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 7.x ("wheezy")

- Debian 8.x ("jessie")

- Ubuntu 12.04 LTS ("Precise Pangolin")

- Ubuntu 14.04 LTS ("Trusty Tahr")

- Ubuntu 16.04 ("Xenial Xerus")

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: 2016-06-02 (revision: 47899)

Table of Contents

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.

Adding[1] the MySQL APT Repository

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

a. Go to the download page for the MySQL APT repository at http://dev.mysql.com/downloads/ repo/apt/.

b. Select and download the release package for your Linux distribution.

c. 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, there are two requests by the dialogue boxes:

• 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 The Socket Peer-Credential Authentication Plugin for connections using a Unix socket file. You can set the root password later using the program `mysql_secure_installation`.

• Indicate if you want to install the test database with "Yes" or "No". Installation of the test database is not recommended for production environments.

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 **Apply**. 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 off 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.

   - 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 by the `apt-get upgrade`, only the library and development packages for MySQL will get upgraded with newer

> versions (if available). To upgrade other components including the server, client, test suite, etc., use the `apt-get install` command, as explained precedingly.

5. The MySQL server always restarts after an update by APT. Once the server restarts, you should run `mysql_upgrade` 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.

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 only replaces distributions of MySQL that were installed using the supported Linux platforms' native software repositories. If you have a distribution of MySQL that you have downloaded and installed from a nonnative repository (for example, from MariaDB or Percona), you have to uninstall it before you can reinstall MySQL using the MySQL APT repository. Follow the distributors' instructions for uninstallation and, before you proceed, make sure you have backed 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.

Adding the 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 libmysqlclient18
```

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.

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-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 14.04, 14.10, and 15.04 with Python 3.2 or later
mysql-connector-python	MySQL Connector/Python for Debian 7.x, Debian 8.x, Ubuntu 12.04 with Python 2.6.3 or later, and for Ubuntu 14.04, 14.10, or 15.04 with Python 2.6.3 to 3.1
mysql-utilities	MySQL Utilities (currently not available for Debian 8.x)
mysql-router	MySQL Router

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:

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

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,""precise," "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-6.2," "utilities-1.4," or "connector-python-2.0" at the end of each entry.

As an example, for a Ubuntu 12.04 platform, here are the lines in your `mysql.list` files for installing MySQL 5.6 and MySQL Connector/Python from the MySQL APT repository:

```
deb http://repo.mysql.com/apt/ubuntu/ precise mysql-5.6
deb http://repo.mysql.com/apt/ubuntu/ precise connector-python-2.0
```

- 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: 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