

A Quick Guide to Using the MySQL APT Repository

Abstract

This is a quick guide to using the MySQL APT repository, which provides [deb](#) packages for installing and managing the MySQL server, client, and other components on the following Linux platforms:

- Debian 8 and 9
- Ubuntu 16.04, 18.04, and 18.10

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.

Document generated on: 2019-05-23 (revision: 62136)

Table of Contents

..... 1

-
- [Steps for a Fresh Installation of MySQL](#)
 - [Selecting a Major Release Version](#)
 - [Installing Additional MySQL Products and Components with APT](#)
 - [Installing MySQL from Source with the MySQL APT Repository](#)
 - [Upgrading MySQL with the MySQL APT Repository](#)
 - [Replacing a Native Distribution of MySQL Using the MySQL APT Repository](#)
 - [Replacing a MySQL Server Installed by a Direct deb Package Download](#)
 - [Removing MySQL with APT](#)
 - [Special Notes on Upgrading the Shared Client Libraries](#)
 - [Installing MySQL NDB Cluster Using the APT Repository](#)
 - [Installing Additional MySQL NDB Cluster Products and Components](#)
 - [Available Packages from the MySQL APT Repository](#)
 - [Appendix A: Adding and Configuring the MySQL APT Repository Manually](#)
 - [Appendix B: Installing MySQL Non-interactively with MySQL APT Repository](#)
 - [Appendix C: Errors with Missing Entries in the Repository Sources List](#)

Steps for a Fresh Installation of MySQL

Note

The following instructions assume that no versions of MySQL (whether distributed by Oracle or other parties) have already been installed on your system; if that is not the case, follow the instructions given in [Replacing a Native Distribution of MySQL Using the MySQL APT Repository](#) or [Replacing a MySQL Server Installed by a Direct deb Package Download](#) instead.

Adding¹the MySQL APT Repository

First, add the MySQL APT repository to your system's software repository list. Follow these steps:

- Go to the download page for the MySQL APT repository at <https://dev.mysql.com/downloads/repo/apt/>.
- Select and download the release package for your Linux distribution.
- Install the downloaded release package with the following command, replacing *version-specific-package-name* with the name of the downloaded package (preceded by its path, if you are not running the command inside the folder where the package is):

```
shell> sudo dpkg -i /PATH/version-specific-package-name.deb
```

For example, for version *w.x.y-z* of the package, the command is:

```
shell> sudo dpkg -i mysql-apt-config_w.x.y-z_all.deb
```

Note that the same package works on all supported Debian and Ubuntu platforms.

- d. During the installation of the package, you will be asked to choose the versions of the MySQL server and other components (for example, the MySQL Workbench) that you want to install. If you are not sure which version to choose, do not change the default options selected for you. You can also choose **none** if you do not want a particular component to be installed. After making the choices for all components, choose **Ok** to finish the configuration and installation of the release package.

You can always change your choices for the versions later; see [Selecting a Major Release Version](#) for instructions.

- e. Update package information from the MySQL APT repository with the following command (*this step is mandatory*):

```
shell> sudo apt-get update
```

Instead of using the release package, you can also add and configure the MySQL APT repository manually; see [Appendix A: Adding and Configuring the MySQL APT Repository Manually](#) for details.

Note

Once the MySQL APT repository is enabled on your system, you will no longer be able to install any MySQL packages from your platform's native software repositories until the MySQL APT repository is disabled.

Installing MySQL with APT

Install MySQL by the following command:

```
shell> sudo apt-get install mysql-server
```

This installs the package for the MySQL server, as well as the packages for the client and for the database common files.

During the installation, you are asked to supply a password for the root user for your MySQL installation.

Important

Make sure you remember the root password you set. Users who want to set a password later can leave the **password** field blank in the dialogue box and just press **Ok**; in that case, root access to the server will be authenticated by [Socket Peer-Credential Pluggable Authentication](#) for connections using a Unix socket file. You can set the root password later using the program [mysql_secure_installation](#).

Starting and Stopping the MySQL Server

The MySQL server is started automatically after installation. You can check the status of the MySQL server with the following command:

```
shell> sudo service mysql status
```

Stop the MySQL server with the following command:

```
shell> sudo service mysql stop
```

To restart the MySQL server, use the following command:

```
shell> sudo service mysql start
```

Note

A few third-party native repository packages that have dependencies on the native MySQL packages may not work with the MySQL APT repository packages and should not be used together with them; these include `akonadi-backend-mysql`, `handlersocket-mysql-5.5`, and `zoneminder`.

Selecting a Major Release Version

By default, all installations and upgrades for your MySQL server and the other required components come from the release series of the major version you have selected during the installation of the configuration package (see [Adding the MySQL APT Repository](#)). However, you can switch to another supported major release series at any time by reconfiguring the configuration package you have installed. Use the following command:

```
shell> sudo dpkg-reconfigure mysql-apt-config
```

A dialogue box then asks you to choose the major release version you want. Make your selection and choose **Ok**. After returning to the command prompt, update package information from the MySQL APT repository with this command:

```
shell> sudo apt-get update
```

The latest version in the selected series will then be installed when you use the `apt-get install` command next time.

You can use the same method to change the version for any other MySQL component you want to install with the MySQL APT repository.

Installing Additional MySQL Products and Components

You can use APT to install individual components of MySQL from the MySQL APT repository. Assuming you already have the MySQL APT repository on your system's repository list (see [Adding the MySQL APT Repository](#) for instructions), first, use the following command to get the latest package information from the MySQL APT repository:

```
shell> sudo apt-get update
```

Install any packages of your choice with the following command, replacing `package-name` with name of the package ([here](#) is a list of available packages):

```
shell> sudo apt-get install package-name
```

For example, to install the MySQL Workbench:

```
shell> sudo apt-get install mysql-workbench-community
```

To install the shared client libraries:

```
shell> sudo apt-get install libmysqlclient18
```

Installing MySQL from Source with the MySQL APT Repository

Note

This feature is only supported on 64-bit systems.

You can download the source code for MySQL and build it using the MySQL APT Repository:

1. Add the MySQL APT repository to your system's repository list and choose the major release series you want (see [Adding the MySQL APT Repository](#) for instructions).
2. Update package information from the MySQL APT repository with the following command (*this step is mandatory*):

```
shell> sudo apt-get update
```

3. Install packages that the build process depends on:

```
shell> sudo apt-get build-dep mysql-server
```

4. Download the source code for the major components of MySQL and then build them (run this command in the folder in which you want the downloaded files and the builds to be located):

```
shell> apt-get source -b mysql-server
```

`deb` packages for installing the various MySQL components are created.

5. Pick the `deb` packages for the MySQL components you need and install them with the command:

```
shell> sudo dpkg -i package-name.deb
```

Notice that dependency relationships exist among the MySQL packages. For a basic installation of the MySQL server, install the database common files package, the client package, the client metapackage, the server package, and the server metapackage (in that order) with the following steps:

- Preconfigure the MySQL server package with the following command:

```
shell> sudo dpkg-preconfigure mysql-community-server_version-and-platform-specific-part.deb
```

You will be asked to provide a password for the root user for your MySQL installation; see important information on root password given in [Installing MySQL with APT](#) above. You might also be asked other questions regarding the installation.

- Install the required packages with a single command:

```
shell> sudo dpkg -i mysql-{common,community-client,client,community-server,server}*.deb
```

- If you are being warned of unmet dependencies by `dpkg`, you can fix them using `apt-get`:

```
sudo apt-get -f install
```

Here are where the files are installed on the system:

- All configuration files (like `my.cnf`) are under `/etc/mysql`
- All binaries, libraries, headers, etc., are under `/usr/bin` and `/usr/sbin`
- The data directory is under `/var/lib/mysql`

See also information given in [Starting and Stopping the MySQL Server](#).

Upgrading MySQL with the MySQL APT Repository

Notes

- Before performing any upgrade to MySQL, follow carefully the instructions in [Upgrading MySQL](#). Among other instructions discussed there, *it is especially important to back up your database before the upgrade*.

- The following instructions assume that MySQL has been installed on your system using the MySQL APT repository; if that is not the case, follow the instructions given in [Replacing a Native Distribution of MySQL Using the MySQL APT Repository](#) or [Replacing a MySQL Server Installed by a Direct deb Package Download](#) instead. Also notice that you cannot use the MySQL APT repository to upgrade a distribution of MySQL that you have installed from a nonnative software repository (for example, from MariaDB or Percona).

Use the MySQL APT repository to perform an in-place upgrade for your MySQL installation (that is, replacing the old version and then running the new version using the old data files) by following these steps:

1. Make sure you already have the MySQL APT repository on your system's repository list (see [Adding the MySQL APT Repository](#) for instructions).
2. Make sure you have the most up-to-date package information on the MySQL APT repository by running:

```
shell> sudo apt-get update
```

3. Note that, by default, the MySQL APT repository will update MySQL to the release series you have selected when you were [adding the MySQL APT repository to your system](#). If you want to upgrade to another release series, select it by following the steps given in [Selecting a Major Release Version](#).

As a general rule, to upgrade from one release series to another, go to the next series rather than skipping a series. For example, if you are currently running MySQL 5.5 and wish to upgrade to a newer series, upgrade to MySQL 5.6 first before upgrading to 5.7, and so on.

Important

- For important information about upgrading from MySQL 5.5 to 5.6, see [Upgrading from MySQL 5.5 to 5.6](#).
- For important information about upgrading from MySQL 5.6 to 5.7, see [Upgrading from MySQL 5.6 to 5.7](#).
- For important information about upgrading from MySQL 5.7 to 8.0, see [Upgrading from MySQL 5.7 to 8.0](#).
- In-place downgrading of MySQL is not supported by the MySQL APT repository. Follow the instructions in [Downgrading MySQL](#).

4. Upgrade MySQL by the following command:

```
shell> sudo apt-get install mysql-server
```

The MySQL server, client, and the database common files are upgraded if newer versions are available. To upgrade any other MySQL package, use the same `apt-get install` command and supply the name for the package you want to upgrade:

```
shell> sudo apt-get install package-name
```

To see the names of the packages you have installed from the MySQL APT repository, use the following command:

```
shell> dpkg -l | grep mysql | grep ii
```

Note

If you perform a system-wide upgrade using `apt-get upgrade`, only the MySQL library and development packages are upgraded with newer versions (if available). To upgrade other components including the server, client, test suite, etc., use the `apt-get install` command.

5. The MySQL server always restarts after an update by APT. Prior to MySQL 8.0.16, run `mysql_upgrade` after the server restarts to check and possibly resolve any incompatibilities between the old data and the upgraded software. `mysql_upgrade` also performs other functions; see [mysql_upgrade — Check and Upgrade MySQL Tables](#) for details. As of MySQL 8.0.16, this step is not required, as the server performs all tasks previously handled by `mysql_upgrade`.

Replacing a Native Distribution of MySQL Using the MySQL APT Repository

Variants and forks of MySQL are distributed by different parties through their own software repositories or download sites. You can replace a native distribution of MySQL installed from your Linux platform's software repository with a distribution from the MySQL APT repository in a few steps.

Note

The MySQL APT repository can only replace distributions of MySQL maintained and distributed by Debian or Ubuntu. It cannot replace any MySQL forks found either inside or outside of the distributions' native repositories. To replace such MySQL forks, you have to uninstall them first before you install MySQL using the MySQL APT repository. Follow the instructions for uninstallation from the forks' distributors and, before you proceed, make sure you back up your data and you know how to restore them to a new server.

Warning

A few third-party native repository packages that have dependencies on the native MySQL packages may not work with the MySQL APT repository packages and should not be used together with them; these include `akonadi-backend-mysql`, `handlersocket-mysql-5.5`, and `zoneminder`.

Backing Up Your Database

To avoid loss of data, always back up your database before trying to replace your MySQL installation using the MySQL APT repository. See [Backup and Recovery](#) for instructions.

Adding²the MySQL APT Repository and Selecting a Release Series

Add the MySQL APT repository to your system's repository list and select the release series you want by following the instructions given in [Adding the MySQL APT Repository](#).

Replacing the Native Distribution by an APT Update

By design, the MySQL APT repository replaces your native distribution of MySQL when you perform upgrades on the MySQL packages. To perform the upgrades, follow the same instructions given in [Step 4](#) in [Upgrading MySQL with the MySQL APT Repository](#).

Warning

Once the native distribution of MySQL has been replaced using the MySQL APT repository, purging the old MySQL packages from the native repository using the `apt-get purge`, `apt-get remove --purge`, or `dpkg -P` command might impact the newly installed MySQL server in various ways.

Therefore, *do not purge the old MySQL packages from the native repository packages.*

Replacing a MySQL Server Installed by a Direct `deb` Package Download

`deb` packages from MySQL for installing the MySQL server and its components can either be downloaded from the [MySQL Developer Zone's MySQL Download page](#) or from the MySQL APT repository. The `deb` packages from the two sources are different, and they install and configure MySQL in different ways.

If you have installed MySQL with the MySQL Developer Zone's `deb` packages and now want to replace the installation using the ones from the MySQL APT repository, follow these steps:

1. Back up your database. See [Backup and Recovery](#) for instructions.
2. Follow the steps given previously for [adding the MySQL APT repository](#).
3. Remove the old installation of MySQL by running:

```
shell> sudo dpkg -P mysql
```

4. Install MySQL from the MySQL APT repository:

```
shell> sudo apt-get install mysql-server
```

5. If needed, restore the data on the new MySQL installation. See [Backup and Recovery](#) for instructions.

Removing MySQL with APT

To uninstall the MySQL server and the related components that have been installed using the MySQL APT repository, first, remove the MySQL server using the following command:

```
shell> sudo apt-get remove mysql-server
```

Then, remove any other software that was installed automatically with the MySQL server:

```
shell> sudo apt-get autoremove
```

To uninstall other components, use the following command, replacing `package-name` with the name of the package of the component you want to remove:

```
shell> sudo apt-get remove package-name
```

To see a list of packages you have installed from the MySQL APT repository, use the following command:

```
shell> dpkg -l | grep mysql | grep ii
```

Special Notes on Upgrading the Shared Client Libraries

You can install the shared client libraries from MySQL APT repository by the following command (see [Installing Additional MySQL Products and Components with APT](#) for more details):

```
shell> sudo apt-get install libmysqlclient20
```

If you already have the shared client libraries installed from you Linux platform's software repository, it can be updated by the MySQL APT repository with its own package by using the same command (see [Replacing the Native Distribution by an APT Update](#) for more details).

After updating MySQL using the APT repository, applications compiled with older versions of the shared client libraries should continue to work.

If you recompile applications and dynamically link them with the updated libraries: as typical with new versions of shared libraries, any applications compiled using the updated, newer shared libraries might require those updated libraries on systems where the applications are deployed. If those libraries are not in place, the applications requiring the shared libraries might fail. Therefore, it is recommended that the packages for the shared libraries from MySQL be deployed on those systems. You can do this by adding the MySQL APT repository to the systems (see [Adding the MySQL APT Repository](#)) and installing the latest shared client libraries using the command given at the beginning of this section.

Installing MySQL NDB Cluster Using the APT Repository

Notes

- The MySQL APT repository supports installation of MySQL NDB Cluster only for release 7.5.6 and later, and only for Debian 7.x and 8.x, and Ubuntu 16.04. For other methods of installing NDB Cluster, see [Installation of NDB Cluster on Linux](#), or [Installation of NDB Cluster on Linux](#), depending on the release series that you are using.
- If you already have the MySQL server or MySQL NDB Cluster installed on your system, make sure it is stopped and you have your data and configuration files backed up before proceeding.
- *Known issues:* The MySQL NDB Cluster packages have dependencies on the [python-paramiko](#) and [libclass-methodmaker-perl](#) packages, which are currently *not* installed automatically when you install the NDB Cluster packages; install those required packages yourself with this command before installing the NDB Cluster packages:

```
shell> sudo apt-get install python-paramiko libclass-methodmaker-perl
```

Adding¹the MySQL APT Repository for MySQL NDB Cluster

Follow the steps in [Adding the MySQL APT Repository](#) to add the MySQL APT repository to your system's repository list. During the installation process of the configuration package, when you are asked which MySQL product you want to configure, choose "MySQL Server & Cluster"; when asked which version you wish to receive, choose "mysql-cluster-*x.y*." After returning to the command prompt, go to Step 2 below.

If you already have the configuration package installed on your system, make sure it is up-to-date by running the following command:

```
shell> sudo apt-get install mysql-apt-config
```

Then, use the same method described in [Selecting a Major Release Version](#) to select MySQL NDB Cluster for installation. When you are asked which MySQL product you want to configure, choose "MySQL Server & Cluster"; when asked which version you wish to receive, choose "mysql-cluster-*x.y*." After returning to the command prompt, update package information from the MySQL APT repository with this command:

```
shell> sudo apt-get update
```

Installing² MySQL NDB Cluster

For a minimal installation of MySQL NDB Cluster, follow these steps:

- Install the components for SQL nodes:

```
shell> sudo apt-get install mysql-cluster-community-server
```

You will be asked to provide a password for the root user for your SQL node; see important information on the root password given in [Installing MySQL with APT](#) above. You might also be asked other questions regarding the installation.

- Install the executables for management nodes:

```
shell> sudo apt-get install mysql-cluster-community-management-server
```

- Install the executables for data nodes:

```
shell> sudo apt-get install mysql-cluster-community-data-node
```

Configuring and Starting MySQL NDB Cluster

See [Initial Configuration of NDB Cluster](#) on how to configure MySQL NDB Cluster and [Initial Startup of NDB Cluster](#) on how to start it for the first time. When following those instructions, adjust them according to the following details regarding the SQL nodes of your NDB Cluster installation:

- All configuration files (like `my.cnf`) are under `/etc/mysql`
- All binaries, libraries, headers, etc., are under `/usr/bin` and `/usr/sbin`
- The data directory is `/var/lib/mysql`

Installing Additional MySQL NDB Cluster Products and Components

You can use APT to install individual components and additional products of MySQL NDB Cluster from the MySQL APT repository (see [Available Packages from the MySQL APT Repository](#) for a list). To do that, assuming you already have the MySQL APT repository on your system's repository list (see [Adding the MySQL Yum Repository for MySQL NDB Cluster](#)), follow the same steps given in [Installing Additional MySQL Products and Components with APT](#).

Note

Known issue: Currently, not all components required for running the MySQL NDB Cluster test suite are installed automatically when you install the test suite package (`mysql-cluster-community-test`). Install the following packages with `apt-get install` before you run the test suite:

- `mysql-cluster-community-auto-installer`
- `mysql-cluster-community-management-server`
- `mysql-cluster-community-data-node`
- `mysql-cluster-community-memcached`
- `mysql-cluster-community-java`
- `ndbclient-dev`

Available Packages from the MySQL APT Repository

Table 1 Available Packages from the MySQL APT Repository

Package Name	Description
mysql-server	Metapackage for installing the MySQL server
mysql-community-server	MySQL server
mysql-client	Metapackage for installing the MySQL client
mysql-cluster-community-auto-installer	Auto installer for NDB Cluster
mysql-cluster-community-client	MySQL client for NDB Cluster
mysql-cluster-community-data-node	NDB Cluster data node
mysql-cluster-community-java	NDB Cluster Java drivers
mysql-cluster-community-management-server	NDB Cluster management node
mysql-cluster-community-memcached	NDB Cluster memcached server
mysql-cluster-community-server	MySQL server for NDB Cluster
mysql-cluster-community-source	Source package for NDB Cluster
mysql-cluster-community-test	NDB Cluster testsuite
mysql-community-client	MySQL client
mysql-common	MySQL database common files
libmysqlclient20	MySQL database client library
libmysqlclient-dev	MySQL database development files
libmysqld-dev	MySQL embedded database development files
mysql-testsuite	Metapackage for installing the MySQL test suite
mysql-community-test	MySQL test suite
mysql-community-bench	MySQL benchmark suite
mysql-community-source	MySQL source code
mysql-workbench-community	MySQL Workbench (not available for Debian platforms)
mysql-connector-python-py3	MySQL Connector/Python for Ubuntu 16.04, 18.04, and 18.10; with Python 3.2 or later
mysql-connector-python	MySQL Connector/Python for Debian 8.x with Python 2.6.3 or later, and for Ubuntu 16.04, 18.04, and 18.10; with Python 2.6.3 to 3.1
mysql-router	MySQL Router
ndbclient	NDB Cluster client
ndbclient-dev	NDB Cluster client development library

Appendix A: Adding and Configuring the MySQL APT Repository Manually

Here are the steps for adding manually the MySQL APT repository to your system's software repository list and configuring it, without using the release packages provided by MySQL:

- Download the MySQL GPG Public key (see [Signature Checking Using GnuPG](#) on how to do that) and save it to a file, without adding any spaces or special characters. Then, add the key to your system's GPG keyring with the following command:

```
shell> sudo apt-key add path/to/signature-file
```

- Alternatively, you can download the GPG key to your APT keyring directly using the apt-key utility:

```
shell> sudo apt-key adv --keyserver pgp.mit.edu --recv-keys 5072E1F5
```

- Create a file named `/etc/apt/sources.list.d/mysql.list`, and put into it repository entries in the following format:

```
shell> deb http://repo.mysql.com/apt/{debian|ubuntu}/ {jessie|wheezy|trusty|utopic|vivid} {mysql-5.6|
```

Pick the relevant options for your repository set up:

- Choose “debian” or “ubuntu” according to whether you are on a Debian or Ubuntu platform.
- Choose “jessie,” “wheezy,” “trusty,” “utopic,” or “vivid,” according to the version of your operating system.
- For installing MySQL server, client, and database common files, choose “mysql-5.6” or “mysql-5.7” according to the MySQL version you want. To switch to another release series later, come back and adjust the entry with your new choice.

Note

If you already have a version of MySQL installed on your system, do not choose a lower version at this step, or it might result in an unsupported downgrade operation.

- For installing components like MySQL Workbench, MySQL Utilities, or MySQL Connector Python, create a single entry for each of them, specifying respectively “workbench-6.3,” “utilities-1.6,” or “connector-python-2.1” at the end of each entry.

As an example, for a Ubuntu 16.04 platform, here are the lines in your `mysql.list` files for installing MySQL 5.7 and MySQL Connector/Python from the MySQL APT repository:

```
shell> deb http://repo.mysql.com/apt/ubuntu/ trusty mysql-5.7
shell> deb http://repo.mysql.com/apt/ubuntu/ trusty connector-python-2.1
```

- Use the following command to get the most up-to-date package information from the MySQL APT repository:

```
shell> sudo apt-get update
```

You have configured your system to use the MySQL APT repository and are now ready to continue with [Installing MySQL with APT](#) or [Installing Additional MySQL Products and Components with APT](#).

Appendix B: Installing MySQL Non-interactively with MySQL APT Repository

For noninteractive installations of MySQL with the MySQL APT repository, answer the interactive questions asked by the server package by setting the relevant `debconf` variables in, for example, your installation script. These are the variables to set:

- `root_password`: This is the root password for the server installation.
- `root_password_again`: This is the root password, entered a second time for confirmation.

The following are steps from a sample script that set the `debconf` variables and then install MySQL noninteractively by setting the environment variable `DEBIAN_FRONTEND` to `noninteractive`:

```
shell> sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/root-pass password
shell> sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/re-root-pass password
```

```
shell> sudo DEBIAN_FRONTEND=noninteractive apt-get -y install mysql-server
```

The password variables are erased automatically after MySQL has been installed, so they need to be set again if the installation is to be repeated.

Appendix C: Errors with Missing Entries in the Repository Sources List

Users who configured their systems some time ago with an early and now outdated version of the release package for the MySQL APT repository might see a warning similar to the one below when running the `apt-get update` command:

```
shell> sudo apt-get update
W: Failed to fetch http://repo.mysql.com/apt/ubuntu/dists/trusty/InRelease
Unable to find expected entry 'utilities-1.4/binary-amd64/Packages' in
Release file (Wrong sources.list entry or malformed file)
```

The `apt-get update` command fails with this, and it is no longer possible to update package information for the MySQL APT repository. This is due to outdated entries in the sources list, and you can resolve the problem by the following steps:

1. Delete the file `/etc/apt/sources.list.d/mysql.list` from your system. That removes all sources list entries for the MySQL APT repository.
2. Download and install the latest version of the release package for the MySQL repository by following the steps given in [Adding the MySQL APT Repository](#), including the `sudo apt-get update` step. It reconfigures your system properly for using the MySQL APT repository.

Legal Notices

Copyright © 1997, 2019, 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.

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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

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.

