Once you've chosen which MySQL version to install, you need to decide which distribution to install for your operating system. For most use cases, a binary distribution is the right choice. Binary distributions are available in native format for many platforms, such as RPM packages for Linux, or DMG packages for OS X. Distributions are also available in more generic formats such as Zip archives or compressed tar files. On Windows, you can use the MySQL Installer to install a binary distribution.

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:
  - -DWITH_LIBWRAP=1 for TCP wrappers support.
  - -DWITH_ZLIB={system|bundled} for features that depend on compression
  - -DWITH_DEBUG=1 for debugging support

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 Git 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.

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

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,
Verifying Package Integrity Using MD5 Checksums or GnuPG

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
shell> md5sum mysql-standard-5.5.48-linux-i686.tar.gz
aaab65abbec64d5e907dcd41b8699945  mysql-standard-5.5.48-linux-i686.tar.gz

shell> md5.exe mysql-installer-community-5.5.48.msi
aaab65abbec64d5e907dcd41b8699945  mysql-installer-community-5.5.48.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 download the GnuPG 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+owwRBAC14GIfUfCYEDSIEpRV3SAUFb7oqKHH/nJZkyQT7hb9PILUWC3 RODjQreyCITIrwdWYROKUGk2PmveMVgw2u2wDMNABLnpprVPWbdCk96+OmsL9brZ f2w0V0UgCyMrVw0hD9hWg4y7Q/THQoAoDgjI/A69/0L07V1W9/8VUH0pgQwGvzv3 BqOaRznNCRvAaUv0vzHrCeAJo0qK1+lISlnZMD1qVufeFexsh575+/yjeKegNW hxPWPvR9wTv3r+ZJ+2Zf2Ze8vypquKXNQ/hnEIUH3RJyI9o8tgxVxVXQ7fjY1L K9z7kTbFlF7wsmEWMHyhCr0eChtlzggIKr5j65sAgACypB/VBfixwh6z2d7IfN kYwPxBAcMcgP87NtkkamtpClia2200ODNANwpk4317jDmgqobX9q9mAnw+uNDI QjXEM6FbI0LzciNLYasfvaPEOMDKqHkqJyI1sNpVud18R0bKPNvqyfeep rv0sxQxyUQXCN9o7wqf8FN3P0kDasDa3pjpFkCwv6z266QZ7qTTX1UUQwUmVs ZWFZ5BfbmpvbLcmuzyA8BiXcnWtYmpbGRAB3N1m9yYNQsZ5j2b01+IgKe EKEECACKGyGMEwIBwMBCCBCUMAEgFDQAIAQXqIAZIAQUOHUGJUmGbLyAa CRCMYcOv7ULHv9vDACKjS19gQVIG/Ve+u5/L1v3yb1+2gCdZ7oa341x7torV3U 6rrD0D9fgeq00Q15IFMIFBhY2h22Zgc1inbmu8yZbRZKgKhd35teXNkb5j b2OpI0xidWlSZE8TeXNxcb5c2+0j1BEMEBCAC8FAK53PAoohSbd1w12eSBNex nc5j2b0gZ21SaBDzd8SW1Hdcvmpmcgzc9vbgAKRCKCMcY70ULH9b9uA9j9X0Kr0 oXJF19yV1903Z41X09AwCwc3r9cNjYs73e9aejlJ0s2/rcclJdyIIZQTEIAQUC R6ytuADJTYBGvAUBLwobDAVAYwIDFgjBaHeeAB1EJt1xjTQCjuH1B2VNBcAAQg kufc7s4GUEjxLoi7V1wCwXcATkGbe0DN34Lr19jFcbwXWOX14ms1ADfb5piEwE EKECIAWAFj4+oEFGwMlAMsACqKQSDhiRfYkI4QCFNWedEnK3vTse59xJ+k jQSAimmAnJu0u2FbPV0Xf0Q3Ng1bg0cU9ifExEEXCAFAWAFaj5CGMfgwlioka CQgtXvXmTca6JD4WkCqCGnmoJmJyop5WxkckykfanokA79E9E8bwrSDK2ou iZMrhnISyui1YEBCBEEAAYAFa+j468ACAgkQRV8JISHJIda/2lGcg21HiNMBYd pIB1UspQJILAOeAnrMurWETujqZj9E9ono9Hldkeb4aIYBEBECEBAAYAFa+jQ3CA CQgKh2avqBzZbtpWcAdfd9F1r7C47W8GK0O3t7zH33N2kArCJlw3w76N6QO 83nki1a2dpeIYEBEEBFAFkNbs+ooACOGqK1jQy4RDEcELQoKXKwJN0 gZztM7kJys4a3AMNtRRMwQVTQ399C9o1Paj3k3a10lltG1YBEECAIAAYAFkTwMYA CQgQWnR9gGtCIWhCQbg/lbA/mxh9Qyf5M4vAnAhztL57wamlK1g m7nVGtUq77fe1wEEBECAIAFaFKgBywEFGWq1YsAcqKQFhQ2id3osejCQCd8sBjn NDC/dms/c8sOG0uX09WMCnGkAnj1YWOD+Qhos3yml/Ut90EAnhnkjcF1wEEBECAAF AkGBy2QwFyl2vAgCqK1jLC36+IITpIl1wCdFVNVUBX8emfXoPmd4d95425ItjMA niISPA/Zaf3j3OMLka4r4F0QPPdrclEwEEBECAIAFkGBy2IFgWyl2soAqCqA3Ds 2ZVD9H3M9gbCbyzRSGPXOoFpR88jKxdmbjweqXeAnhRs4G2g/3Q7duhTL1SPT1SH JXWUXEwEEBECIAAAFKhKxyQwGwUXqWyXgAqCqQf5XXCsEpp8j1QVChGvVvKpQowsw w7P9wscEwetvFkvAn+rH/4/Djy10Lk2In5jzK1dpVF1wEEBECAIAFkIR7w0f gwVvWSNHAcyGQKuKXruxavwvEWGzkL59Qpyc9l1GcF8a1yaULv4AN35xKumm 9gcIcI1IPR2r14FzQriAGpWywEEBECAIAFkAN7QfoGw6d81AACqPKXEnNITJ6+G3 XACdC0A53A5OGM7w750W1uk4qJz9C9aRMMvAnGq3YtoXlplf2UJpZsQs1qEwE EheECAIAFKA+o1rAwgmdpIAcKqQ8tddtY0+C2Zzyw3xwCQtwCdwLw20w0cNuXPLNe EUAFTwNan6RH8O0noSvADVDTKZxZacJU35M5QwEEBECIAAAFkEEOcKQwFgwaWm5Qg ACGqCoc0rQ9d1l/1/833CEQIAEwEhEAECAIAFkAPeTQFwUji/1gAcqoKBa3JhXYn6djJACcD1A4 UtxK84J2135Qyoh+9y2d4714N1w1ssl0/9dICJkoxq2j51HDFq6o1wEwEEBECAAF A5JNTYQFwqVWQvCqQLzbv36U3YTMDFACG1t505NVR5f9j5656SIFP2s9z2kqa niuc3nh9/1w4rsbIy/K/9OJP3JnMINEExEAEAAFAKPMGvQwlXJMYBACqQZnH/b y2MC2t/1f8rCFi6ex7s5IBH80wkmF232kcbz0/gbAAnJ2rUqEpbaoqkQcG5tnBNmRv NAAjI6wEEBECAIAFkAIPkYg24Fgw0m14ACqGcadaEDHkrsbq97cFg7L29W3n+rYf nUlN1fHo4H8ZM04839k0utj7/phonAqYIpuu1TfTwdB1LUEXECABAFCwcwAQQD
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
```

```
gpg: requesting key 5072E1F5 from hkp server keys.gnupg.net
```

```
gpg: key 5072E1F5: "MySQL Release Engineering <mysql-build@oss.oracle.com>"
53 new signatures
```

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

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

```
shell> gpg --keyserver hkp://keys.gnupg.net/keys/5072E1F5
```

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

```
```
Signature Checking Using GnuPG

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:

```
shell> rpm --import mysql_pubkey.asc
```

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-5.5.48-linux-i686.tar.gz</td>
</tr>
<tr>
<td>Signature file</td>
<td>mysql-standard-5.5.48-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-5.5.48-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-5.5.48-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>"
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**.
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.
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>
<thead>
<tr>
<th>File Type</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution file</td>
<td>mysql-installer-community-5.5.48.msi</td>
</tr>
<tr>
<td>Signature file</td>
<td>mysql-installer-community-5.5.48.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.
Figure 2.6 The Decrypt/Verify Results: Bad

The 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-5.5.48-0.glibc23.i386.rpm
MySQL-server-5.5.48-0.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
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:

```shell
```

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 6.1, “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.
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, 5.5.48), 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.

Warning

MySQL has a dependency on the libaio library. Data directory initialization and subsequent server startup steps will fail if this library is not installed locally. If necessary, install it using the appropriate package manager. For example, on Yum-based systems:

shell> yum search libaio  # search for info
shell> yum install libaio  # install library

Or, on APT-based systems:

shell> apt-cache search libaio  # search for info
shell> apt-get install libaio1  # install library

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>
</tbody>
</table>
Create a mysql User and Group

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<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 `mysqld` binary are available as `mysqld-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:

```
shell> groupadd mysql
shell> useradd -r -g mysql -s /bin/false mysql
shell> cd /usr/local
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.gz
shell> ln -s full-path-to-mysql-VERSION-OS mysql
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
# 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
```

**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 9.4, “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
```
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
```

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/](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 9, *Postinstallation Setup and Testing*. 

---

**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.
Chapter 4 Installing MySQL from Source

Table of Contents

4.1 MySQL Layout for Source Installation ......................................................... 23
4.2 Installing MySQL Using a Standard Source Distribution ................................. 23
4.3 Installing MySQL Using a Development Source Tree ..................................... 27
4.4 MySQL Source-Configuration Options ......................................................... 29
4.5 Dealing with Problems Compiling MySQL ................................................... 41
4.6 MySQL Configuration and Third-Party Tools ............................................... 42

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 Oracle produces 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.

Note
This section describes how to build MySQL from source using CMake. Before MySQL 5.5, source builds used the GNU autotools on Unix-like systems. Source builds on Windows used Autotools but not CMake, you might find these transition instructions helpful: Autotools to CMake Transition Guide.

Source Installation Methods

There are two methods for installing MySQL from source:

• Use a standard MySQL source distribution. To obtain a standard distribution, see Section 2.2, “How to Get MySQL”. For instructions on building from a standard distribution, see Section 4.2, “Installing MySQL Using a Standard Source Distribution”.

Standard distributions are available as compressed tar 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 5.5.48. File names for source distributions can be distinguished from those for precompiled binary distributions in that source distribution names are generic and include no platform name, whereas binary distribution names include a platform name indicating the type of system for which the distribution is intended (for example, pc-linux-i686 or winx64).

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

Source Installation System Requirements

Installation of MySQL from source requires several development tools. Some of these tools are needed no matter whether you use a standard source distribution or a development source tree. Other tool requirements depend on which installation method you use.
To install MySQL from source, the following system requirements must be satisfied, regardless of installation method:

- **CMake**, which is used as the build framework on all platforms. CMake can be downloaded from [http://www.cmake.org](http://www.cmake.org).

- 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/](http://www.gnu.org/software/make/).

- A working ANSI C++ compiler. GCC 4.2.1 or later, Sun Studio 12 or later, Visual Studio 2008 or later, and many current vendor-supplied compilers are known to work.

- Perl is needed if you intend to run test scripts. Most Unix-like systems include Perl. On Windows, you can use a version such as ActiveState Perl.

To install MySQL from a standard source distribution, one of the following tools is required to unpack the distribution file:

- For a `.tar.gz` compressed tar file: 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/](http://www.gnu.org/software/tar/).

- For a `.zip` Zip archive: WinZip or another tool that can read `.zip` files.

- For an `.rpm` RPM package: The rpmbuild program used to build the distribution unpacks it.

To install MySQL from a development source tree, the following additional tools are required:

- The Git revision control system is required to obtain the development source code. The GitHub Help provides instructions for downloading and installing Git on different platforms. MySQL officially joined GitHub in September, 2014. For more information about MySQL’s move to GitHub, refer to the announcement on the MySQL Release Engineering blog: [MySQL on GitHub](https://github.com/mysql/mysql-server).

- **bison** 2.1 or newer, available from [http://www.gnu.org/software/bison/](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. bison is available from [http://www.gnu.org/software/bison/](http://www.gnu.org/software/bison/). bison for Windows can be downloaded from [http://gnuwin32.sourceforge.net/packages/bison.htm](http://gnuwin32.sourceforge.net/packages/bison.htm). Download the package labeled “Complete package, excluding sources”. On Windows, the default location for bison is the `C:\Program Files \GnuWin32` directory. Some utilities may fail to find bison because of the space in the directory name. Also, Visual Studio may simply hang if there are spaces in the path. You can resolve these problems by installing into a directory that does not contain a space; for example `C:\GnuWin32`.

- On OpenSolaris and Solaris Express, m4 must be installed in addition to bison. m4 is available from [http://www.gnu.org/software/m4/](http://www.gnu.org/software/m4/).

**Note**

If you have to install any programs, modify your PATH environment variable to include any directories in which the programs are located. See Setting Environment Variables.
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 source, the installation step installs files under /usr/local/mysql. The component locations under the installation directory are the same as for binary distributions. See Table 3.1, “MySQL Installation Layout for Generic Unix/Linux Binary Package”, and Section 5.1, “MySQL Installation Layout on Microsoft Windows”. To configure installation locations different from the defaults, use the options described at Section 4.4, “MySQL Source-Configuration Options”.

4.2 Installing MySQL Using a Standard Source Distribution

To install MySQL from a standard source distribution:

1. Verify that your system satisfies the tool requirements listed at Chapter 4, Installing MySQL from Source.
2. Obtain a distribution file using the instructions in Section 2.2, “How to Get MySQL”.
3. Configure, build, and install the distribution using the instructions in this section.
4. Perform postinstallation procedures using the instructions in Chapter 9, Postinstallation Setup and Testing.

In MySQL 5.5, CMake is used as the build framework on all platforms. The instructions given here should enable you to produce a working installation. For additional information on using CMake to build MySQL, see How to Build MySQL Server with CMake.

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 or Zip archive source distribution is similar to the process for installing from a generic binary distribution (see Chapter 3, Installing MySQL on Unix/Linux Using Generic Binaries), except that it is used on all platforms and includes steps to configure and compile the distribution. For example, with a compressed tar file source distribution on Unix, the basic installation command sequence looks like this:

```
# Preconfiguration setup
shell> groupadd mysql
shell> useradd -r -g mysql -s /bin/false mysql
# Beginning of source-build specific instructions
shell> cd mysql-VERSION.tar.gz
shell> tar xzvf mysql-VERSION.tar.gz
shell> cd mysql-VERSION
shell> cmake .
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> scripts/mysql_install_db --user=mysql
shell> chown -R root .
```
Perform Preconfiguration Setup

A more detailed version of the source-build specific instructions is shown following.

Note
The procedure shown here does not set up any passwords for MySQL accounts. After following the procedure, proceed to Chapter 9, Postinstallation Setup and Testing, for postinstallation setup and testing.

Perform Preconfiguration Setup

On Unix, 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. Then perform the following steps as the mysql user, except as noted.

Obtain and Unpack the Distribution

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

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

Unpack the distribution into the current directory:

• To unpack a compressed tar file, tar can uncompress and unpack the distribution if it has z option support:

```shell
shell> tar zxvf mysql-VERSION.tar.gz
```

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

```shell
shell> gunzip < mysql-VERSION.tar.gz | tar xvf -
```

Alternatively, CMake can uncompress and unpack the distribution:

```shell
shell> cmake -E tar zxvf mysql-VERSION.tar.gz
```

• To unpack a Zip archive, use WinZip or another tool that can read .zip files.

Unpacking the distribution file creates a directory named mysql-VERSION.

Configure the Distribution

Change location into the top-level directory of the unpacked distribution:

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

Configure the source directory. The minimum configuration command includes no options to override configuration defaults:
Configure the Distribution

shell> cmake .

On Windows, specify the development environment. For example, the following commands configure MySQL for 32-bit or 64-bit builds, respectively:

shell> cmake . -G "Visual Studio 9 2008"
shell> cmake . -G "Visual Studio 9 2008 Win64"

On OS X, to use the Xcode IDE:

shell> cmake . -G Xcode

When you run `cmake`, you might want to add options to the command line. Here are some examples:

- `-DBUILD_CONFIG=mysql_release`: Configure the source with the same build options used by Oracle to produce binary distributions for official MySQL releases.
- `-DCMAKE_INSTALL_PREFIX=dir_name`: Configure the distribution for installation under a particular location.
- `-DCPACK_MONOLITHIC_INSTALL=1`: Cause `make package` to generate a single installation file rather than multiple files.
- `-DWITH_DEBUG=1`: Build the distribution with debugging support.

For a more extensive list of options, see Section 4.4, “MySQL Source-Configuration Options”.

To list the configuration options, use one of the following commands:

shell> cmake . -L # overview
shell> cmake . -LH # overview with help text
shell> cmake . -LAH # all params with help text
shell> ccmaker . # interactive display

If `CMake` fails, you might need to reconfigure by running it again with different options. If you do reconfigure, take note of the following:

- If `CMake` is run after it has previously been run, it may use information that was gathered during its previous invocation. This information is stored in `CMakeCache.txt`. When `CMake` 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 `CMake`, 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 object files or configuration information from being used, run these commands on Unix before re-running `CMake`:

shell> make clean
shell> rm CMakeCache.txt

Or, on Windows:

shell> devenv MySQL.sln /clean
shell> del CMakeCache.txt

If you build out of the source tree (as described later), the `CMakeCache.txt` file and all built files are in the build directory, so you can remove that directory to object files and cached configuration information.
If you are going to send mail to a MySQL mailing list to ask for configuration assistance, first check the files in the `CMakeFiles` directory for useful information about the failure. To file a bug report, please use the instructions in How to Report Bugs or Problems.

## Build the Distribution

### On Unix:

```
shell> make
shell> make VERBOSE=1
```

The second command sets `VERBOSE` to show the commands for each compiled source.

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

### On Windows:

```
shell> devenv MySQL.sln /build RelWithDebInfo
```

It is possible to build out of the source tree to keep the tree clean. If the top-level source directory is named `mysql-src` under your current working directory, you can build in a directory named `bld` at the same level like this:

```
shell> mkdir bld
shell> cd bld
shell> cmake ../mysql-src
```

The build directory need not actually be outside the source tree. For example, to build in a directory, you can build in a directory named `bld` under the top-level source tree, do this, starting with `mysql-src` as your current working directory:

```
shell> mkdir bld
shell> cd bld
shell> cmake ..
```

If you have multiple source trees at the same level (for example, to build multiple versions of MySQL), the second strategy can be advantageous. The first strategy places all build directories at the same level, which requires that you choose a unique name for each. With the second strategy, you can use the same name for the build directory within each source tree.

If you have gotten to the compilation stage, but the distribution does not build, see Section 4.5, “Dealing with Problems Compiling MySQL”, for help. If that does not solve the problem, please enter it 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.

## Install the Distribution

### On Unix:

```
shell> make install
```

This installs the files under the configured installation directory (by default, `/usr/local/mysql`). You might need to run the command as `root`. 
Perform Postinstallation Setup

To install in a specific directory, add a `DESTDIR` parameter to the command line:

```
shell> make install DESTDIR="/opt/mysql"
```

Alternatively, generate installation package files that you can install where you like:

```
shell> make package
```

This operation produces one or more `.tar.gz` files that can be installed like generic binary distribution packages. See Chapter 3, *Installing MySQL on Unix/Linux Using Generic Binaries*. If you run `CMake` with `-DCPACK_MONOLITHIC_INSTALL=1`, the operation produces a single file. Otherwise, it produces multiple files.

On Windows, generate the data directory, then create a `.zip` archive installation package:

```
shell> devenv MySQL.sln /build RelWithDebInfo /project initial_database
shell> devenv MySQL.sln /build RelWithDebInfo /project package
```

You can install the resulting `.zip` archive where you like. See Section 5.7, “Installing MySQL on Microsoft Windows Using a noinstall Zip Archive”.

### Perform Postinstallation Setup

The remainder of the installation process involves setting up the configuration file, creating the core databases, and starting the MySQL server. For instructions, see Chapter 9, *Postinstallation Setup and Testing*.

#### 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 9, *Postinstallation Setup and Testing*.

### 4.3 Installing MySQL Using a Development Source Tree

This section describes how to install MySQL from the latest development source code, which is currently hosted on GitHub. To obtain the MySQL Server source code from this repository hosting service, you can set up a local MySQL Git repository.

On GitHub, MySQL Server and other MySQL projects are found on the MySQL page. The MySQL Server project is a single repository that contains branches for several MySQL series, such as 5.5, 5.6, and 5.7.

MySQL officially joined GitHub in September, 2014. For more information about MySQL’s move to GitHub, refer to the announcement on the MySQL Release Engineering blog: [MySQL on GitHub](https://dev.mysql.com/doc/mysql/release/5.7/en/mysql-on-github.html)

#### Prerequisites for Installing from Development Source

To install MySQL from a development source tree, your system must satisfy the tool requirements outlined in Chapter 4, *Installing MySQL from Source*.

#### Setting Up a MySQL Git Repository

To set up a MySQL Git repository on your machine, use this procedure:

1. Clone the MySQL Git repository to your machine. The following command clones the MySQL Git repository to a directory named `mysql-server`. The download size is approximately 437 MB. The initial download will take some time to complete, depending on the speed of your connection.
Setting Up a MySQL Git Repository

```bash
~$ git clone https://github.com/mysql/mysql-server.git
Cloning into 'mysql-server'...
remote: Counting objects: 1035465, done.
remote: Receiving objects: 100% (1035465/1035465), 437.48 MiB | 5.10 MiB/s, done.
Resolving deltas: 100% (855607/855607), done.
Checking connectivity... done.

2. When the clone operation completes, the contents of your local MySQL Git repository appear similar to
the following:

```bash
~$ cd mysql-server
~/mysql-server$ ls
BUILD            COPYING             libmysqld    regex     tests
BUILD-CMAKE      dbug                libservices  scripts   unitest
client           Docs                man          sql        VERSION
cmake            extra               mysql-test   sql-bench   vio
CMakeLists.txt   include             mysys        sql-common  win
cmd-line-utils   INSTALL-SOURCE      packaging    storage    zlib
config.h.cmake   INSTALL-WIN-SOURCE  plugin       strings
configure.cmake  libmysql            README       support-files
```

3. Use the `git branch -r` command to view the remote tracking branches for the MySQL repository.

```bash
~/mysql-server$ git branch -r
origin/5.5
origin/5.6
origin/5.7
origin/HEAD -> origin/5.7
origin/cluster-7.2
origin/cluster-7.3
origin/cluster-7.4
```

4. To view the branches that are checked out in your local repository, issue the `git branch` command. When you cloned the MySQL Git repository, the MySQL 5.7 branch was checked out automatically. The asterisk identifies the 5.7 branch as the active branch.

```bash
~/mysql-server$ git branch
* 5.7
```

5. To check out a different MySQL branch, run the `git checkout` command, specifying the branch name. For example, to checkout the MySQL 5.5 branch:

```bash
~/mysql-server$ git checkout 5.5
Branch 5.5 set up to track remote branch 5.5 from origin.
Switched to a new branch '5.5'
```

6. Run `git branch` again to verify that the MySQL 5.5 branch is present. MySQL 5.5, which is the last branch you checked out, is marked by an asterisk indicating that it is the active branch.

```bash
~/mysql-server$ git branch
* 5.5
  5.7
```

The `git checkout` command is also used to switch branches. For example, to make MySQL 5.7 the active branch again, you would run `git checkout 5.7`.

7. To obtain changes made after your initial setup of the MySQL Git repository, switch to the branch you want to update and issue the `git pull` command:

```bash
~/mysql-server$ git checkout 5.5
```
To examine the commit history, use the `git log` option:

```
~/mysql-server$ git log
```

You can also browse commit history and source code on the GitHub MySQL site.

If you see changes or code that you have a question about, send an email to the MySQL internals mailing list. See MySQL Mailing Lists. For information about contributing a patch, see Contributing to MySQL Server.

8. After you have cloned the MySQL Git repository and have checked out the branch you want to build, you can build MySQL Server from the source code. Instructions are provided in Section 4.2, “Installing MySQL Using a Standard Source Distribution”, except that you skip the part about obtaining and unpacking the distribution.

Be careful about installing a build from a distribution source tree 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 CMake with values for the `CMAKE_INSTALL_PREFIX`, `MYSQL_TCP_PORT`, and `MYSQL_UNIX_ADDR` 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.

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

### 4.4 MySQL Source-Configuration Options

The CMake program provides a great deal of control over how you configure a MySQL source distribution. Typically, you do this using options on the CMake command line. For information about options supported by CMake, run either of these commands in the top-level source directory:

```
shell> cmake . -LH
shell> ccmake .
```

You can also affect CMake using certain environment variables. See Chapter 11, EnvironmentVariables.

The following table shows the available CMake options. In the Default column, PREFIX stands for the value of the `CMAKE_INSTALL_PREFIX` option, which specifies the installation base directory. This value is used as the parent location for several of the installation subdirectories.

<table>
<thead>
<tr>
<th>Formats</th>
<th>Description</th>
<th>Default</th>
<th>Introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILD_CONFIG</td>
<td>Use same build options as official releases</td>
<td></td>
<td>5.5.7</td>
</tr>
<tr>
<td>CMAKE_BUILD_TYPE</td>
<td>Type of build to produce</td>
<td>RelWithDebInfo</td>
<td>5.5.7</td>
</tr>
<tr>
<td>CMAKE_CXX_FLAGS</td>
<td>Flags for C++ Compiler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMAKE_C_FLAGS</td>
<td>Flags for C Compiler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMAKE_INSTALL_PREFIX</td>
<td>Installation base directory</td>
<td>/usr/local/mysql</td>
<td>5.5.8</td>
</tr>
<tr>
<td>COMPILATION_COMMENT</td>
<td>Comment about compilation environment</td>
<td></td>
<td>5.5.7</td>
</tr>
<tr>
<td>Formats</td>
<td>Description</td>
<td>Default</td>
<td>Introduced</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>CPACK_MONOLITHIC_INSTALL</td>
<td>Whether package build produces single file</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>DEFAULT_CHARSET</td>
<td>The default server character set</td>
<td>latin1</td>
<td>5.5.7</td>
</tr>
<tr>
<td>DEFAULT_COLLATION</td>
<td>The default server collation</td>
<td>latin1_swedish_ci</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLED_LOCAL_INFILE</td>
<td>Whether to enable LOCAL for LOAD DATA INFILE</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLED_PROFILING</td>
<td>Whether to enable query profiling code</td>
<td>ON</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLE_DEBUG_SYNC</td>
<td>Whether to enable Debug Sync support</td>
<td>ON</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLE_DOWNLOADS</td>
<td>Whether to download optional files</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLE_DTRACE</td>
<td>Whether to include DTrace support</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ENABLE_GCOV</td>
<td>Whether to include gcov support</td>
<td>OFF</td>
<td>5.5.14</td>
</tr>
<tr>
<td>IGNORE_AIO_CHECK</td>
<td>With DBUILD_CONFIG=mysql_release, ignore libaio check</td>
<td>OFF</td>
<td>5.5.9</td>
</tr>
<tr>
<td>INSTALL_BINDIR</td>
<td>User executables directory</td>
<td>PREFIX/bin</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_DOCDIR</td>
<td>Documentation directory</td>
<td>PREFIX/docs</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_DOCREADMEDIR</td>
<td>README file directory</td>
<td>PREFIX</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_INCLUDEDIR</td>
<td>Header file directory</td>
<td>PREFIX/include</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_INFODIR</td>
<td>Info file directory</td>
<td>PREFIX/docs</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_LAYOUT</td>
<td>Select predefined installation layout</td>
<td>STANDALONE</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_LIBDIR</td>
<td>Library file directory</td>
<td>PREFIX/lib</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_MANDIR</td>
<td>Manual page directory</td>
<td>PREFIX/man</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_MYSQLSHAREDIR</td>
<td>Shared data directory</td>
<td>PREFIX/share</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_MYSQLTESTDIR</td>
<td>mysql-test directory</td>
<td>PREFIX/mysql-test</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_PLUGINDIR</td>
<td>Plugin directory</td>
<td>PREFIX/lib/plugin</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_SBINDIR</td>
<td>Server executable directory</td>
<td>PREFIX/bin</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_SCRIPTDIR</td>
<td>Scripts directory</td>
<td>PREFIX/scripts</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_SHAREDIR</td>
<td>aclocal/mysql.m4 installation directory</td>
<td>PREFIX/share</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_SQLBENCHDIR</td>
<td>sql-bench directory</td>
<td>PREFIX</td>
<td>5.5.7</td>
</tr>
<tr>
<td>INSTALL_SUPPORTFILESDIR</td>
<td>Extra support files directory</td>
<td>PREFIX/support-files</td>
<td>5.5.7</td>
</tr>
<tr>
<td>MEMCACHED_HOME</td>
<td>Path to memcached</td>
<td>[none]</td>
<td>5.5.16-ndb-7.2.2</td>
</tr>
<tr>
<td>MYSQL_DATADIR</td>
<td>Data directory</td>
<td></td>
<td>5.5.7</td>
</tr>
<tr>
<td>Formats</td>
<td>Description</td>
<td>Default</td>
<td>Introduced</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MYSQL_MAINTAINER_MODE</td>
<td>Whether to enable MySQL maintainer-specific development environment</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>MYSQL_PROJECT_NAME</td>
<td>Windows/OS X project name</td>
<td>3306</td>
<td>5.5.21</td>
</tr>
<tr>
<td>MYSQL_TCP_PORT</td>
<td>TCP/IP port number</td>
<td>3306</td>
<td>5.5.7</td>
</tr>
<tr>
<td>MYSQL_UNIX_ADDR</td>
<td>Unix socket file</td>
<td>/tmp/mysql.sock</td>
<td>5.5.7</td>
</tr>
<tr>
<td>ODBC_INCLUDES</td>
<td>ODBC includes directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ODBC_LIB_DIR</td>
<td>ODBC library directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSCONFDIR</td>
<td>Option file directory</td>
<td></td>
<td>5.5.7</td>
</tr>
<tr>
<td>TMPDIR</td>
<td>tmpdir default value</td>
<td></td>
<td>5.5.36</td>
</tr>
<tr>
<td>WITHOUT_SERVER</td>
<td>Do not build the server</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>WITHOUT_xxx_STORAGE_ENGINE</td>
<td>Exclude storage engine xxx from build</td>
<td></td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_ASAN</td>
<td>Enable AddressSanitizer</td>
<td>OFF</td>
<td>5.5.35</td>
</tr>
<tr>
<td>WITH_BUNDLED_LIBEVENT</td>
<td>Use bundled libevent when building ndbmemcache</td>
<td>ON</td>
<td>5.5.16-ndb-7.2.2</td>
</tr>
<tr>
<td>WITH_BUNDLED_MEMCACHED</td>
<td>Use bundled memcached when building ndbmemcache</td>
<td>ON</td>
<td>5.5.16-ndb-7.2.2</td>
</tr>
<tr>
<td>WITH_CLASSPATH</td>
<td>Classpath to use when building MySQL Cluster Connector for Java. Default is an empty string.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WITH_DEBUG</td>
<td>Whether to include debugging support</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_EMBEDDED_SERVER</td>
<td>Whether to build embedded server</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_EMBEDDED_SHARED_LIBRARY</td>
<td>Whether to build a shared embedded server library</td>
<td>OFF</td>
<td>5.5.37</td>
</tr>
<tr>
<td>WITH_ERROR_INSERT</td>
<td>Enable error injection in the NDB storage engine. Should not be used for building binaries intended for production.</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>WITH_EXTRA_CHARSETS</td>
<td>Which extra character sets to include</td>
<td>all</td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_LIBEDIT</td>
<td>Use bundled libedit library</td>
<td>ON</td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_LIBWRAP</td>
<td>Whether to include libwrap (TCP wrappers) support</td>
<td>OFF</td>
<td>5.5.7</td>
</tr>
<tr>
<td>WITH_NDBCLUSTER</td>
<td>Build the NDB storage engine; alias for WITH_NDBCLUSTER_STORAGE_ENGINE</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>WITH_NDBCLUSTER_STORAGE_ENGINE</td>
<td>Build the NDB storage engine</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>WITH_NDBMTID</td>
<td>Build multi-threaded data node.</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>WITH_NDB_BINLOG</td>
<td>Enable binary logging by default by mysqld</td>
<td>ON</td>
<td></td>
</tr>
</tbody>
</table>
The following sections provide more information about CMake options.

- **General Options**
- **Installation Layout Options**
- **Feature Options**
- **CMake Options for Compiling MySQL Cluster**
- **Compiler Flags**

For boolean options, the value may be specified as 1 or ON to enable the option, or as 0 or OFF to disable the option.

Many options configure compile-time defaults that can be overridden at server startup. For example, the CMAKE_INSTALL_PREFIX, MYSQL_TCP_PORT, and MYSQL_UNIX_ADDR 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. Where applicable, configuration option descriptions indicate the corresponding mysqld startup option.

### General Options

- **-DBUILD_CONFIG=mysql_release**
  
  This option configures a source distribution with the same build options used by Oracle to produce binary distributions for official MySQL releases.

- **-DCMAKE_BUILD_TYPE=type**
  
  The type of build to produce:
Installation Layout Options

- **RelWithDebInfo**: Enable optimizations and generate debugging information. This is the default MySQL build type.

- **Debug**: Disable optimizations and generate debugging information. This build type is also used if the `WITH_DEBUG` option is enabled. That is, `-DWITH_DEBUG=1` has the same effect as `-DCMAKE_BUILD_TYPE=Debug`.

- `-DCPACK_MONOLITHIC_INSTALL=bool`
  
  This option affects whether the `make package` operation produces multiple installation package files or a single file. If disabled, the operation produces multiple installation package files, which may be useful if you want to install only a subset of a full MySQL installation. If enabled, it produces a single file for installing everything.

### Installation Layout Options

The `CMAKE_INSTALL_PREFIX` option indicates the base installation directory. Other options with names of the form `INSTALL_xxx` that indicate component locations are interpreted relative to the prefix and their values are relative pathnames. Their values should not include the prefix.

- `-DCMAKE_INSTALL_PREFIX=dir_name`
  
  The installation base directory.
  
  This value can be set at server startup with the `--basedir` option.

- `-DINSTALL_BINDIR=dir_name`
  
  Where to install user programs.

- `-DINSTALL_DOCDIR=dir_name`
  
  Where to install documentation.

- `-DINSTALL_DOCREADMEDIR=dir_name`
  
  Where to install README files.

- `-DINSTALL_INCLUDEDIR=dir_name`
  
  Where to install header files.

- `-DINSTALL_INFODIR=dir_name`
  
  Where to install Info files.

- `-DINSTALL_LAYOUT=name`
  
  Select a predefined installation layout:

  - **STANDALONE**: Same layout as used for `.tar.gz` and `.zip` packages. This is the default.
  
  - **RPM**: Layout similar to RPM packages.

  - **SVR4**: Solaris package layout.

  - **DEB**: DEB package layout (experimental).
You can select a predefined layout but modify individual component installation locations by specifying other options. For example:

```shell
  cmake . -DINSTALL_LAYOUT=SVR4 -DMYSQL_DATADIR=/var/mysql/data
```

- **-DINSTALL_LIBDIR=dir_name**
  Where to install library files.

- **-DINSTALL_MANDIR=dir_name**
  Where to install manual pages.

- **-DINSTALL_MYSQLSHAREDIR=dir_name**
  Where to install shared data files.

- **-DINSTALL_MYSQLTESTDIR=dir_name**
  Where to install the `mysql-test` directory. As of MySQL 5.5.32, to suppress installation of this directory, explicitly set the option to the empty value (**-DINSTALL_MYSQLTESTDIR=**).

- **-DINSTALL_PLUGINDIR=dir_name**
  The location of the plugin directory.
  This value can be set at server startup with the **--plugin_dir** option.

- **-DINSTALL_SBINDDIR=dir_name**
  Where to install the `mysqld` server.

- **-DINSTALL_SCRIPTDIR=dir_name**
  Where to install `mysql_install_db`.

- **-DINSTALL_SHAREDDIR=dir_name**
  Where to install `aclocal/mysql.m4`.

- **-DINSTALL_SQLBENCHDIR=dir_name**
  Where to install the `sql-bench` directory. To suppress installation of this directory, explicitly set the option to the empty value (**-DINSTALL_SQLBENCHDIR=**).

- **-DINSTALL_SUPPORTFILES=dir_name**
  Where to install extra support files.

- **-DMYSQL_DATADIR=dir_name**
  The location of the MySQL data directory.
  This value can be set at server startup with the **--datadir** option.

- **-DODBC_INCLUDES=dir_name**
  The location of the ODBC includes directory, and may be used while configuring Connector/ODBC.
Storage Engine Options

- **-DODBC_LIB_DIR=dir_name**
  
The location of the ODBC library directory, and may be used while configuring Connector/ODBC.

- **-DSYSCONFDIR=dir_name**
  
The default my.cnf option file directory.
  
  This location cannot be set at server startup, but you can start the server with a given option file using the **--defaults-file=file_name** option, where *file_name* is the full path name to the file.

- **-DTMPDIR=dir_name**
  
The default location to use for the tmpdir system variable. If unspecified, the value defaults to P_tmpdir in <stdio.h>. This option was added in MySQL 5.6.16.

### Storage Engine Options

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

The MyISAM, MERGE, MEMORY, and CSV engines are mandatory (always compiled into the server) and need not be installed explicitly.

To compile a storage engine statically into the server, use **-DWITH_engine_STORAGE_ENGINE=1**. Some permissible engine values are ARCHIVE, BLACKHOLE, EXAMPLE, FEDERATED, INNODB (InnoDB), NDBCLUSTER (NDB), PARTITION (partitioning support), and PERFSCHEMA (Performance Schema).

Examples:

```
-DWITH_INNODB_STORAGE_ENGINE=1
-DWITH_ARCHIVE_STORAGE_ENGINE=1
-DWITH_BLACKHOLE_STORAGE_ENGINE=1
-DWITH_PERFSCHEMA_STORAGE_ENGINE=1
```

**Note**

WITH_NDBCLUSTER_STORAGE_ENGINE is supported only when building MySQL Cluster using the MySQL Cluster sources. It cannot be used to enable clustering support in other MySQL source trees or distributions. In MySQL Cluster NDB 7.2 source distributions, it is enabled by default. See [Building MySQL Cluster from Source on Linux](#), and [Compiling and Installing MySQL Cluster from Source on Windows](#), for more information.

To exclude a storage engine from the build, use **-DWITHOUT_engine_STORAGE_ENGINE=1**. Examples:

```
-DWITHOUT_EXAMPLE_STORAGE_ENGINE=1
-DWITHOUT_FEDERATED_STORAGE_ENGINE=1
-DWITHOUT_PARTITION_STORAGE_ENGINE=1
```

If neither **-DWITH_engine_STORAGE_ENGINE** nor **-DWITHOUT_engine_STORAGE_ENGINE** are specified for a given storage engine, the engine is built as a shared module, or excluded if it cannot be built as a shared module.

### Feature Options

- **-DCOMPILATION_COMMENT=string**
A descriptive comment about the compilation environment.

- **-DDEFAULT_CHARSET=charset_name**

The server character set. By default, MySQL uses the latin1 (cp1252 West European) character set. 

`charset_name` may be one of binary, armSCII8, ascii, big5, cp1250, cp1251, cp1256, cp1257, cp850, cp852, cp866, cp932, dec8, eucjms, euckr, gb2312, gbk, geostd8, greek, hebrew, hp8, keybcs2, koi8r, koi8u, latin1, latin2, latin5, latin7, macce, macroman, sjis, swe7, tis620, ucs2, ujis, utf8, utf8mb4, utf16, utf32. The permissible character sets are listed in the cmake/character_sets.cmake file as the value of CHARSETS_AVAILABLE.

This value can be set at server startup with the --character_set_server option.

- **-DDEFAULT_COLLATION=collation_name**

The server collation. By default, MySQL uses latin1_swedish_ci. Use the SHOW COLLATION statement to determine which collations are available for each character set.

This value can be set at server startup with the --collation_server option.

- **-DENABLE_DEBUG_SYNC=bool**

Whether to compile the Debug Sync facility into the server. This facility is used for testing and debugging. This option is enabled by default, but has no effect unless MySQL is configured with debugging enabled. If debugging is enabled and you want to disable Debug Sync, use -DENABLE_DEBUG_SYNC=0.

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.

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

- **-DENABLE_DOWNLOADS=bool**

Whether to download optional files. For example, with this option enabled, CMake downloads the Google Test distribution that is used by the test suite to run unit tests.

- **-DENABLE_DTRACE=bool**

Whether to include support for DTrace probes. For information about DTrace, see Tracing mysqld Using DTrace.

- **-DENABLE_GCOV=bool**

Whether to include gcov support (Linux only).

- **-DENABLED_LOCAL_INFILE=bool**

Whether to enable LOCAL capability in the client library for LOAD DATA INFILE.

This option controls client-side LOCAL capability, but the capability can be set on the server side at server startup with the --local-infile option. See Security Issues with LOAD DATA LOCAL.

- **-DENABLED_PROFILING=bool**
Feature Options

Whether to enable query profiling code (for the `SHOW PROFILE` and `SHOW PROFILES` statements).

- `-DIGNORE_AIO_CHECK=bool`

  If the `-DBUILD_CONFIG=mysql_release` option is given on Linux, the `libaio` library must be linked in by default. If you do not have `libaio` or do not want to install it, you can suppress the check for it by specifying `-DIGNORE_AIO_CHECK=1`. This option was added in MySQL 5.5.9.

- `-DMYSQL_MAINTAINER_MODE=bool`

  Whether to enable a MySQL maintainer-specific development environment. If enabled, this option causes compiler warnings to become errors.

- `-DMYSQL_PROJECT_NAME=name`

  For Windows or OS X, the project name to incorporate into the project file name. This option was added in MySQL 5.5.21.

- `-DMYSQL_TCP_PORT=port_num`

  The port number on which the server listens for TCP/IP connections. The default is 3306.
  
  This value can be set at server startup with the `--port` option.

- `-DMYSQL_UNIX_ADDR=file_name`

  The Unix socket file path on which the server listens for socket connections. This must be an absolute path name. The default is `/tmp/mysql.sock`.
  
  This value can be set at server startup with the `--socket` option.

- `-DWITH_ASAN=bool`

  Whether to enable AddressSanitizer, for compilers that support it. The default is off. This option was added in MySQL 5.5.35.

- `-DWITH_DEBUG=bool`

  Whether to include debugging support.
  
  Configuring 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.

- `-DWITH_EMBEDDED_SERVER=bool`

  Whether to build the `libmysqld` embedded server library.

- `-DWITH_EMBEDDED_SHARED_LIBRARY=bool`

  Whether to build a shared `libmysqld` embedded server library. This option was added in MySQL 5.5.37.

- `-DWITH_EXTRA_CHARSETS=name`

  Which extra character sets to include:
Feature Options

• **all**: All character sets. This is the default.
• **complex**: Complex character sets.
• **none**: No extra character sets.

• **-DWITH_LIBEDIT=bool**
Whether to use the *libedit* library bundled with the distribution.

• **-DWITH_LIBWRAP=bool**
Whether to include *libwrap* (TCP wrappers) support.

• **-DWITH_READLINE=bool**
Whether to use the *readline* library bundled with the distribution.

• **-DWITH_SSL=ssl_type**
The type of SSL support to include, if any:
  • **no**: No SSL support. This is the default.
  • **yes**: Use the system SSL library if present, else the library bundled with the distribution.
  • **bundled**: Use the SSL library bundled with the distribution.
  • **system**: Use the system SSL library.

For information about using SSL support, see Using Secure Connections.

• **-DWITH_UNIXODBC=I**
Enables unixODBC support, for Connector/ODBC.

• **-DWITH_VALGRIND=bool**
Whether to compile in the Valgrind header files, which exposes the Valgrind API to MySQL code. The default is OFF.

To generate a Valgrind-aware debug build, **-DWITH_VALGRIND=1** normally is combined with **-DWITH_DEBUG=1**. See Building Debug Configurations.

• **-DWITH_ZLIB=zlib_type**
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** indicates the source of *zlib* support:
  • **bundled**: Use the *zlib* library bundled with the distribution.
  • **system**: Use the system *zlib* library. This is the default.

• **-DWITHOUT_SERVER=bool**
Whether to build without the MySQL server. The default is OFF, which does build the server.
Compiler Flags

- **DCMAKE_C_FLAGS="flags"**
  
  Flags for the C Compiler.

- **DCMAKE_CXX_FLAGS="flags"**
  
  Flags for the C++ Compiler.

To specify your own C and C++ compiler flags, for flags that do not affect optimization, use the `CMAKE_C_FLAGS` and `CMAKE_CXX_FLAGS` CMake options.

When providing your own compiler flags, you might want to specify `CMAKE_BUILD_TYPE` as well.

For example, to create a 32-bit release build on a 64-bit Linux machine, do this:

```
shell> mkdir bld
shell> cd bld
shell> cmake .. -DCMAKE_C_FLAGS=-m32 \
     -DCMAKE_CXX_FLAGS=-m32 \
     -DCMAKE_BUILD_TYPE=RelWithDebInfo
```

If you set flags that affect optimization (`-O`), you must set the `CMAKE_C_FLAGS_build_type` and/or `CMAKE_CXX_FLAGS_build_type` options, where `build_type` corresponds to the `CMAKE_BUILD_TYPE` value. To specify a different optimization for the default build type (`RelWithDebInfo`) set the `CMAKE_C_FLAGS_RELWITHDEBINFO` and `CMAKE_CXX_FLAGS_RELWITHDEBINFO` options. For example, to compile on Linux with `-O3` and with debug symbols, do this:

```
shell> cmake .. -DCMAKE_C_FLAGS_RELWITHDEBINFO="-O3 -g" \
     -DCMAKE_CXX_FLAGS_RELWITHDEBINFO="-O3 -g"
```

CMake Options for Compiling MySQL Cluster

The following options are for use when building MySQL Cluster NDB 7.2 or later. These options are supported only with the MySQL Cluster NDB 7.2 and later MySQL Cluster sources; they are not supported when using sources from the MySQL 5.5 Server tree.

- **DMEMCACHED_HOME=dir_name**
  
  Perform the build using the memcached (version 1.6 or later) installed in the system directory indicated by `dir_name`. Files from this installation that are used in the build include the memcached binary, header files, and libraries, as well as the `memcached_utilities` library and the header file `engine_testapp.h`.

  You must leave this option unset when building `ndbmemcache` using the bundled memcached sources (`WITH_BUNDLED_MEMCACHED` option); in other words, the bundled sources are used by default).

  This option was added in MySQL Cluster NDB 7.2.2.

  While additional CMake options—such as for SASL authorization and for providing dtrace support—are available for use when compiling `memcached` from external sources, these options are currently not enabled for the `memcached` sources bundled with MySQL Cluster.

- **WITH_BUNDLED_LIBEVENT={ON|OFF}**
Use the **libevent** included in the MySQL Cluster sources when building MySQL Cluster with ndbmemcached support (MySQL Cluster NDB 7.2.2 and later). Enabled by default. OFF causes the system's libevent to be used instead.

- **-DWITH_BUNDLED_MEMCACHED={ON|OFF}**

Build the memcached sources included in the MySQL Cluster source tree (MySQL Cluster NDB 7.2.3 and later), then use the resulting memcached server when building the ndbmemcache engine. In this case, `make install` places the `memcached` binary in the installation `bin` directory, and the ndbmemcache engine shared object file `ndb_engine.so` in the installation `lib` directory.

This option is ON by default.

- **-DWITH_CLASSPATH=path**

Sets the classpath for building MySQL Cluster Connector for Java. The default is empty. In MySQL Cluster NDB 7.2.9 and later, this option is ignored if `-DWITH_NDB_JAVA=OFF` is used.

- **-DWITH_ERROR_INSERT={ON|OFF}**

Enables error injection in the NDB kernel. For testing only; not intended for use in building production binaries. The default is OFF.

- **-DWITH_NDBCLUSTER_STORAGE_ENGINE={ON|OFF}**

Build and link in support for the NDB (NDBCLUSTER) storage engine in `mysqld`. The default is ON.

- **-DWITH_NDBCLUSTER={ON|OFF}**

This is an alias for `WITH_NDBCLUSTER_STORAGE_ENGINE`.

- **-DWITH_NDBMTD={ON|OFF}**

Build the multi-threaded data node executable `ndbmtd`. The default is ON.

- **-DWITH_NDB_BINLOG={ON|OFF}**

Enable binary logging by default in the `mysqld` built using this option. ON by default.

- **-DWITH_NDB_DEBUG={ON|OFF}**

Enable building the debug versions of the MySQL Cluster binaries. OFF by default.

- **-DWITH_NDB_JAVA={ON|OFF}**

Enable building MySQL Cluster with Java support, including ClusterJ.

This option was added in MySQL Cluster NDB 7.2.9, and is ON by default. If you do not wish to compile MySQL Cluster with Java support, you must disable it explicitly by specifying `-DWITH_NDB_JAVA=OFF` when running CMake. Otherwise, if Java cannot be found, configuration of the build fails.

- **-DWITH_NDB_PORT=port**

Causes the MySQL Cluster management server (`ndb_mgmd`) that is built to use this `port` by default. If this option is unset, the resulting management server tries to use port 1186 by default.

- **-DWITH_NDB_TEST={ON|OFF}**
Dealing with Problems Compiling MySQL

If enabled, include a set of NDB API test programs. The default is OFF.

4.5 Dealing with Problems Compiling MySQL

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

- If CMake is run after it has previously been run, it may use information that was gathered during its previous invocation. This information is stored in CMakeCache.txt. When CMake 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 CMake, 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 object files or configuration information from being used, run the following commands before re-running CMake:

On Unix:

shell> make clean
shell> rm CMakeCache.txt

On Windows:

shell> devenv MySQL.sln /clean
shell> del CMakeCache.txt

If you build outside of the source tree, remove and recreate your build directory before re-running CMake. For instructions on building outside of the source tree, see How to Build MySQL Server with CMake.

On some systems, warnings may occur due to differences in system include files. The following list describes other problems that have been found to occur most often when compiling MySQL:

- To define which C and C++ compilers to use, you can define the CC and CXX environment variables. For example:

shell> CC=gcc
shell> CXX=g++
shell> export CC CXX

To specify your own C and C++ compiler flags, use the CMAKE_C_FLAGS and CMAKE_CXX_FLAGS CMake options. See Compiler Flags.

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

- To see what commands are executed during the compile stage, after using CMake to configure MySQL, run make VERBOSE=1 rather than just make.

- If compilation fails, check whether the MYSQL_MAINTAINER_MODE option is enabled. This mode causes compiler warnings to become errors, so disabling it may enable compilation to proceed.

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

"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 a recent version of bison (the GNU version of yacc) and use that instead.

Versions of bison older than 1.75 may report this error:

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.

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

4.6 MySQL Configuration and Third-Party Tools

Third-party tools that need to determine the MySQL version from the MySQL source can read the VERSION file in the top-level source directory. The file lists the pieces of the version separately. For example, if the version is MySQL 5.7.4-m14, the file looks like this:

MYSQL_VERSION_MAJOR=5
MYSQL_VERSION_MINOR=7
MYSQL_VERSION_PATCH=4
MYSQL_VERSION_EXTRA=-m14

If the source is not for a General Availability (GA) release, the MYSQL_VERSION_EXTRA value will be nonempty. For the example, the value corresponds to Milestone 14.

To construct a five-digit number from the version components, use this formula:

MYSQL_VERSION_MAJOR*10000 + MYSQL_VERSION_MINOR*100 + MYSQL_VERSION_PATCH
Chapter 5 Installing MySQL on Microsoft Windows

**Table of Contents**

5.1 MySQL Installation Layout on Microsoft Windows .......................................................... 46
5.2 Choosing An Installation Package .................................................................................. 46
5.3 Installing MySQL on Microsoft Windows Using MySQL Installer .................................... 47
   5.3.1 MySQL Installer GUI ................................................................................................. 49
   5.3.2 MySQL Installer Console ......................................................................................... 73
5.4 MySQL Notifier .................................................................................................................. 76
   5.4.1 Remote monitoring set up and installation instructions .............................................. 83
5.5 Installing MySQL on Microsoft Windows Using an MSI Package ................................... 88
   5.5.1 Using the MySQL Installation Wizard ...................................................................... 90
   5.5.2 Automating MySQL Installation on Microsoft Windows Using the MSI Package .......... 93
   5.5.3 Removing MySQL When Installed from the MSI Package ....................................... 94
5.6 MySQL Server Instance Configuration Wizard ............................................................... 95
   5.6.1 Starting the MySQL Server Instance Configuration Wizard ..................................... 95
   5.6.2 Choosing a Maintenance Option ............................................................................... 96
   5.6.3 Choosing a Configuration Type ................................................................................. 96
   5.6.4 The Server Type Dialog ........................................................................................... 97
   5.6.5 The Database Usage Dialog .................................................................................... 99
   5.6.6 The InnoDB Tablespace Dialog ............................................................................... 100
   5.6.7 The Concurrent Connections Dialog ........................................................................ 101
   5.6.8 The Networking and Strict Mode Options Dialog ..................................................... 102
   5.6.9 The Character Set Dialog ....................................................................................... 103
   5.6.10 The Service Options Dialog .................................................................................. 103
   5.6.11 The Security Options Dialog .................................................................................. 104
   5.6.12 The Confirmation Dialog ....................................................................................... 106
   5.6.13 MySQL Server Instance Config Wizard: Creating an Instance from the Command Line .... 107
5.7 Installing MySQL on Microsoft Windows Using a noinstall Zip Archive ......................... 110
   5.7.1 Extracting the Install Archive ................................................................................... 110
   5.7.2 Creating an Option File ............................................................................................ 111
   5.7.3 Selecting a MySQL Server Type .............................................................................. 112
   5.7.4 Starting the Server for the First Time ...................................................................... 112
   5.7.5 Starting MySQL from the Windows Command Line ............................................... 114
   5.7.6 Customizing the PATH for MySQL Tools ................................................................ 114
   5.7.7 Starting MySQL as a Windows Service .................................................................... 115
   5.7.8 Testing The MySQL Installation ............................................................................. 118
5.8 Troubleshooting a Microsoft Windows MySQL Server Installation ................................. 118
5.9 Windows Postinstallation Procedures ............................................................................. 120
5.10 Upgrading MySQL on Windows .................................................................................... 122

There are several different methods to install MySQL on Microsoft Windows.

**Simple Installation Method**

The simplest and recommended method is to download MySQL Installer (for Windows) and let it install and configure all of the MySQL products on your system. Here is how:

Note

Unlike the standard MySQL Installer, the smaller "web-community" version does not bundle any MySQL applications but it will download the MySQL products you choose to install.

- Choose the appropriate Setup Type for your system. Typically you will choose Developer Default to install MySQL server and other MySQL tools related to MySQL development, helpful tools like MySQL Workbench. Or, choose the Custom setup type to manually select your desired MySQL products.

Note

Multiple versions of MySQL server can exist on a single system. You can choose one or multiple versions.

- Complete the installation process by following the MySQL Installation wizard's instructions. This will install several MySQL products and start the MySQL server.

- MySQL is now installed. You probably configured MySQL as a service that will automatically start MySQL server every time you restart your system.

Note

You probably also installed other helpful MySQL products like MySQL Workbench and MySQL Notifier on your system. Consider loading MySQL Workbench to check your new MySQL server connection, and Section 5.4, “MySQL Notifier” to view the connection's status. By default, these two programs automatically start after installing MySQL.

This process also installs the MySQL Installer application on your system, and later you can use MySQL Installer to upgrade or reconfigure your MySQL products.

Additional Installation Information

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 5.7.7, “Starting MySQL as a 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.
Additional Installation Information

Note
MySQL Installer will install and configure Connector/ODBC for you.

• 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.

Note
MySQL Installer will install and configure Connector/NET for you.


MySQL for Windows is available in several distribution formats, detailed here. Generally speaking, you should use MySQL Installer. It contains more features and MySQL products than the older MSI, is simpler to use than the Zip file, and you need no additional tools to get MySQL up and running. MySQL Installer automatically installs MySQL Server and additional MySQL products, creates an options file, starts the server, and enables you to create default user accounts. For more information on choosing a package, see Section 5.2, “Choosing An Installation Package”.

• Binary installer distributions. There are two different installable distributions that come packaged as a Microsoft Windows Installer (MSI) package that you can install manually or automatically on your systems. The preferred MySQL Installer package includes MySQL Server and additional MySQL products including MySQL Workbench, MySQL Notifier, and MySQL for Excel. MySQL Installer can also be used to upgrade these products in the future. The older MSI package contains all the files you need to install and configure MySQL server, but no additional components.

For instructions on installing MySQL using MySQL Installer, see Section 5.3, “Installing MySQL on Microsoft Windows Using MySQL Installer”.

• 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 5.7, “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 Chapter 4, Installing MySQL from Source.

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 5.7.2, “Creating an Option File”.

5.1 MySQL Installation Layout on Microsoft Windows

For MySQL 5.5 on Windows, the default installation directory is C:\Program Files\MySQL\MySQL Server 5.5. 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.

All of the files are located within this parent directory, using the structure shown in the following table.

Table 5.1 Default MySQL Installation Layout for Microsoft Windows

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin, scripts</td>
<td>mysqlld server, client and utility programs</td>
<td></td>
</tr>
<tr>
<td>%ALLUSERSPROFILE%\MySQL\MySQL Server 5.5\</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>
<tr>
<td>%PROGRAMDATA%\MySQL\MySQL Server 5.5\</td>
<td>Log files, databases (Vista, Windows 7, Windows Server 2008, and newer)</td>
<td>The Windows system variable %PROGRAMDATA% defaults to C:\ProgramData</td>
</tr>
<tr>
<td>examples</td>
<td>Example programs and scripts</td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>Include (header) files</td>
<td></td>
</tr>
<tr>
<td>lib</td>
<td>Libraries</td>
<td></td>
</tr>
<tr>
<td>share</td>
<td>Miscellaneous support files, including error messages, character set files, sample configuration files, SQL for database installation</td>
<td></td>
</tr>
</tbody>
</table>

If you install MySQL using a Windows MSI package, this package creates and sets up the data directory that the installed server will use, but as of MySQL 5.5.5, it also creates a pristine “template” data directory named data under the installation directory. This directory can be useful when the machine will be used to run multiple instances of MySQL: After an installation has been performed using an MSI package, the template data directory can be copied to set up additional MySQL instances. See Running Multiple MySQL Instances on One Machine.

5.2 Choosing An Installation Package

For MySQL 5.5, there are multiple installation package formats to choose from when installing MySQL on Windows.
5.3 Installing MySQL on Microsoft Windows Using MySQL Installer

MySQL Installer is an application that manages MySQL products on Microsoft Windows. It installs, updates, removes, and configures MySQL products, and remains on the system as its own application. MySQL Installer is only available for Microsoft Windows, and includes both GUI and command-line interfaces.
The supported MySQL products include:

- **MySQL Server** (one or multiple versions on the same system)
- **MySQL Workbench**
- **MySQL Connectors** (.Net / Python / ODBC / Java / C / C++)
- **MySQL Notifier**
- **MySQL for Excel**
- **MySQL for Visual Studio**
- **MySQL Utilities and MySQL Fabric**
- **MySQL Samples and Examples**
- **MySQL Documentation**
- **MySQL Installer** is also installed and remains on the system as its own application, that is used to install additional MySQL products, and also to update and configure existing MySQL products
- **The Enterprise edition installs the Enterprise versions of the above products, and also includes MySQL Enterprise Backup and MySQL Enterprise Firewall**

### Installer package types

- **Full**: Bundles all of the MySQL products (including the MySQL server). The file size is over 300MB, and its name has the form `mysql-installer-community-VERSION.N.msi` where `VERSION` is the MySQL Server version number such as 5.7 and `N` is the package number, which begins at 0.

- **Web**: Only contains the Installer and configuration files, and it downloads the MySQL products you choose to install. The size of this file is about 2MB; the name of the file has the form `mysql-installer-community-web-VERSION.N.msi` where `VERSION` is the MySQL Server version number such as 5.7 and `N` is the package number, which begins at 0.

- **Updates**: MySQL Installer can upgrade itself, so an additional download is not required to update MySQL Installer.

### Installer editions


- **Commercial edition**: Downloadable at either [My Oracle Support](https://education.oracle.com/) or [https://education.oracle.com/](https://education.oracle.com/). It installs the commercial version of all MySQL products, including Workbench SE/EE, MySQL Enterprise Backup, and MySQL Enterprise Firewall. It also integrates with your MOS account.

**Note**

Entering your MOS credentials is optional when installing bundled MySQL products, but your credentials are required when choosing non-bundled MySQL products that MySQL Installer must download.

For notes detailing the changes in each release of MySQL Installer, see [MySQL Installer Release Notes](http://dev.mysql.com/downloads/installer/).
MySQL Installer is compatible with pre-existing installations, and adds them to its list of installed components. While the standard MySQL Installer is bundled with a specific version of MySQL server, a single MySQL Installer instance can install and manage multiple MySQL server versions. For example, a single MySQL Installer instance can install (and update) versions 5.5, 5.6, and 5.7 on the same host.

**Note**

A single host can *not* have both community and commercial editions of MySQL server installed. For example, if you want both MySQL Server 5.6 and 5.7 installed on a single host, both must be the same edition.

MySQL Installer handles the initial configuration and set up of the applications. For example:

1. It creates the configuration file (`my.ini`) that is used to configure the MySQL Server. The values written to this file are influenced by choices you make during the installation process.

   **Note**

   Some definitions are host dependent. For example, query_cache is enabled if the host has fewer than three cores.

2. It can optionally import example databases.

3. By default, a Windows service for the MySQL server is added.

4. It can optionally create MySQL Server user accounts with configurable permissions based on general roles, such as DB Administrator, DB Designer, and Backup Admin. It optionally creates a Windows user named `MysqlSys` with limited privileges, which would then run the MySQL Server.

   User accounts may also be added and configured in MySQL Workbench.

5. Checking *Show Advanced Options* allows additional *Logging Options* to be set. This includes defining custom file paths for the error log, general log, slow query log (including the configuration of seconds it requires to execute a query), and the binary log.

MySQL Installer can optionally check for updated components and download them for you.

### 5.3.1 MySQL Installer GUI

Installing MySQL Installer adds a link to the Start menu under the **MySQL** group. Click **Start, All Programs MySQL, MySQL Installer** to reload the MySQL Installer GUI.

**Note**

Full permissions are granted to the user executing MySQL Installer to all generated files, such as `my.ini`. This does not apply to files and directories for specific products, such as the MySQL server data directory in `%ProgramData%` that is owned by `SYSTEM`.

MySQL Installer requires you to accept the license agreement before it will install MySQL products.
Installing New Packages

Choose the appropriate Setup Type for your system. This type determines which MySQL products are initially installed on your system, or select Custom to manually choose the products.

- **Developer**: Install all products needed to develop applications with MySQL. This is the default option.
- **Server only**: Only install the MySQL server.
- **Client only**: Only install the MySQL client products, such as MySQL Workbench. This does not include the MySQL server.
- **Full**: Install all available MySQL products.
- **Custom**: Manually select the MySQL products to install, and optionally configure custom MySQL data and installation paths.

**Note**

After the initial installation, you may use MySQL Installer to manually select MySQL products to install or remove. In other words, MySQL Installer becomes a MySQL product management system.
MySQL Installer checks your system for the external requirements (pre-requisites) required to install the selected MySQL products. MySQL Installer can download and install some prerequisites, but others require manual intervention. Download and install all prerequisites that have **Status** set to "Manual". Click **Check** to recheck if a manual prerequisite was installed. After manually installing those requirements, click **Execute** to download and install the other prerequisites. Once finished, click **Next** to continue.
Figure 5.3 MySQL Installer - Check Requirements

The next window lists the MySQL products that are scheduled for installation:
As components are installed, their **Status** changes from a progress percentage to "Complete".

After all components are installed, the next step configures some of the recently installed MySQL products. The **Configuration Overview** window displays the progress and then loads a configuration window, if required. Our example configures MySQL Server 5.6.x.

**Configuring MySQL Server**

Configuring the MySQL server begins with defining several **Type and Networking** options.
Server Configuration Type

Choose the MySQL server configuration type that describes your setup. This setting defines the amount of system resources (memory) that will be assigned to your MySQL server instance.

- **Developer**: A machine that will host many other applications, and typically this is your personal workstation. This option configures MySQL to use the least amount of memory.

- **Server**: Several other applications will be running on this machine, such as a web server. This option configures MySQL to use a medium amount of memory.

- **Dedicated**: A machine that is dedicated to running the MySQL server. Because no other major applications will run on this server, such as a web server, this option configures MySQL to use the majority of available memory.

Connectivity

Connectivity options control how the connection to MySQL is made. Options include:

- **TCP/IP**: You may enable TCP/IP Networking here as otherwise only localhost connections are allowed. Also define the **Port Number** and whether to open the firewall port for network access.

- **Named Pipe**: Enable and define the pipe name, similar to using the `--enable-named-pipe` option.
• **Shared Memory**: Enable and then define the memory name, similar to using the `--shared-memory` option.

**Advanced Configuration**

Check **Show Advanced Options** to set additional **Logging Options**. This includes defining custom file paths for the error log, general log, slow query log (including the configuration of seconds it requires to execute a query), and the binary log.

**Figure 5.6 MySQL Installer - MySQL Server Configuration: Type and Networking**

**Accounts and Roles**

Next, define your MySQL account information. Assigning a root password is required.

Optionally, you can add additional MySQL user accounts with predefined user roles. Each predefined role, such as "DB Admin", are configured with their own set of privileges. For example, the "DB Admin" role has more privileges than the "DB Designer" role. Click the **Role** dropdown for a list of role descriptions.

**Note**

If the MySQL Server is already installed, then you must also enter the **Current Root Password**.
Next, configure the **Windows Service** details. This includes the service name, whether the MySQL server should be loaded at startup, and how the MySQL server Windows service is executed.
Figure 5.9 MySQL Installer - MySQL Server Configuration: Windows Service

**Note**

When configuring **Run Windows Services as ...** using a **Custom User**, the custom user must have privileges to log on to Microsoft Windows as a service. The **Next** button will be disabled until this user is configured with the required privileges.

On Microsoft Windows 7, this is configured by loading the **Start Menu, Control Panel, Administrative Tools, Local Security Policy, Local Policies, User Rights Assignment**, then **Log On As A Service**. Choose **Add User or Group** here to add the custom user, and then **OK, OK** to save.

**Advanced Options**

The next configuration step is available if the **Advanced Configuration** option was checked. This section includes options that are related to the MySQL log files:
Figure 5.10 MySQL Installer - MySQL Server Configuration: Logging Options

Click Next to continue on to the final page before all of the requested changes are applied. This Apply Server Configuration page details the configuration steps that will be performed.
Click **Execute** to execute the configuration steps. The icon for each step toggles from white to green on success, or the process stops on failure. Click the **Log** tab to view the log.

After the MySQL Installer configuration process is finished, MySQL Installer reloads the opening page where you can execute other installation and configuration related actions.

MySQL Installer is added to the Microsoft Windows Start menu under the **MySQL** group. Opening MySQL Installer loads its dashboard where installed MySQL products are listed, and other MySQL Installer actions are available:
Adding MySQL Products

Click Add to add new products. This loads the **Select Products and Features** page:
From here, choose the MySQL products you want to install from the left Available Products pane, and then click the green right arrow to queue products for installation.

Optionally, click Edit to open the product and features search filter:
For example, you might choose to include Pre-Release products in your selections, such as a Beta product that has not yet reached General Availability (GA) status.

Select all of the MySQL products you want to install, then click Next to continue using the defaults, or highlight a selected product and click Advanced Options to optionally alter options such as the MySQL server data and installation paths. Click Execute to execute the installation process to install all of the selected products.

5.3.1.1 MySQL Product Catalog

MySQL Installer stores a MySQL product catalog. The catalog can be updated either manually or automatically, and the catalog change history is also available. The automatic update is enabled by default.

Note

The product catalog update also checks for a newer version of MySQL Installer, and prompts for an update if one is present.

Manual updates

You can update the MySQL product catalog at any time by clicking Catalog on the Installer dashboard.
Figure 5.15 MySQL Installer - Open the MySQL Product Catalog

From there, click Execute to update the product catalog.

**Automatic updates**

MySQL Installer can automatically update the MySQL product catalog. By default, this feature is enabled to execute each day at 12:00 AM. To configure this feature, click the wrench icon on the Installer dashboard.

The next window configures the *Automatic Catalog Update*. Enable or disable this feature, and also set the hour.
Figure 5.16 MySQL Installer - Configure the Catalog Scheduler

This option uses the Windows Task Scheduler to schedule a task named "ManifestUpdate".

Change History

MySQL Installer tracks the change history for all of the MySQL products. Click Catalog from the dashboard, optionally update the catalog (or, toggle the Do not update at this time checkbox), click Next/Execute, and then view the change history.
5.3.1.2 Remove MySQL Products

MySQL Installer can also remove MySQL products from your system. To remove a MySQL product, click **Remove** from the Installer dashboard. This opens a window with a list of installed MySQL products. Select the MySQL products you want to remove (uninstall), and then click **Execute** to begin the removal process.

**Note**

To select all MySQL products, click the [] checkbox to the left of the **Product** label.
Figure 5.18 MySQL Installer - Removing Products: Select

Click [Execute] to remove the selected packages.
5.3.1.3 Alter MySQL Products

Use MySQL Installer to modify, configure, or upgrade your MySQL product installations.

**Upgrade**

Upgradable MySQL products are listed on the main dashboard with an arrow icon (👉) next to their version number.
Figure 5.20 MySQL Installer - Upgrade a MySQL Product

Click Upgrade to upgrade the available products. Our example indicates that MySQL Workbench 6.2.4 can be upgraded version 6.3.1 or 6.2.5, and MySQL server from 5.5.41 to 5.5.42.

Note

The "upgrade" functionality requires a current product catalog. This catalog is updated either manually or automatically (daily) by enabling the Automatic Catalog Update feature. For additional information, see Section 5.3.1.1, "MySQL Product Catalog".
Figure 5.21 MySQL Installer - Select Products To Upgrade

If multiple upgrade versions are available (such as our MySQL Workbench example above), select the desired version for the upgrade in the **Available Upgrades** area.

**Note**

Optionally, click the **Changes** link to view the version's release notes.

After selecting (checking) the products and versions to upgrade, click **Next** to begin the upgrade process.
A MySQL server upgrade will also check and upgrade the server's database. Although optional, this step is recommended.
Figure 5.23 MySQL Installer - Check and Upgrade Database

Upon completion, your upgraded products will be upgraded and available to use. A MySQL server upgrade also restarts the MySQL server.

Reconfigure

Some MySQL products, such as the MySQL server, include a Reconfigure option. It opens the same configuration options that were set when the MySQL product was installed, and is pre-populated with the current values.

To execute, click the Reconfigure link under the Quick Action column on the main dashboard for the MySQL product that you want to reconfigure.
In the case of the MySQL server, this opens a configuration wizard that relates to the selected product. For example, for MySQL Server this includes setting the type, ports, log paths, and so on.

**Modify**

Many MySQL products contain feature components that can be added or removed. For example, Debug binaries and Client Programs are subcomponents of the MySQL server.

The modify the features of a product, click **Modify** on the main dashboard.
MySQL Installer Console

5.3.2 MySQL Installer Console

MySQL-installer-console provides functionality similar to the GUI version of MySQL Installer, but from the command-line. It is installed when MySQL Installer is initially executed, and then available within the MySQL Installer directory. Typically that is in C:\Program Files (x86)\MySQL\MySQL Installer\, and the console must be executed with administrative privileges.

To use, invoke the Command Prompt with administrative privileges by choosing Start, Accessories, then right-click on Command Prompt and choose Run as administrator. And from the command-line, optionally change the directory to where MySQL-installer-console is located:

C:\> cd "C:\Program Files (x86)\MySQL\MySQL Installer for Windows"
C:\> MySQL-installer-console.exe help

C:\Program Files (x86)\MySQL\MySQL Installer for Windows>MySQL-installer-console.exe help

The following commands are available:

Configure - Configures one or more of your installed programs.
Help - Provides list of available commands.
Install - Install and configure one or more available MySQL programs.
List - Provides an interactive way to list all products available.
Modify - Modifies the features of installed products.

Click Execute to execute the modification request.
MySQL Installer Console

Remove - Removes one or more products from your system.
Status - Shows the status of all installed products.
Update - Update the current product catalog.
Upgrade - Upgrades one or more of your installed programs.

MySQLInstallerConsole supports the following options, which are specified on the command line:

- **configure** [product1]:[setting]=[value]; [product2]:[setting]=[value]; [...]  
  Configure one or more MySQL products on your system. Multiple setting=value pairs can be configured for each product.
  
  Switches include:
  
  - **-showsettings** - Displays the available options for the selected product, by passing in the product name after `-showsettings`.
  - **-silent** - Disable confirmation prompts.

  ```
  C:\> MySQLInstallerConsole configure -showsettings server
  C:\> MySQLInstallerConsole configure server:port=3307
  ```

- **help** [command]  
  Displays a help message with usage examples, and then exits. Pass in an additional command to receive help specific to that command.

  ```
  C:\> MySQLInstallerConsole help
  C:\> MySQLInstallerConsole help install
  ```

- **install** [product]:[features]:[config block]:[config block]:[config block]; [...]  
  Install one or more MySQL products on your system.
  
  Switches and syntax options include:
  
  - **-type=[SetupType]** - Installs a predefined set of software. The "SetupType" can be one of the following:

    - **Developer** - Installs a complete development environment.
    - **Server** - Installs a single MySQL server
    - **Client** - Installs client programs and libraries
    - **Full** - Installs everything
    - **Custom** - Installs user selected products. This is the default option.
MySQL Installer Console

- **-showsettings**: Displays the available options for the selected product, by passing in the product name after `-showsettings`.

- **-silent**: Disable confirmation prompts.

- **[config block]**: One or more configuration blocks can be specified. Each configuration block is a semicolon separated list of key value pairs. A block can include either a "config" or "user" type key, where "config" is the default type if one is not defined.

  Configuration block values that contain a colon (":") must be wrapped in double quotes. For example, `installdir="C:\MySQL\MySQL Server 5.6"`.

  Only one "config" type block can be defined per product. A "user" block should be defined for each user that should be created during the product's installation.

  **Note**
  Adding users is not supported when a product is being reconfigured.

- **[feature]**: The feature block is a semicolon separated list of features, or "*" to select all features.

  ```
  C:\> MySQLInstallerConsole install server;5.6.25:*:port=3307;serverid=2:type=user;username=foo;password=bar;role=DBManager
  C:\> MySQLInstallerConsole install server;5.6.25;x64 -silent
  ```

  An example that passes in additional configuration blocks, broken up by `^` to fit this screen:

  ```
  C:\> MySQLInstallerConsole install server;5.6.25;x64:*:type=config;openfirewall=true;generallog=true;binlog=true;serverid=3306;enable_tcpip=true;port=3306;rootpasswd=pass;^installdir="C:\MySQL\MySQL Server 5.6":type=user;datadir="C:\MySQL\data";username=foo;password=bar
  ```

- **list**

  Lists an interactive console where all of the available MySQL products can be searched. Execute `MySQLInstallerConsole list` to launch the console, and enter in a substring to search.

  ```
  C:\> MySQLInstallerConsole list
  ```

- **modify [product1:-removelist|+addlist] [product2:-removelist|+addlist] [...]**

  Modifies or displays features of a previously installed MySQL product.

  - **-silent**: Disable confirmation prompts.

  ```
  C:\> MySQLInstallerConsole modify server
  C:\> MySQLInstallerConsole modify server:+documentation
  C:\> MySQLInstallerConsole modify server:-debug
  ```

- **remove [product1] [product2] [...]**

  Removes one ore more products from your system.

  - *****: Pass in "*" to remove all of the MySQL products.

  - **-continue**: Continue the operation even if an error occurs.
• **-silent**: Disable confirmation prompts.

```c:
MySQLInstallerConsole remove *
MySQLInstallerConsole remove server
```

**status**

Provides a quick overview of the MySQL products that are installed on the system. Information includes product name and version, architecture, date installed, and install location.

```c:
MySQLInstallerConsole status
```

• **upgrade [product1:version] [product2:version], […]**

Upgrades one or more products on your system. Syntax options include:

• *: Pass in * to upgrade all products to the latest version, or pass in specific products.

• !: Pass in ! as a version number to upgrade the MySQL product to its latest version.

• **-silent**: Disable confirmation prompts.

```c:
MySQLInstallerConsole upgrade *
MySQLInstallerConsole upgrade workbench:6.3.5
MySQLInstallerConsole upgrade workbench:!
MySQLInstallerConsole upgrade workbench:6.3.5 excel:1.3.2
```

• **update**

Downloads the latest MySQL product catalog to your system. On success, the download catalog will be applied the next time either MySQLInstaller or MySQLInstallerConsole is executed.

```c:
MySQLInstallerConsole update
```

**Note**

The **Automatic Catalog Update** GUI option executes this command from the Windows Task Scheduler.

### 5.4 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.

Notice
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.

Notice
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.
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 5.27 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.”
MySQL Notifier

- **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 5.28 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 5.29 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 5.31 MySQL Notifier Adding new services

And instances:
Remote monitoring set up and installation instructions

5.4.1 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.

Note
Remote monitoring is available since MySQL Notifier 1.1.0.

Note
The Instances tab available since MySQL Notifier 1.1.0.
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 5.33 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 GEDIT.MSC, and then click OK.
   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:

   ```plaintext
   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

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:

```bash
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]":

```bash
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.
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.
   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.

### 5.5 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 Complete install package, which is 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 5.5.1, “Using the MySQL Installation Wizard”.

- For information on installing using the command line using the MSI package, see Section 5.5.2, “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 5.5.3, “Removing MySQL When Installed from the MSI Package”.

The workflow sequence for using the installer is shown in the figure below:
Figure 5.34 Installation Workflow for Windows Using MSI Installer

Note

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.

### 5.5.1 Using the MySQL Installation Wizard

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 Configuration Wizard, enables a user to install and configure a MySQL server that is ready for use immediately after installation.

The MySQL Installation Wizard is the standard installer for all MySQL server distributions, version 4.1.5 and higher. Users of previous versions of MySQL need to shut down and remove their existing MySQL installations manually before installing MySQL with the MySQL Installation Wizard. See Section 5.5.1.6, “Upgrading MySQL with the Installation Wizard”, for more information on upgrading from a previous version.

Microsoft has included an improved version of their Microsoft Windows Installer (MSI) in the recent versions of Windows. MSI has become the de-facto standard for application installations on Windows 2000, Windows XP, and Windows Server 2003. The MySQL Installation Wizard makes use of this technology to provide a smoother and more flexible installation process.

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 recently. 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.

### 5.5.1.1 Downloading and Starting the MySQL Installation Wizard

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

**Note**

If you are installing on Windows Vista or newer, it is best to open a network port before beginning the installation. To do this, first ensure that you are logged in as an Administrator, go to the Control Panel, and double-click the Windows Firewall icon. Choose the Allow a program through Windows Firewall option and click the Add port button. Enter MySQL into the Name text box and 3306 (or the port of your choice) into the Port number text box. Also ensure that the TCP protocol radio button is selected. If you wish, you can also limit access to the MySQL server by choosing the Change scope button. Confirm your choices by clicking the OK button. If you do not open a port prior to installation, you cannot configure the MySQL server immediately after installation. Additionally, when running the MySQL Installation Wizard on Windows Vista or newer, ensure that you are logged in as a user with administrative rights.

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.

### 5.5.1.2 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 5.5.1.3, “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 5.5.1.3, “The Custom Install Dialog”.
5.5.1.3 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.

5.5.1.4 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.

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

5.5.1.5 Changes Made by MySQL Installation Wizard

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`.

The MySQL Installation Wizard creates a key named after the release series of the server that is being installed, such as `MySQL Server 5.5`. 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 5.5\`. The **Version** string contains the release number. For example, for an installation of MySQL Server 5.5.48, the key contains a value of `5.5.48`.

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 5.5, the MySQL Installation Wizard creates a MySQL Server 5.5 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 Configuration 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.

Changes to the File System

The MySQL Installation Wizard by default installs the MySQL 5.5 server to `C:\Program Files\MySQL\MySQL Server 5.5`, where `Program Files` is the default location for applications in your system, and `5.5` 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 5.5
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.

The default location of the data directory is the `AppData` directory configured for the user that installed the MySQL application.

5.5.1.6 Upgrading MySQL with the Installation Wizard

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 5.5.5 to MySQL 5.5.6, but not from MySQL 5.1 to MySQL 5.5.

See Section 5.10, “Upgrading MySQL on Windows”.

5.5.2 Automating MySQL Installation on Microsoft Windows Using the MSI Package

The Microsoft Installer (MSI) supports 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 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-5.5.48.msi /quiet
```
Removing MySQL When Installed from the MSI Package

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-5.5.48.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-5.5.48.msi /uninstall
```

To install MySQL and configure a MySQL instance from the command line, see Section 5.6.13, “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.

5.5.3 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 5.1, MySQL 5.5, 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, 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.
5.6 MySQL Server Instance Configuration Wizard

The MySQL Server Instance Configuration 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 MySQL Server Instance Configuration Wizard is included with the MySQL 5.5 server. The MySQL Server Instance Configuration Wizard is only available for Windows.

5.6.1 Starting the MySQL Server Instance Configuration Wizard

The MySQL Server Instance Configuration Wizard is normally started as part of the installation process. You should only need to run the MySQL Server Instance Configuration 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 Configuration Wizard after installation. However, you must open a port in the Windows Firewall. To do this see the instructions given in Section 5.5.1.1, "Downloading and Starting the MySQL Installation Wizard". 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 Configuration Wizard on Windows Vista or newer, ensure that you are logged in as a user with administrative rights.
Choosing a Maintenance Option

You can launch the MySQL Configuration 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 Configuration 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 5.5\my.ini"
```

Here, C:\Program Files\MySQL\MySQL Server 5.5 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 Configuration 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/ 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 Configuration 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.

5.6.2 Choosing a Maintenance Option

If the MySQL Server Instance Configuration 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 Configuration 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.

5.6.3 Choosing a Configuration Type

When you start the MySQL Server Instance Configuration 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 Configuration 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 5.6.10, “The Service Options Dialog”, and Section 5.6.11, “The Security Options Dialog”, respectively.

### 5.6.4 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 Configuration 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 configuration wizard.
The Database Usage Dialog

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

- Multifunctional Database: MIXED
- Transactional Database Only: INNODB
- Non-Transactional Database Only: MYISAM
When these options are processed through the default template (my-template.ini) the result is:

<table>
<thead>
<tr>
<th>Multifunctional Database:</th>
<th>default-storage-engine=InnoDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>_myisam_pct=50</td>
<td></td>
</tr>
<tr>
<td>Transactional Database Only:</td>
<td>default-storage-engine=InnoDB</td>
</tr>
<tr>
<td>_myisam_pct=5</td>
<td></td>
</tr>
<tr>
<td>Non-Transactional Database Only:</td>
<td>default-storage-engine=MyISAM</td>
</tr>
<tr>
<td>_myisam_pct=100</td>
<td>skip-innodb</td>
</tr>
</tbody>
</table>

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

### 5.6.6 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.
The Concurrent Connections Dialog

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.

5.6.7 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: 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.

5.6.8 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.
5.6.9 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.

5.6.10 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 Configuration 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
The Security Options Dialog

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 5.5.0 as mysql50500.

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.

### 5.6.11 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.
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.

5.6.12 The Confirmation Dialog

The final dialog in the MySQL Server Instance Configuration 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 Configuration Wizard without configuring the server, click the Cancel button.
After you click the **Execute** button, the MySQL Server Instance Configuration Wizard performs a series of tasks and displays the progress onscreen as the tasks are performed.

The MySQL Server Instance Configuration 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 Configuration 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 Configuration Wizard creates and starts the service. If you are reconfiguring an existing service, the MySQL Server Instance Configuration Wizard restarts the service to apply your configuration changes.

If you chose to set a **root** password, the MySQL Configuration 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 Configuration Wizard has completed its tasks, it displays a summary. Click the **Finish** button to exit the MySQL Server Configuration Wizard.

**5.6.13 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.
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 to 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 5.2 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 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 5.3 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>AddBinToPath={yes</td>
<td>Specifies whether to add the binary directory of MySQL to the standard PATH environment variable</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
ServerType={DEVELOPMENT | SERVER | DEDICATED} | Specify the server type. For more information, see Section 5.6.4, “The Server Type Dialog”
DatabaseType={MIXED | INNODB | MYISAM} | Specify the default database type. For more information, see Section 5.6.5, “The Database Usage Dialog”
ConnectionUsage={DSS | OLTP} | Specify the type of connection support, this automates the setting for the number of concurrent connections (see the ConnectionCount parameter). For more information, see Section 5.6.7, “The Concurrent Connections Dialog”
ConnectionCount=| Specify the number of concurrent connections to support. For more information, see Section 5.6.4, “The Server Type Dialog”
SkipNetworking={yes | no} | Specify whether network support should be supported. Specifying yes disables network access altogether
Port=# | Specify the network port number to use for network connections. For more information, see Section 5.6.8, “The Networking and Strict Mode Options Dialog”
StrictMode={yes | no} | Specify whether to use the strict SQL mode. For more information, see Section 5.6.8, “The Networking and Strict Mode Options Dialog”
Charset=| Specify the default character set. For more information, see Section 5.6.9, “The Character Set Dialog”
RootPassword=| Specify the root password
RootCurrentPassword= | Specify the current root password then stopping or reconfiguring an existing service

**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, "-cC:\mysql.ini".

The following command installs a MySQL Server 5.5 instance from the directory C:\Program Files\MySQL\MySQL Server 5.5 using the service name MySQL55 and setting the root password to 1234.

```shell
shell> MySQLInstanceConfig.exe -i -q "-lC:\mysql_install_log.txt" -"-nMySQL Server 5.5" "-pC:\Program Files\MySQL\MySQL Server 5.5" -v5.5.48 "-tmy-template.ini" "-cC:\mytest.ini" ServerType=DEVELOPMENT DatabaseType=MIXED ConnectionUsage=DSS Port=3311 ServiceName=MySQL55 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 5.5
Version: 5.5.48
Installation Path: C:\Program Files\MySQL\MySQL Server 5.5\
Creating configuration file C:\mytest.ini using template my-template.ini.
```
Installing MySQL on Microsoft Windows Using a noinstall Zip Archive

Options:
DEVELOPMENT
MIXED
DSS
STRICTMODE

Variables:
port: 3311
default-character-set: latin1
basedir: "C:/Program Files/MySQL/MySQL Server 5.5/"
datadir: "C:/Program Files/MySQL/MySQL Server 5.5/Data/"

Creating Windows service entry.
Service name: "MySQL55"
Parameters: "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqld" --defaults-file="C:\mytest.ini" MySQL55.
Windows service MySQL55 installed.

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

Table 5.4 Return Value from MySQL Server Instance Config Wizard

<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 5.5.2, “Automating MySQL Installation on Microsoft Windows Using the MSI Package”.

5.7 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.

5.7.1 Extracting the Install Archive

To install MySQL manually, do the following:
Creating an Option File

1. If you are upgrading from a previous version please refer to Section 5.10, “Upgrading MySQL on 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 C:\mysql. The MySQL Installation Wizard installs MySQL under C:\Program Files\MySQL. If you do not install MySQL at C:\mysql, you must specify the path to the install directory during startup or in an option file. See Section 5.7.2, “Creating an Option File”.

   Note
   The MySQL Installer installs MySQL under C:\Program Files\MySQL.

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.

5.7.2 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 (C:\Program Files\MySQL Server 5.5 and C:\Program Files\MySQL\MySQL Server 5.5\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, C:\, and the MySQL installation directory (for the full list of locations, see Using Option Files). The Windows directory typically is named something like C:\WINDOWS. You can determine its exact location from the value of the WINDIR environment variable using the following command:

C:\> echo %WINDIR%

MySQL looks for options in each location first in the my.ini file, and then in the my.cnf file. However, to avoid confusion, it is best if you use only one file. If your PC uses a boot loader where C:\ is not the boot drive, your only option is to use the my.ini file. Whichever option file you use, it must be a plain text file.

Note
When using the MySQL Installer to install MySQL Server, it will create the my.ini at the default location. And as of MySQL Server 5.5.27, the user running MySQL Installer is granted full permissions to this new my.ini.

In other words, be sure that the MySQL Server user has permission to read the my.ini 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 E:\mysql and the data directory is in E:\mydata\data, you can create an option file containing a [mysqld] section to specify values for the basedir and datadir options:
Selecting a MySQL Server Type

[mysqld]
# set basedir to your installation path
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:

[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.

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 5.5\data) to E:\mydata.

2. Use a --datadir option to specify the new data directory location each time you start the server.

5.7.3 Selecting a MySQL Server Type

The following table shows the available servers for Windows in MySQL 5.5.

<table>
<thead>
<tr>
<th>Binary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqlld</td>
<td>Optimized binary with named-pipe support</td>
</tr>
<tr>
<td>mysqlld-debug</td>
<td>Like mysqlld, 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 5.5 servers have support for symbolic linking of database directories.

MySQL supports TCP/IP on all Windows platforms. MySQL servers on Windows also support named pipes, 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. The default is to use TCP/IP regardless of platform because named pipes are slower than TCP/IP in many Windows configurations.

5.7.4 Starting the Server 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.
Starting the Server for the First Time

Note
The MySQL server will automatically start after using the MySQL Installer, and the MySQL Notifier GUI can be used to start/stop/restart at any time.

The examples in these sections assume that MySQL is installed under the default location of C:\Program Files\MySQL\MySQL Server 5.5. 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 5.7.3, “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 5.5\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: '5.5.48' 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 5.5\data by default). The error log is the file with the .err extension, and may be set using the \--)=log-error option.
5.7.5 Starting MySQL 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.

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 9.4, “Securing the Initial MySQL Accounts”.

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 5.5\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:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqladmin" -u root shutdown
```

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.

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 5.5\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 DEBUG Package.

Use `mysqld --verbose --help` to display all the options that `mysqld` supports.

5.7.6 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:
Starting MySQL as a Windows Service

• 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 5.5\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.

5.7.7 Starting MySQL as a 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. 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:
Starting MySQL as a Windows Service

C:\> "C:\Program Files\MySQL\MySQL Server 5.5\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 5.5\bin\mysqld" --install

The service-installation command does not start the server. Instructions for that are given later in this section.

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 5.5\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.
Starting MySQL as a Windows Service

• 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 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 5.5\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 5.5\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 5.5\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:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqld" --remove
```

If mysqld is not running as a service, you can start it from the command line. For instructions, see Section 5.7.5, "Starting MySQL from the Windows Command Line".

If you encounter difficulties during installation, see Section 5.8, "Troubleshooting a Microsoft Windows MySQL Server Installation".

#### 5.7.8 Testing The MySQL Installation

You can test whether the MySQL server is working by executing any of the following commands:

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqlshow"
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqlshow" -u root mysql
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqladmin" version status proc
C:\> "C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql" test
```

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.

#### 5.8 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 5.5\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 5.7.7, “Starting MySQL as a 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 5.5` and `C:\Program Files\MySQL\MySQL Server 5.5\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 5.5`, 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 5.5
  # 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 5.7.2, “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 Installer, you might 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](http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/delsrv-o.asp) and use the `delsrv mysql` syntax.

### 5.9 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 statements.
- **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 MySQL Installer, you may have already assigned passwords to the accounts. (See Section 5.3, “Installing MySQL on Microsoft Windows Using MySQL Installer”.) Otherwise, use the password-assignment procedure given in Section 9.4, “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 5.7.4, “Starting the Server for the First Time”). You can also set up a MySQL service that runs automatically when Windows starts (see Section 5.7.7, “Starting MySQL as a 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 MySQL Installer (see Section 5.3, “Installing MySQL on Microsoft Windows Using MySQL Installer”), the default installation directory is `C:\Program Files\MySQL\MySQL Server 5.5`:

```
C:\> cd "C:\Program Files\MySQL\MySQL Server 5.5"
```

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 5.7.6, “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              |
| performance_schema |
| 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 MySQL using MySQL Installer, 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 must use those options if you have already secured the initial MySQL accounts.) With `-p`, the client program prompts for the `root` password. For example:

```
C:\> bin\mysqlshow -u root -p
Enter password: (enter root password here)
```

```
+--------------------+
|     Databases     |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
```

If you specify a database name, `mysqlshow` displays a list of the tables within the database:

```
C:\> bin\mysqlshow mysql
```
**Upgrading MySQL on Windows**

Use the `mysql` program to select information from a table in the `mysql` database:

```bash
C:\> bin\mysql -e "SELECT User, Host, plugin FROM mysql.user" mysql
+----------+----------+-----------------------+
| User     | Host     | plugin                |
|----------+----------+-----------------------+
| root     | localhost| mysql_native_password |
+----------+----------+-----------------------+
```

For more information about `mysql` and `mysqlshow`, see [mysql — The MySQL Command-Line Tool](http://dev.mysql.com/doc/mysql/en/mysql.html), and [mysqlshow — Display Database, Table, and Column Information](http://dev.mysql.com/doc/mysql/en/mysqlshow.html).

### 5.10 Upgrading MySQL on Windows

To upgrade MySQL on Windows, follow these steps:


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 5.1 to 5.5, use `mysqladmin` from MySQL 5.1 as follows:
Upgrading MySQL on Windows

5. Before upgrading to MySQL 5.5 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:

```bash
C:\> C:\mysql\bin\mysqld --remove
```

**Note**

If you do not remove the existing service, the MySQL Installation Wizard may fail to properly install the new MySQL service.

6. If you are using the MySQL Installer, start it as described in Section 5.3, “Installing MySQL on Microsoft Windows Using MySQL Installer”.

   If you are using the MySQL Installation Wizard, start the wizard as described in Section 5.5.1, “Using the MySQL Installation Wizard”.

7. 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

8. 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 5.7.7, “Starting MySQL as a Windows Service”.)

9. Restart the server. For example, use `NET START MySQL` if you run MySQL as a service, or invoke `mysqld` directly otherwise.
10. 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`.

11. If you encounter errors, see Section 5.8, “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 ................................................................. 125
6.2 Installing MySQL on OS X Using Native Packages ...................................................... 126
6.3 Installing a MySQL Launch Daemon ............................................................................. 132
6.4 Installing and Using the MySQL Preference Pane ......................................................... 135

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, 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. The user you use to perform the installation must have administrator privileges.

• Compressed TAR archive, 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.

In addition to the core installation, the Package Installer also includes Section 6.3, “Installing a MySQL Launch Daemon” and Section 6.4, “Installing and Using the MySQL Preference Pane”, both of which simplify the management of your installation.

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:

• As of MySQL server 5.5.45, the DMG bundles a launchd daemon instead of the deprecated startup item. Startup items do not function as of OS X 10.10 (Yosemite), so using launchd is preferred. The available MySQL preference pane under OS X System Preferences was also updated to use launchd.

• 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.

• 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. By default, the MySQL directories are installed under /usr/local/.
• 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:

```bash
alias mysql=/usr/local/mysql/bin/mysql
alias mysqladmin=/usr/local/mysql/bin/mysqladmin
```

For `tcsh`, use:

```bash
alias mysql /usr/local/mysql/bin/mysql
alias mysqladmin /usr/local/mysql/bin/mysqladmin
```

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`.

• Prior to OS X 10.7, MySQL server was bundled with OS X Server.

## 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), the preference pane, or `mysqladmin shutdown` on the command line.

When installing from the package version, you can 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 6.4, “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-5.5.48-osx10.9-x86_64.dmg` installs MySQL into `/usr/local/mysql-5.5.48-osx10.9-x86_64/`. The following table shows the layout of the installation directory.

**Table 6.1 MySQL Installation Layout on OS X**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents of Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin, scripts</td>
<td><code>mysqlld</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>
</tbody>
</table>
Directory | Contents of Directory
--- | ---
mysql-test | MySQL test suite
share | Miscellaneous support files, including error messages, sample configuration files, SQL for database installation
sql-bench | Benchmarks
support-files | Scripts and sample configuration files
/tmp/mysql.sock | Location of the MySQL Unix socket

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 file. Double-click the disk image to open it.

![Figure 6.1 MySQL Package Installer: DMG Contents](image)

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 5.5.48, double-click `mysql-5.5.48-osx-10.9-x86_64.pkg`.

3. You will be presented with the opening installer dialog. Click **Continue** to begin installation.
4. If you have downloaded the community version of MySQL, you will be shown a copy of the relevant GNU General Public License. Click Continue and then Agree to continue.

5. From the **Installation Type** page you can either click Install to execute the installation wizard using all defaults, click Customize to alter which components to install (MySQL server, Preference Pane, Launchd Support -- all enabled by default), or click Change Installation Location to change the type of installation for either all users, only the user executing the Installer, or define a custom location.
Figure 6.3 MySQL Package Installer: Installation Type
Figure 6.4 MySQL Package Installer: Destination Select (Change Installation Location)
6. Click Install to begin the installation process.

7. Once the installation has been completed successfully, you will be shown an **Install Succeeded** message with a short summary. Now, Close the wizard and begin using the MySQL server.
MySQL server is now installed, but it is not loaded (started) by default. Use either launchctl from the command line, or start MySQL by clicking “Start” using the MySQL preference pane. For additional information, see Section 6.3, “Installing a MySQL Launch Daemon”, and Section 6.4, “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.

Note

Before MySQL 5.5.45, the OS X builds installed startup items instead of launchd daemons. However, startup items do not function as of OS X 10.10 (Yosemite). The OS X builds now install launchd daemons.

By default, the installation package (DMG) on OS X installs a launchd file named `/Library/LaunchDaemons/com.oracle.oss.mysql.mysqlsd.plist` that contains a plist definition similar to:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
  "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
```
Installing a MySQL Launch Daemon

```xml
<dict>
    <key>Label</key>             <string>com.oracle.oss.mysql.mysqld</string>
    <key>ProcessType</key>       <string>Interactive</string>
    <key>Disabled</key>          <false/>
    <key>RunAtLoad</key>         <true/>
    <key>KeepAlive</key>         <true/>
    <key>SessionCreate</key>     <true/>
    <key>LaunchOnlyOnce</key>    <false/>
    <key>UserName</key>          <string>_mysql</string>
    <key>GroupName</key>         <string>_mysql</string>
    <key>ExitTimeOut</key>       <integer>600</integer>
    <key>Program</key>           <string>/usr/local/mysql/bin/mysqld</string>
    <key>ProgramArguments</key>
        <array>
            <string>/usr/local/mysql/bin/mysqld</string>
            <string>--user=_mysql</string>
            <string>--basedir=/usr/local/mysql</string>
            <string>--datadir=/usr/local/mysql/data</string>
            <string>--plugin-dir=/usr/local/mysql/lib/plugin</string>
            <string>--log-error=/usr/local/mysql/data/mysqld.local.err</string>
            <string>--pid-file=/usr/local/mysql/data/mysqld.local.pid</string>
            <string>--port=3306</string>
        </array>
    <key>WorkingDirectory</key>  <string>/usr/local/mysql</string>
</dict>
</plist>

Note

Some users report that adding a plist DOCTYPE declaration causes the launchd operation to fail, despite it passing the lint check. We suspect it's a copy-n-paste error. The md5 checksum of a file containing the above snippet is 60d7963a0bb2994b69b8b9c123db09df.

To enable the launchd service, you can either:

- Click Start MySQL Server from the MySQL preference pane.
Figure 6.7 MySQL Preference Pane: Location
Figure 6.8 MySQL Preference Pane: Usage

- Or, manually load the launchd file.

```shell
shell> cd /Library/LaunchDaemons
shell> sudo launchctl load -F com.oracle.oss.mysql.mysqld.plist
```

**Note**

When upgrading MySQL server, the launchd installation process will remove the old startup items that were installed with MySQL server 5.5.44 and below.

### 6.4 Installing and Using the MySQL Preference Pane

The MySQL Installation Package includes a MySQL preference pane that enables you to start, stop, and control automated startup during boot of your MySQL installation.

This preference pane is installed by default, and is listed under your system's *System Preferences* window.
Figure 6.9 MySQL Preference Pane: Location

To install the MySQL Preference Pane:

1. Download and open the MySQL package installer, which is provided on a disk image (.dmg) that includes the main MySQL installation package.

   **Note**

   Before MySQL 5.5.45, OS X packages included the deprecated startup items instead of launchd daemons, and the preference pane managed that instead of launchd.
2. Go through the process of installing the MySQL server, as described in the documentation at Section 6.2, “Installing MySQL on OS X Using Native Packages”.

3. Click Customize at the Installation Type step. The “Preference Pane” option is listed there and enabled by default.
4. Complete the MySQL server installation process.

   **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 bottom section of the preference panes list.
Figure 6.12 MySQL Preference Pane: Location
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.
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”.
- Installing using a precompiled Debian package. For more information, see Section 7.2, “Installing MySQL on Linux Using Debian Packages”.
- Installing using Oracle’s Unbreakable Linux Network (ULN). For more information, see Installing MySQL Using Unbreakable Linux Network (ULN).

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 7.3, “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

**Note**

To install or upgrade to MySQL 5.5.31, be sure to read the special instructions at the end of this section.

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”.

---

141
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 7.1 MySQL Installation Layout for Linux RPM Packages**

<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 <code>mysqld</code> server</td>
</tr>
<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.** Standard MySQL server RPMs built by MySQL do not provide support for
the NDBCLUSTER storage engine.

---

**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 MySQL Cluster Installation and
Upgrades.

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.

If the data directory exists at RPM installation time, the installation process does not modify existing data.
This has the effect, for example, that accounts in the grant tables are not initialized to the default set of
accounts.

If you get a dependency failure when trying to install MySQL packages (for example, `error: removing
these packages would break dependencies: libmysqlclient.so.10 is needed
by ...`), you should also install the MySQL-shared-compat package, which includes the shared
libraries for older releases for backward compatibility.

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. If you plan
to install multiple RPM packages, you may wish to download the RPM Bundle `tar` file instead, which
contains multiple RPM packages so that you need not download them separately.

- **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-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. Prior to MySQL 5.5.6, 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. It contains single-threaded and thread-safe libraries. Install this package 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. Before MySQL 5.5.6, MySQL-shared-compat also includes the libraries for the current release, so if you install it, you should not also install MySQL-shared. As of 5.5.6, MySQL-shared-compat does not include the current library version, so there is no conflict.

As of MySQL 5.5.23, the MySQL-shared-compat RPM package enables users of Red Hat-provided mysql-*-5.1 RPM packages to migrate to Oracle-provided MySQL-*-5.5 packages. MySQL-shared-compat replaces the Red Hat mysql-libs package by replacing libmysqlclient.so files of the latter package, thus satisfying dependencies of other packages on mysql-libs. This change affects only users of Red Hat (or Red Hat-compatible) RPM packages. Nothing is different for users of Oracle RPM packages.

• MySQL-embedded-VERSION.glibc23.i386.rpm

The embedded MySQL server library.

• 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:

```
.VERSION.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>
<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>el6</td>
<td>Enterprise Linux 6</td>
</tr>
<tr>
<td>sles10, sles11</td>
<td>SuSE Linux Enterprise Server 10 or 11</td>
</tr>
</tbody>
</table>

In MySQL 5.5, 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):

```shell
cd /path/to/mysql
rpm -qpl MySQL-server-VERSION.glibc23.i386.rpm
```

To perform a standard minimal installation, install the server and client RPMs:

```shell
cd /path/to/mysql
rpm -i MySQL-server-VERSION.glibc23.i386.rpm
rpm -i MySQL-client-VERSION.glibc23.i386.rpm
```

To install only the client programs, install just the client RPM:

```shell
cd /path/to/mysql
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 9.5, “Starting and Stopping MySQL Automatically”, for more information on how MySQL can be started automatically on system startup.

In MySQL 5.5.5 and later, during a new installation using RPM packages, the server boot scripts are installed, but the MySQL server is not started at the end of the installation, since the status of the server during an unattended installation is not known.

In MySQL 5.5.5 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 9, *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`, `.lessht`, `.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.

In MySQL 5.5.31, the RPM spec file has been updated, which has the following consequences:

- For a non-upgrade installation (no existing MySQL version installed), it possible to install MySQL using `yum`.
- For upgrades, it is necessary to clean up any earlier MySQL installations. In effect, the update is performed by removing the old installations and installing the new one.

Additional details follow.

For a non-upgrade installation of MySQL 5.5.31, it is possible to install using `yum`:

```
shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
```

For upgrades to MySQL 5.5.31, the upgrade is performed by removing the old installation and installing the new one. To do this, use the following procedure:

1. Remove the existing 5.5.X installation. `OLDVERSION` is the version to remove.

   ```
   shell> rpm -e MySQL-server-OLDVERSION.glibc23.i386.rpm
   ```

   Repeat this step for all installed MySQL RPMs.

2. Install the new version. `NEWVERSION` is the version to install.

   ```
   shell> rpm -ivh MySQL-server-NEWVERSION.glibc23.i386.rpm
   ```

Alternatively, the removal and installation can be done using `yum`:

```
shell> yum remove MySQL-server-OLDVERSION.glibc23.i386.rpm
shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
```

### 7.2 Installing MySQL on Linux Using Debian Packages

Oracle provides Debian packages for installation on Debian or Debian-like Linux systems. To obtain a package, see Section 2.2, “How to Get MySQL”.

**Note**

Debian distributions of MySQL are also provided by other vendors. Be aware that they may differ from those built by us 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.

Debian package files have names in `mysql-MVER-DVER-CPU.deb` format. `MVER` is the MySQL version and `DVER` is the Debian version. The `CPU` value indicates the processor type or family for which the package is built, as shown in the following table.

**Table 7.4 MySQL Installation Packages for Linux CPU Identifiers**

<table>
<thead>
<tr>
<th>CPU Value</th>
<th>Intended Processor Type or Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>i686</td>
<td>Pentium processor or better, 32 bit</td>
</tr>
<tr>
<td>x86_64</td>
<td>64-bit x86 processor</td>
</tr>
</tbody>
</table>

After downloading a Debian package, use the following command to install it:

```
shell> dpkg -i mysql-MVER-DVER-CPU.deb
```

The Debian package installs files in the `/opt/mysql/server-5.5` directory.

You may also need to install the `libaio` library if it is not already present on your system:

```
shell> sudo apt-get install libaio1
```

### 7.3 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:

  ```
  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
  ```
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`:

```
root-shell> service mysqld start
```

To enable the server to be started and stopped automatically during boot, use `chkconfig`:

```
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.

```
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
  libmysqlclient15off libmysqlclient16 libnet-daemon-perl libplrpc-perl mailx
  mysql-common postfix
Suggested packages:
  dbishell libbinc-sharedcache-perl tinyca procmail postfix mysql postfix-pgsql
  postfix-ldap postfix-pcre sasl2-bin resolvconf postfix-cdb
The following NEW packages will be installed
  bsd-mailx libdbd-mysql-perl libdbi-perl libhtml-template-perl
  libmysqlclient15off 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 libdbd-perl (1.607-1)...
... Setting up libmysqlclient15off (5.1.30really5.0.75-0ubuntu10.5)...
... Setting up libmysqlclient16 (5.1.31-lubuntu2)...
... Setting up mysql-client-5.1 (5.1.31-lubuntu2)...
... Setting up mysql-server-5.1 (5.1.31-lubuntu2)...
  * Stopping MySQL database server mysqld
  ...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 mysqld
  ...done.
  * Checking for corrupt, not cleanly closed and upgrade needing tables.
```

A sample installation of the MySQL packages might look like this (some sections trimmed for clarity):
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 ................................................................. 154
8.2 Installing MySQL on OpenSolaris Using IPS ................................................................. 155

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.

To obtain a binary MySQL distribution for Solaris in tarball or PKG format, http://dev.mysql.com/downloads/mysql/5.5.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:

  ```
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-5.5.48-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-5.5.48-solaris10-x86_64.pkg
```

The following packages are available:
```
  1 mysql MySQL Community Server (GPL) (i86pc) 5.5.48
```

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
shell> pkgrm mysql
shell> pkgadd -d mysql-5.5.48-solaris10-x86_64.pkg
shell> mysql_safe &
shell> mysql_upgrade
```
You should check the notes in Chapter 10, 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:

```bash
shell> pfexec pkg install SUNWmysql55
```

The package set installs three individual packages, SUNWmysql55lib, which contains the MySQL client libraries; SUNWmysql55r which contains the root components, including SMF and configuration files; and SUNWmysql55u 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 5.5, the full installation is located in /usr/mysql/5.5. The default data directory is /var/mysql/5.5/data. The configuration file is installed in /etc/mysql/5.5/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_55. To start and enable MySQL to be started at boot time:

```bash
shell> svcadm enable mysql
```

To disable MySQL from starting during boot time, and shut the MySQL server down if it is running, use:

```bash
shell> svcadm disable mysql
```

To restart MySQL, for example after a configuration file changes, use the restart option:

```bash
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:

```bash
shell> svccfg
svc:> select mysql:version_55
svc:/application/database/mysql:version_55> 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_55> setprop mysql/enable_64bit=1

You need to refresh the SMF after setting these options:

shell> svcadm refresh mysql
Chapter 9 Postinstallation Setup and Testing

Table of Contents

9.1 Initializing the Data Directory ................................................................. 157
  9.1.1 Problems Running mysql_install_db .............................................. 159
9.2 Starting the Server ........................................................................... 161
  9.2.1 Troubleshooting Problems Starting the MySQL Server ............... 162
9.3 Testing the Server ............................................................................ 164
9.4 Securing the Initial MySQL Accounts ............................................. 166
9.5 Starting and Stopping MySQL Automatically .................................. 170

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 9.1, “Initializing the Data Directory”.

• For instructions, see Section 9.2, “Starting the Server”, and Section 9.3, “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 9.4, “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 9.5, “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.

9.1 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, as described in Chapter 9, 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 5.9, “Windows Postinstallation Procedures”.) For some suggested commands that you can use to test whether the server is accessible and working properly, see Section 9.3, “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:

    \[\text{shell> cd BASEDIR}\]

    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.

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.

    \[\text{shell> chown -R mysql .}\]
    \[\text{shell> chgrp -R mysql .}\]

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.

    \[\text{shell> scripts/mysql_install_db --user=mysql}\]

    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.

    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:
Problems Running mysql_install_db

If you do not want to have the test database, you can remove it after starting the server, using the instructions in Section 9.4, “Securing the Initial MySQL Accounts”.

If you have trouble with mysql_install_db at this point, see Section 9.1.1, “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.

```bash
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 9.5, “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.

9.1.1 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.
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.

  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
  ```
Starting the Server

Shell> bin/mysqld_safe --user=mysql &

If mysql_install_db is located in the scripts directory, modify the first command to scripts/mysql_install_db.

See How to Protect or Change the MySQL Unix Socket File, and Chapter 11, 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.

9.2 Starting the Server

This section describes how start the server on Unix and Unix-like systems. (For Windows, see Section 5.7.4, “Starting the Server for the First Time”.) For some suggested commands that you can use to test whether the server is accessible and working properly, see Section 9.3, “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 9.2.1, “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 12, 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.

9.2.1 Troubleshooting Problems Starting the MySQL Server

This section provides troubleshooting suggestions for problems starting the server. For additional suggestions for Windows systems, see Section 5.8, “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.

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 mysqld 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 invoking mysqld 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 `mysqld` 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 `mysqld` 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 `mysqld` with those options followed by the `--verbose` and `--help` options. For example, if you change location into the directory where `mysqld` 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 `mysqld`, 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 5.5\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:
Testing the Server

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.

9.3 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 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:
Testing the Server

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 5.5.48, for pc-linux-gnu on i686
   Server version          5.5.48
   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 `mysqld_safe` or by invoking `mysqld` directly. For example:

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

If `mysqld_safe` fails, see Section 9.2.1, “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
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
```

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              |
```
Securing the Initial MySQL Accounts

<table>
<thead>
<tr>
<th>db</th>
<th>event</th>
<th>func</th>
</tr>
</thead>
<tbody>
<tr>
<td>general_log</td>
<td>help_category</td>
<td>help_keyword</td>
</tr>
<tr>
<td>help_relation</td>
<td>help_topic</td>
<td>host</td>
</tr>
<tr>
<td>ndb_binlog_index</td>
<td>plugin</td>
<td>proc</td>
</tr>
<tr>
<td>procs_priv</td>
<td>proxies_priv</td>
<td>servers</td>
</tr>
<tr>
<td>servers</td>
<td>slow_log</td>
<td>tables_priv</td>
</tr>
<tr>
<td>time_zone</td>
<td>time_zone_leap_second</td>
<td>time_zone_name</td>
</tr>
<tr>
<td>time_zone_transition</td>
<td>time_zone_transition_type</td>
<td>user</td>
</tr>
<tr>
<td>user</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use the `mysql` program to select information from a table in the `mysql` database:

```
shell> bin/mysql -e "SELECT User, Host, plugin FROM mysql.user" mysql
```

```
<table>
<thead>
<tr>
<th>User</th>
<th>Host</th>
<th>plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>localhost</td>
<td>mysql_native_password</td>
</tr>
</tbody>
</table>
```

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 9.4, “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`.

### 9.4 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 9, *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`, the IP address `127.0.0.1`, or the IPv6 address `::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, the IPv6 address ::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. The ::1 account is used for IPv6 connections.

- 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.

- 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.

- The 'root'@'localhost' account also has a row in the mysql.proxies_priv table that enables granting the PROXY privilege for ''@'', that is, for all users and all hosts. This enables root to set up proxy users, as well as to delegate to other accounts the authority to set up proxy users. See Proxy Users.

To display which accounts exist in the mysql.user table and check whether their passwords are empty, use the following statement:

```sql
mysql> SELECT User, Host, Password FROM mysql.user;
```

<table>
<thead>
<tr>
<th>User</th>
<th>Host</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>localhost</td>
<td></td>
</tr>
<tr>
<td>root</td>
<td>myhost.example.com</td>
<td></td>
</tr>
<tr>
<td>root</td>
<td>127.0.0.1</td>
<td></td>
</tr>
<tr>
<td>root</td>
<td>::1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>localhost</td>
<td></td>
</tr>
<tr>
<td></td>
<td>myhost.example.com</td>
<td></td>
</tr>
</tbody>
</table>

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
Assigning root Account Passwords

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'@'::1' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'host_name' = 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'@'::1' = PASSWORD('new_password');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('new_password');
```
Assigning Anonymous Account Passwords

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> 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> 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.

The mysqladmin method of setting the root account passwords does not work for the 'root'@'127.0.0.1' or 'root'@'::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 ''@'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
```
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
Enter password: (enter root password here)
mysql> DROP USER ''@'localhost';
```

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%';
```

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;
```

9.5 Starting and Stopping MySQL Automatically

This section discusses methods for starting and stopping the MySQL server.

Generally, you start the `mysql` 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 5.7.7, “Starting MySQL as a Windows Service.”
Starting and Stopping MySQL Automatically

- 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 6.4, "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 9.1 MySQL Startup Scripts and Supported Server Option Groups**

<table>
<thead>
<tr>
<th>Script</th>
<th>Option Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>[mysql], [server], [mysqld-major_version]</td>
</tr>
<tr>
<td>mysql_safe</td>
<td>[mysql], [server], [mysql_safe]</td>
</tr>
<tr>
<td>mysql.server</td>
<td>[mysql], [mysql.server], [server]</td>
</tr>
</tbody>
</table>

[mysqld-major_version] means that groups with names like `[mysqld-5.1]` and `[mysqld-5.5]` are read by servers having versions 5.1.x, 5.5.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 `mysql_safe` also reads the `[safe_mysqld]` group. However, you should update your option files to use the `[mysql.server]` and `[mysqld_safe]` groups instead.

For more information on MySQL configuration files and their structure and contents, see Using Option Files.
Chapter 10 Upgrading or Downgrading MySQL

Table of Contents

10.1 Upgrading MySQL ............................................................... 173
  10.1.1 Changes Affecting Upgrades to 5.5 .................................. 177
10.2 Downgrading MySQL .......................................................... 183
  10.2.1 Changes Affecting Downgrades from MySQL 5.5 ............... 187
  10.2.2 Downgrading from MySQL Enterprise Edition to MySQL Community Server .................. 187
10.3 Checking Whether Tables or Indexes Must Be Rebuilt .................... 188
10.4 Rebuilding or Repairing Tables or Indexes ................................ 190
10.5 Copying MySQL Databases to Another Machine ............................. 192

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.

10.1 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 5.1 to 5.5.
For in-place and logical upgrade procedures, see Performing an In-place Upgrade, and Performing a Logical Upgrade.

If you run MySQL Server on Windows, see Section 5.10, “Upgrading MySQL on Windows”.

**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 5.5.45 to 5.5.46 is supported. Skipping release series versions is also supported. For example, upgrading from 5.5.44 to 5.5.46 is supported.

- Upgrading one release level is supported. For example, upgrading from 5.1 to 5.5 is supported. Upgrading to the latest release series version is recommended before upgrading to the next release level. For example, upgrade to the latest 5.1 release before upgrading to 5.5.

- 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 5.0 and wish to upgrade to a newer series, upgrade to MySQL 5.1 first before upgrading to MySQL 5.5, and so forth. For information on upgrading to MySQL 5.1 see the MySQL 5.1 Reference Manual.

- Direct upgrades that skip a release level (for example, upgrading directly from MySQL 5.0 to 5.5) 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 example, upgrading from 5.5.7 to 5.5.8 is not supported, as 5.5.7 is not a GA status release.

- 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 5.5 or differ from those found in earlier MySQL releases. Some of these changes may result in incompatibilities.

- Review Section 10.1.1, “Changes Affecting Upgrades to 5.5”. This section describes changes that may require action before or after upgrading.

- Check Section 10.3, “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 10.4, “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 10.1.1, “Changes Affecting Upgrades to 5.5” 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 -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 old MySQL server. For example:

   ```shell>
   bin/mysqladmin -u root -password shutdown
   ```

4. Upgrade the MySQL binaries or packages in place (replace the old binaries with the new ones).

5. Start the MySQL 5.5 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 -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.
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 10.1.1, “Changes Affecting Upgrades to 5.5” for steps to be performed before upgrading.

2. Export your existing data from the previous MySQL version:

   ```
   shell> mysql dump --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 5.5. 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/5.5-datadir
   ```

6. Start the MySQL 5.5 server, using the new data directory. For example:

   ```
   shell> bin/mysqld_safe --user=mysql --datadir=/path/to/5.5-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
   ```

   **Note**

   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.
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/5.5-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. Recompilation 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.

---

**10.1.1 Changes Affecting Upgrades to 5.5**

Before upgrading to MySQL 5.5, 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 10.4, “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 5.5

- **Server Changes**
- **SQL Changes**

### Configuration Changes

**Incompatible change:** The InnoDB Plugin is included in MySQL 5.5 releases. It becomes the built-in version of InnoDB in MySQL Server, replacing the version previously included as the built-in InnoDB engine. InnoDB Plugin is also available in MySQL 5.1 as of 5.1.38, but it is an optional storage engine that must be enabled explicitly using two server options:

```
[mysqld]
ignore-built-in-innodb
plugin-load=innodb=ha_innodb_plugin.so
```

If you were using InnoDB Plugin in MySQL 5.1 by means of those options, you must remove them after an upgrade to 5.5 or the server will fail to start.

In addition, in InnoDB Plugin, the `innodb_file_io_threads` system variable has been removed and replaced with `innodb_read_io_threads` and `innodb_write_io_threads`. If you upgrade from MySQL 5.1 to MySQL 5.5 and previously explicitly set `innodb_file_io_threads` at server startup, you must change your configuration. Either remove any reference to `innodb_file_io_threads` or replace it with references to `innodb_read_io_threads` and `innodb_write_io_threads`.

**Incompatible change:** In MySQL 5.5, the server includes a plugin services interface that complements the plugin API. The services interface enables server functionality to be exposed as a “service” that plugins can access through a function-call interface. The `libmysqlservices` library provides access to the available services and dynamic plugins now must be linked against this library (use the `--lmysqlservices` flag). For an example showing how to configure for CMake, see MySQL Services for Plugins.

### Server Changes

- On Linux systems, the `libaio` library may be needed. Install it first, if it is not already present on your system.

**Known issue:** As of MySQL 5.5.32, 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:

```
ALTER TABLE mysql.help_category MODIFY url TEXT NOT NULL;
ALTER TABLE mysql.help_topic MODIFY url TEXT NOT NULL;
```

**Incompatible change:** As of MySQL 5.5.3, due to work done for Bug #989, `FLUSH TABLES` is not permitted when there is an active `LOCK TABLES ... READ`. To provide a workaround for this restriction, `FLUSH TABLES` has a new variant, `FLUSH TABLES tbl_list WITH READ LOCK`, that enables tables to be flushed and locked in a single operation. As a result of this change, applications that previously used this statement sequence to lock and flush tables will fail:

```
LOCK TABLES tbl_list READ;
FLUSH TABLES tbl_list;
```

Such applications should now use this statement instead:
**Changes Affecting Upgrades to 5.5**

| FLUSH TABLES | tbl_list WITH READ LOCK; |

- **Incompatible change**: As of MySQL 5.5.7, the server requires that a new grant table, `proxies_priv`, be present in the `mysql` database. If you are upgrading to 5.5.7 from a previous MySQL release rather than performing a new installation, the server will find that this table is missing and exit during startup with the following message:

```plaintext
Table 'mysql.proxies_priv' doesn't exist
```

To create the `proxies_priv` table, start the server with the `--skip-grant-tables` option to cause it to skip the normal grant table checks, then run `mysql_upgrade`. For example:

```plaintext
shell> mysqld --skip-grant-tables &
shell> mysql_upgrade
```

Then stop the server and restart it normally.

You can specify other options on the `mysqld` command line if necessary. Alternatively, if your installation is configured so that the server normally reads options from an option file, use the `--defaults-file` option to specify the file (enter each command on a single line):

```plaintext
shell> mysqld --defaults-file=/usr/local/mysql/etc/my.cnf
    --skip-grant-tables &
shell> mysql_upgrade
```

With the `--skip-grant-tables` option, the server does no password or privilege checking, so any client can connect and effectively have all privileges. For additional security, use the `--skip-networking` option as well to prevent remote clients from connecting.

**Note**

This problem is fixed in MySQL 5.5.8; the server treats a missing `proxies_priv` table as equivalent to an empty table. However, after starting the server, you should still run `mysql_upgrade` to create the table.

- **Incompatible change**: As of MySQL 5.5.7, InnoDB always uses the fast truncation technique, equivalent to `DROP TABLE` and `CREATE TABLE`. It no longer performs a row-by-row delete for tables with parent-child foreign key relationships. `TRUNCATE TABLE` returns an error for such tables. Modify your SQL to issue `DELETE FROM table_name` for such tables instead.

- **Incompatible change**: Prior to MySQL 5.5.7, 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 renames 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:

```plaintext
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 5.5.7; 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:
Changes Affecting Upgrades to 5.5

• Incompatible change: As of MySQL 5.5.6, handling of `CREATE TABLE IF NOT EXISTS ... SELECT` statements has been changed for the case that the destination table already exists:

  • Previously, for `CREATE TABLE IF NOT EXISTS ... SELECT`, MySQL produced a warning that the table exists, but inserted the rows and wrote the statement to the binary log anyway. By contrast, `CREATE TABLE ... SELECT` (without `IF NOT EXISTS`) failed with an error, but MySQL inserted no rows and did not write the statement to the binary log.

  • MySQL now handles both statements the same way when the destination table exists, in that neither statement inserts rows or is written to the binary log. The difference between them is that MySQL produces a warning when `IF NOT EXISTS` is present and an error when it is not.

This change in handling of `IF NOT EXISTS` results in an incompatibility for statement-based replication from a MySQL 5.1 master with the original behavior and a MySQL 5.5 slave with the new behavior. Suppose that `CREATE TABLE IF NOT EXISTS ... SELECT` is executed on the master and the destination table exists. The result is that rows are inserted on the master but not on the slave. (Row-based replication does not have this problem.)

To address this issue, statement-based binary logging for `CREATE TABLE IF NOT EXISTS ... SELECT` is changed in MySQL 5.1 as of 5.1.51:

  • If the destination table does not exist, there is no change: The statement is logged as is.

  • If the destination table does exist, the statement is logged as the equivalent pair of `CREATE TABLE IF NOT EXISTS` and `INSERT ... SELECT` statements. (If the `SELECT` in the original statement is preceded by `IGNORE` or `REPLACE`, the `INSERT` becomes `INSERT IGNORE` or `REPLACE`, respectively.)

This change provides forward compatibility for statement-based replication from MySQL 5.1 to 5.5 because when the destination table exists, the rows will be inserted on both the master and slave. To take advantage of this compatibility measure, the 5.1 server must be at least 5.1.51 and the 5.5 server must be at least 5.5.6.

To upgrade an existing 5.1-to-5.5 replication scenario, upgrade the master first to 5.1.51 or higher. Note that this differs from the usual replication upgrade advice of upgrading the slave first.

A workaround for applications that wish to achieve the original effect (rows inserted regardless of whether the destination table exists) is to use `CREATE TABLE IF NOT EXISTS` and `INSERT ... SELECT` statements rather than `CREATE TABLE IF NOT EXISTS ... SELECT` statements.

Along with the change just described, the following related change was made: Previously, if an existing view was named as the destination table for `CREATE TABLE IF NOT EXISTS ... SELECT`, rows were inserted into the underlying base table and the statement was written to the binary log. As of MySQL 5.1.51 and 5.5.6, nothing is inserted or logged.

• Incompatible change: Prior to MySQL 5.5.6, if the server was started with `character_set_server` set to `utf16`, it crashed during full-text stopword initialization. Now the stopword file is loaded and searched using `latin1` if `character_set_server` is `ucs2`, `utf16`, or `utf32`. If any table was created with `FULLTEXT` indexes while the server character set was `ucs2`, `utf16`, or `utf32`, it should be repaired using this statement:
REPAIR TABLE  tbl_name  QUICK;

• **Incompatible change**: As of MySQL 5.5.5, all numeric operators and functions on integer, floating-point and DECIMAL values throw an “out of range” error (ER_DATA_OUT_OF_RANGE) rather than returning an incorrect value or NULL, when the result is out of the supported range for the corresponding data type. See Out-of-Range and Overflow Handling.

• **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.

  • When trying replicate any CREATE TABLE or ALTER TABLE statement containing TIMESTAMP (N) from a master MySQL server that supports the TIMESTAMP (N) syntax to a MySQL 5.5.3 or newer slave, the statement causes replication to fail. Similarly, when you try to restore from a binary log written by a server that supports TIMESTAMP (N) to a MySQL 5.5.3 or newer server, any CREATE TABLE or ALTER TABLE statement containing TIMESTAMP (N) causes the backup to fail. This holds true regardless of the logging format.

    It may be possible to fix such issues using a hex editor, by replacing any width arguments used with TIMESTAMP, and the parentheses containing them, with space characters (hexadecimal 20). Be sure to use a programmer's binary hex editor and not a regular text editor or word processor for this; otherwise, the result is almost certain to be a corrupted binary log file. To guard against accidental corruption of the binary log, you should always work on a copy of the file rather than the original.

    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**: As of MySQL 5.5.3, the Unicode implementation has been extended to provide support for supplementary characters that lie outside the Basic Multilingual Plane (BMP). Noteworthy features:

  • utf16 and utf32 character sets have been added. These correspond to the UTF-16 and UTF-32 encodings of the Unicode character set, and they both support supplementary characters.

  • The utf8mb4 character set has been added. This is similar to utf8, but its encoding allows up to four bytes per character to enable support for supplementary characters.

  • The ucs2 character set is essentially unchanged except for the inclusion of some newer BMP characters.

In most respects, upgrading to MySQL 5.5 should present few problems with regard to Unicode usage, although there are some potential areas of incompatibility. These are the primary areas of concern:

• For the variable-length character data types (VARCHAR and the TEXT types), the maximum length in characters is less for utf8mb4 columns than for utf8 columns.
Changes Affecting Upgrades to 5.5

- For all character data types (CHAR, VARCHAR, and the TEXT types), the maximum number of characters that can be indexed is less for utf8mb4 columns than for utf8 columns.

Consequently, if you want to upgrade tables from utf8 to utf8mb4 to take advantage of supplementary-character support, it may be necessary to change some column or index definitions.

For additional details about the new Unicode character sets and potential incompatibilities, see Unicode Support, and Upgrading from Previous to Current Unicode Support.

- **Incompatible change:** As of MySQL 5.5.3, the server includes dtoa, a library for conversion between strings and numbers by David M. Gay. In MySQL, this library provides the basis for improved conversion between string or DECIMAL values and approximate-value (FLOAT or DOUBLE) numbers.

Because the conversions produced by this library differ in some cases from previous results, the potential exists for incompatibilities in applications that rely on previous results. For example, applications that depend on a specific exact result from previous conversions might need adjustment to accommodate additional precision.

For additional information about the properties of dtoa conversions, see Type Conversion in Expression Evaluation.

- **Incompatible change:** In MySQL 5.5, several changes were made regarding the language and character set of error messages:

  - The --language option for specifying the directory for the error message file is now deprecated. The new lc_messages_dir and lc_messages system variables should be used instead, and the server treats --language as an alias for lc_messages_dir.

  - The language system variable has been removed and replaced with the new lc_messages_dir and lc_messages system variables. lc_messages_dir has only a global value and is read only. lc_messages has global and session values and can be modified at runtime, so the error message language can be changed while the server is running, and individual clients each can have a different error message language by changing their session lc_messages value to a different locale name.

  - Error messages previously were constructed in a mix of character sets. This issue is resolved by constructing error messages internally within the server using UTF-8 and returning them to the client in the character set specified by the character_set_results system variable. The content of error messages therefore may in some cases differ from the messages returned previously.

For more information, see Setting the Error Message Language, and Character Set for Error Messages.

- **Incompatible change:** MySQL 5.5 implements new functions used to calculate row placement for tables partitioned by KEY and LINEAR KEY. Tables that were created using KEY or LINEAR KEY partitioning in MySQL 5.1 can be upgraded in MySQL 5.5.31 and later using ALTER TABLE ... PARTITION BY ALGORITHM=2 [LINEAR] KEY (...).(Bug #14521864, Bug #66462)

**SQL Changes**

- **Incompatible change:** Previously, the parser accepted an INTO clause in nested SELECT statements, which is invalid because such statements must return their results to the outer context. As of MySQL 5.5.3, this syntax is no longer permitted and statements that use it must be changed.

- **Incompatible change:** In MySQL 5.5.3, several changes were made to alias resolution in multiple-table DELETE statements so that it is no longer possible to have inconsistent or ambiguous table aliases.
Downgrading MySQL

• In MySQL 5.1.23, alias declarations outside the `table_references` part of the statement were disallowed for the `USING` variant of multiple-table `DELETE` syntax, to reduce the possibility of ambiguous aliases that could lead to ambiguous statements that have unexpected results such as deleting rows from the wrong table.

As of MySQL 5.5.3, alias declarations outside `table_references` are disallowed for all multiple-table `DELETE` statements. Alias declarations are permitted only in the `table_references` part.

Incorrect:

```
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;
```

Correct:

```
DELETE FROM t1 USING t1 AS a1 INNER JOIN t2 AS a2;
DELETE t1 FROM t1 AS a1 INNER JOIN t2 AS a2;
```

• Previously, for alias references in the list of tables from which to delete rows in a multiple-table delete, the default database is used unless one is specified explicitly. For example, if the default database is `db1`, the following statement does not work because the unqualified alias reference `a2` is interpreted as having a database of `db1`:

```
DELETE a1, a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2
WHERE a1.id=a2.id;
```

To correctly match an alias that refers to a table outside the default database, you must explicitly qualify the reference with the name of the proper database:

```
DELETE a1, db2.a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2
WHERE a1.id=db2.a2.id;
```

As of MySQL 5.5.3, alias resolution does not require qualification and alias references should not be qualified with the database name. Qualified names are interpreted as referring to tables, not aliases.

Statements containing alias constructs that are no longer permitted must be rewritten.

• Some keywords may be reserved in MySQL 5.5 that were not reserved in MySQL 5.1. See Keywords and Reserved Words.

### 10.2 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 5.5.46 to 5.5.45.

- **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 5.5.46 to 5.5.45 and for downgrades from 5.5 to 5.1.

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 downgrade methods. For example, downgrading from 5.5.46 to 5.5.45 is supported. Skipping release series versions is also supported. For example, downgrading from 5.5.46 to 5.5.44 is supported.

- Downgrading one release level is supported using the *logical downgrade* method. For example, downgrading from 5.5 to 5.1 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 5.5 to 5.1, and then to 5.0.

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. For example, downgrading from MySQL 5.5.8 to MySQL 5.5.7 is not supported, as 5.5.7 is not a GA status release.

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 10.2.1, “Changes Affecting Downgrades from MySQL 5.5”. 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 10.2.1, “Changes Affecting Downgrades from MySQL 5.5” that require action before or after downgrading are...
Performing an In-place Downgrade

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 10.3, “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 MySQL versions, downgrade the affected tables using the instructions in Section 10.4, “Rebuilding or Repairing Tables or Indexes”.

- If you use XA transactions with InnoDB, run XA RECOVER before downgrading to check for uncommitted XA transactions. If results are returned, either commit or rollback the XA transactions by issuing an XA COMMIT or XA ROLLBACK statement.

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 10.2.1, “Changes Affecting Downgrades from MySQL 5.5” 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. After the slow shutdown, remove the InnoDB redo log files (the ib_logfile* files) from the data directory to avoid downgrade issues related to redo log file format changes that may have occurred between releases.

   ```
   shell> rm ib_logfile*
   ```

5. Downgrade the MySQL binaries or packages in-place by replacing the newer binaries or packages with the older ones.

6. Start the older (downgraded) MySQL server, using the existing data directory. For example:

   ```
   shell> bin/mysqld_safe --user=mysql --datadir=/path/to/existing-datadir
   ```

7. Run mysql_upgrade. For example:
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 10.2.1, “Changes Affecting Downgrades from MySQL 5.5” for steps to be performed before downgrading.

2. Dump all databases. For example:

   ```
   bin/mysqldump --add-drop-table --events -u root -p password --all-databases --force > all_5_5_databases_dump.sql
   ```

3. Shut down the newer MySQL server. For example:

   ```
   bin/mysqladmin -u root -p password shutdown
   ```

4. Initialize an older MySQL instance, with a new data directory. For example:

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

5. Start the older MySQL server, using the new data directory. For example:

   ```
   bin/mysqld_safe --user=mysql --datadir=/path/to/new-datadir
   ```

6. Load the dump file into the older MySQL server. For example:

   ```
   bin/mysql -u root -p password --execute="source all_5_5_databases_dump.sql" --force
   ```

7. Run `mysql_upgrade`. For example:

   ```
   bin/mysql_upgrade -u root -p password
   ```

8. If you use InnoDB, configure MySQL to perform a slow shutdown by setting `innodb_fast_shutdown` to 0. For example:

   ```
   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:

   ```
   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 10.5, “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.

**10.2.1 Changes Affecting Downgrades from MySQL 5.5**

Before downgrading from MySQL 5.5, review the changes described in this section. Some changes may require action before or after downgrading.

- **System Tables.** The mysql.proc.comment column definition changed between MySQL 5.1 and 5.5. After downgrading from 5.5 to 5.1, this table is seen as corrupt and in need of repair. Running mysql_upgrade from the version of MySQL to which you downgraded (as documented in the downgrade procedures) reverts the mysql.proc.comment column definition.

- **InnoDB.** MySQL 5.5 uses InnoDB Plugin as the built-in version of InnoDB. MySQL 5.1 includes InnoDB Plugin as of 5.1.38, but as an option that must be enabled explicitly. See the Release Notes for MySQL 5.1.38.

- **Tables partitioned by [LINEAR] KEY.** MySQL 5.5 implements new functions used to calculate row placement for tables partitioned by KEY and LINEAR KEY. Tables that were created using KEY or LINEAR KEY partitioning in MySQL 5.5 cannot be used by a MySQL 5.1 server. In MySQL 5.5.31 and later, you can downgrade such tables with ALTER TABLE ... PARTITION BY ALGORITHM=1 [LINEAR] KEY (...) to make them compatible with MySQL 5.1.

**10.2.2 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 5.5 and 5.1). Information about upgrading and downgrading between MySQL release series can be found elsewhere in this chapter; see Section 10.1, “Upgrading MySQL”, and Section 10.2, “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. This includes the following commercial extensions:

- **MySQL External Authentication for Windows:** Following uninstallation of this plugin, existing MySQL user accounts must be re-created using local authentication. See MySQL User Account Management, for more information.
• *MySQL External Authentication for PAM:* Following uninstallation of this plugin, existing MySQL user accounts must be re-created using local authentication. See [MySQL User Account Management](#), for more information.

• *MySQL Enterprise Thread Pool:* Following uninstallation of this plugin, existing MySQL servers revert to default thread and connection handling.

• *MySQL Enterprise Audit:* Following uninstallation of this plugin, no logging of user logins or query activity occurs.

• *MySQL High Availability:* Following uninstallation of this plugin, 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.

### 10.3 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 10.4, “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, accessing those tables causes the server to crash, even if you have run `mysql_upgrade` or `CHECK TABLE ... FOR UPGRADE`. To work around this problem, use `mysqldump` to dump all *ARCHIVE* tables.
before upgrading, and reload them into MySQL 5.1 after upgrading. The same problem occurs for binary downgrades from MySQL 5.1 to 5.0.

The upgrade 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

In MySQL 5.6.3, the length limit for index prefix keys is increased from 767 bytes to 3072 bytes, for `InnoDB` tables using `ROW_FORMAT=DYNAMIC` or `ROW_FORMAT=COMPRESSED`. See Limits on InnoDB Tables for details. This change is also backported to MySQL 5.5.14. If you downgrade from one of these releases or higher, to an earlier release with a lower length limit, the index prefix keys could be truncated at 767 bytes or the downgrade could fail. This issue could only occur if the configuration option `innodb_large_prefix` was enabled on the server being downgraded.

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 10.4, “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`. 189
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)**

  Affects indexes for columns that use any of these character sets: `eucjpms, euc_kr, gb2312, latin7, macce, ujls`

  Affected tables can be detected by `CHECK TABLE ... FOR UPGRADE` as of MySQL 5.1.29 (see Bug #39585).

### 10.4 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 10.3, “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`. `REPAIR TABLE` only applies to `MyISAM, ARCHIVE`, and `CSV` tables.

**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> 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> mysqldump db_name > dump.sql
shell> mysql db_name < dump.sql
```

To rebuild all tables in all databases, use the `--all-databases` option:

```
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 an InnoDB table, use this statement:

```
mysql> ALTER TABLE t1 ENGINE = InnoDB;
```

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 need to rebuild an InnoDB table because a `CHECK TABLE` operation indicates that a table upgrade is required, use `mysqldump` to create a dump file and `mysql` to reload the file, as described earlier. If the `CHECK TABLE` operation indicates that there is a corruption or causes InnoDB to fail, refer to Forcing InnoDB Recovery for information about using the `innodb_force_recovery` option to restart InnoDB. To understand the type of problem that `CHECK TABLE` may be encountering, refer to the InnoDB notes in `CHECK TABLE Syntax`.

For MyISAM, ARCHIVE, or CSV tables, you can use `REPAIR TABLE` if the table checking operation indicates that there is a corruption or that an upgrade is required. For example, to repair a MyISAM table, use this statement:

```
mysql> REPAIR TABLE t1;
```

`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.

10.5 Copying MySQL Databases to Another Machine

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.

Note
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.

Use mysqldump --help to see what options are available.

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:

```bash
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:

```bash
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 11 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.

<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 CMake).</td>
</tr>
<tr>
<td>CC</td>
<td>The name of your C compiler (for running CMake).</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 mysql history file is $HOME/.mysql_history.</td>
</tr>
<tr>
<td>LD_RUN_PATH</td>
<td>Used to specify the location of libmysqlclient.so.</td>
</tr>
<tr>
<td>LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN</td>
<td>Enable <code>mysql_clear_password</code> authentication plugin; see The Cleartext Client-Side Authentication Plugin.</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 mysql history file. If this variable is set, its value overrides the default for $HOME/.mysql_history.</td>
</tr>
<tr>
<td>MYSQL_HOME</td>
<td>The path to the directory in which the server-specific my.cnf file resides.</td>
</tr>
<tr>
<td>MYSQL_HOST</td>
<td>The default host name used by the mysql command-line client.</td>
</tr>
<tr>
<td>MYSQL_PS1</td>
<td>The command prompt to use in the mysql command-line client.</td>
</tr>
<tr>
<td>MYSQL_PWD</td>
<td>The default password when connecting to mysqld. Using this is insecure. See End-User Guidelines for Password Security.</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 following table.</td>
</tr>
<tr>
<td>USER</td>
<td>The default user name on Windows when connecting to mysqld.</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 \((\text{UMASK} \mid \text{0600})\) 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 \((\text{UMASK_DIR} \mid \text{0700})\) as the base mode for directory creation, which then is AND-ed with \(\sim(\sim\text{UMASK} \& \text{0666})\), 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 12 Perl Installation Notes

Table of Contents

12.1 Installing Perl on Unix ................................................................. 197
12.2 Installing ActiveState Perl on Windows .......................................... 198
12.3 Problems Using the Perl DBI/DBD Interface ...................................... 199

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 5.5.

12.1 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.
Installing ActiveState Perl on Windows

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`

   `shell> make`

   `shell> make test`

   `shell> 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-

12.2 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/ 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:

   `C:\> set HTTP_proxy=my.proxy.com:3128`

4. Start the PPM program:

   `C:\> C:\perl\bin\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`
Problems Using the Perl DBI/DBD Interface

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:

```perl
use DBI;
$dbh= DBI->connect("DBI:ODBC:$dsn",$user,$password) ||
    die "Got error $DBI::errstr when connecting to $dsn\n";
```

12.3 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`. 
Appendix A Licenses for Third-Party Components

Table of Contents

A.1 ANTLR 3 License .......................................................... 203
A.2 dtoa.c License ............................................................. 204
A.3 Editline Library (libedit) License ........................................ 204
A.4 FindGTest.cmake License .................................................. 206
A.5 Fred Fish's Dbug Library License ......................................... 207
A.6 getarg License ............................................................ 208
A.7 GNU General Public License Version 2.0, June 1991 .................... 209
A.8 GNU Lesser General Public License Version 2.1, February 1999 ...... 214
A.9 GNU Libtool License ....................................................... 222
A.10 GNU readline License ..................................................... 223
A.11 Google Controlling Master Thread I/O Rate Patch License ............ 223
A.12 Google Perftools (TCMalloc utility) License .......................... 224
A.13 Google SMP Patch License ............................................... 224
A.14 lib_sql.cc License ....................................................... 225
A.15 Libaio License ........................................................... 225
A.16 libevent License .......................................................... 225
A.17 Linux-PAM License ....................................................... 228
A.18 md5 (Message-Digest Algorithm 5) License ............................. 228
A.19 memcached License ....................................................... 229
A.20 nt_servc (Windows NT Service class library) License ................. 229
A.21 OpenPAM License ........................................................ 229
A.22 Paramiko License ........................................................ 230
A.23 Percona Multiple I/O Threads Patch License ............................ 230
A.24 Python License ............................................................ 231
A.25 Red HAT RPM Spec File License ......................................... 241
A.26 RegEX-Spencer Library License .......................................... 242
A.27 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License .............. 243
A.28 Richard A. O'Keefe String Library License ............................ 243
A.29 SHA-1 in C License ........................................................ 243
A.30 zlib License .............................................................. 243

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.

• 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 10881 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 5.5

• 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, “Libaio License”
• Section A.16, “libevent License”
• Section A.17, “Linux-PAM License”
• Section A.18, “md5 (Message-Digest Algorithm 5) License”
• Section A.20, “nt_servc (Windows NT Service class library) License”
• Section A.21, “OpenPAM License”
• Section A.23, “Percona Multiple I/O Threads Patch License”
• Section A.25, “Red HAT RPM Spec File License”
• Section A.26, “RegEX-Spencer Library License”
• Section A.27, “RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License”
• Section A.28, “Richard A. O’Keefe String Library License”
• Section A.29, “SHA-1 in C License”
• Section A.30, “zlib License”

MySQL Cluster 7.2

• 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”
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
A.2 dtoa.c License

The following software may be included in this product:

`dtoa.c`

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 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.

THE SOFTWARE IS PROVIDED "AS IS" 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");
Fred Fish's Dbug Library License

The following software may be included in this product:

Fred Fish's Dbug Library

N O T I C E

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 product and explicitly disclaims any implied warranties of merchantability or fitness for any particular purpose.

The dbug_analyze.c file is subject to the following notice:

Copyright June 1987, Binayak Banerjee
All rights reserved.

This program may be freely distributed under the same terms and conditions as Fred Fish's Dbug package.

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
INTERUPTION) 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 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}
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 interface-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
GNU Readline License

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

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 Libaio License

The following software may be included in this product:

libaio

This component is licensed under Section A.8, “GNU Lesser General Public License Version 2.1, February 1999”.

A.16 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.
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.

==

==
libevent License

---

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.17 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 (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.18 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.19 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 (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.20 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.21 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.22 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.23 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.24 Python License

The following software may be included in this product:

Python Programming Language

This is the official license for the Python 2.7 release:

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands
as a successor of a language called ABC. Guido remains Python's
principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for
National Research Initiatives (CNRI, see http://www.cnri.reston.va.us)
in Reston, Virginia where he released several versions of the
software.

In May 2000, Guido and the Python core development team moved to
BeOpen.com to form the BeOpen PythonLabs team. In October of the same
year, the PythonLabs team moved to Digital Creations (now Zope
Corporation, see http://www.zope.com). In 2001, the Python Software
Foundation (PSF, see http://www.python.org/psf/) was formed, a
non-profit organization created specifically to own Python-related
Intellectual Property. Zope Corporation is a sponsoring member of
the PSF.

All Python releases are Open Source (see http://www.opensource.org for
the Open Source Definition). Historically, most, but not all, Python
releases have also been GPL-compatible; the table below summarizes
the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td>1.2</td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.5.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.6</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>
Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under
the GPL. All Python licenses, unlike the GPL, let you distribute
a modified version without making your changes open source. The
GPL-compatible licenses make it possible to combine Python with
other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible,
because its license has a choice of law clause. According to
CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1
is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's
direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
-----------------------------------------------

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF
hereby grants Licensee a nonexclusive, royalty-free, world-wide
license to reproduce, analyze, test, perform and/or display publicly,
prepare derivative works, distribute, and otherwise use Python
alone or in any derivative version, provided, however, that PSF's
License Agreement and PSF's notice of copyright, i.e., "Copyright (c)
Reserved" are retained in Python alone or in any derivative version
prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0
-------------------------------------------

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture
between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
---------------------------------------

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the
law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2
--------------------------------------------------

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, 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.

Licenses and Acknowledgements for Incorporated Software
--------------------------------------------------------

This section is an incomplete, but growing list of licenses and acknowledgements for third-party software incorporated in the Python distribution.

Mersenne Twister
---------------

The _random module includes code based on a download from http://www.math.keio.ac.jp/ matumoto/MT2002/emt19937ar.html. The following are the verbatim comments from the original code:

A C-program for MT19937, with initialization improved 2002/1/26. Coded by Takuji Nishimura and Makoto Matsumoto.

Before using, initialize the state by using init_genrand(seed) or init_by_array(init_key, key_length).

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, 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 names of its contributors may not 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.

Any feedback is very welcome.
http://www.math.keio.ac.jp/matumoto/emt.html
email: matumoto@math.keio.ac.jp

Sockets
-------
The socket module uses the functions, getaddrinfo(), and getnameinfo(), which are coded in separate source files from the WIDE Project, http://www.wide.ad.jp/.

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 project 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.

Floating point exception control
---------------------------------
The source for the fpectl module includes the following notice:

\/*
  Copyright (c) 1996.
  The Regents of the University of California.
  All rights reserved.
*/

Permission to use, copy, modify, and distribute this software for
<ghost@aladdin.com>. Other authors are noted in the change history that follows (in reverse chronological order):

2002-04-13 lpd Removed support for non-ANSI compilers; removed references to Ghostscript; clarified derivation from RFC 1321; now handles byte order either statically or dynamically.
1999-11-04 lpd Edited comments slightly for automatic TOC extraction.
1999-10-18 lpd Fixed typo in header comment (ansi2knr rather than md5); added conditionalization for C++ compilation from Martin Purschke <purschke@bnl.gov>.
1999-05-03 lpd Original version.

Asynchronous socket services
============================
The asynchat and asyncore modules contain the following notice:

Copyright 1996 by Sam Rushing

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Sam Rushing not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SAM RUSHING DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SAM RUSHING BE LIABLE FOR ANY SPECIAL, 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.

Cookie management
=================
The Cookie module contains the following notice:

Copyright 2000 by Timothy O'Malley <timo@alum.mit.edu>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Timothy O'Malley not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Timothy O'Malley DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Timothy O'Malley BE LIABLE FOR ANY SPECIAL, 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.

Profiling
--------
The profile and pstats modules contain the following notice:
Copyright 1994, by InfoSeek Corporation, all rights reserved.
Written by James Roskind

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose (subject to the restriction in the following sentence) without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of InfoSeek not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. This permission is explicitly restricted to the copying and modification of the software to remain in Python, compiled Python, or other languages (such as C) wherein the modified or derived code is exclusively imported into a Python module.

INFOSEEK CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INFOSEEK CORPORATION BE LIABLE FOR ANY SPECIAL, 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.

Execution tracing
--------------------
The trace module contains the following notice:

portions copyright 2001, Autonomous Zones Industries, Inc., all rights...
err... reserved and offered to the public under the terms of the
Python 2.2 license.
Author: Zooko O'Whielacronx
http://zooko.com/
mailto:zooko@zooko.com

Copyright 2000, Mojam Media, Inc., all rights reserved.
Author: Skip Montanaro

Copyright 1999, Bioreason, Inc., all rights reserved.
Author: Andrew Dalke

Copyright 1995-1997, Automatrix, Inc., all rights reserved.
Author: Skip Montanaro

Copyright 1991-1995, Stichting Mathematisch Centrum, all rights reserved.

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of neither Automatrix, Bioreason or Mojam Media be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

UUencode and UUdecode functions
----------------------------------
The uu module contains the following notice:

Copyright 1994 by Lance Ellinghouse
Cathedral City, California Republic, United States of America.
All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Lance Ellinghouse
not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LANCE ELLINGHOUSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL LANCE ELLINGHOUSE CENTRUM BE LIABLE FOR ANY SPECIAL, 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.

Modified by Jack Jansen, CWI, July 1995:
- Use binascii module to do the actual line-by-line conversion between ascii and binary. This results in a 1000-fold speedup. The C version is still 5 times faster, though.
- Arguments more compliant with Python standard

XML Remote Procedure Calls

The xmlrpcclib module contains the following notice:

The XML-RPC client interface is

Copyright (c) 1999-2002 by Secret Labs AB
Copyright (c) 1999-2002 by Fredrik Lundh

By obtaining, using, and/or copying this software and/or its associated documentation, you agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its associated documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Secret Labs AB or the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SECRET LABS AB AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL SECRET LABS AB OR THE AUTHOR BE LIABLE FOR ANY SPECIAL, 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.

test_epoll
----------

The test_epoll contains the following notice:

Copyright (c) 2001-2006 Twisted Matrix Laboratories.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
A.25 Red HAT RPM Spec File License

The following software may be included in this product:

Red Hat RPM Spec File
You are receiving a copy of the Red Hat spec file. The terms of the Oracle license do NOT apply to the Red Hat spec file; it is licensed under the following license, separately from the Oracle programs you receive.

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.
[for rest of text, see following link]

This component is licensed under Section A.7, “GNU General Public License Version 2.0, June 1991”

A.26 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.27 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
A.28 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_cctype.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.29 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.30 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