

MySQL 5.7 Reference Manual /  
Installing and Upgrading MySQL / Postinstallation Setup and Testing / Initializing the Data Directory / Initializing the Data Directory Manually Using mysqld

### 2.10.1.1 Initializing the Data Directory Manually Using mysqld

This section describes how to initialize the data directory using **mysqld**, the MySQL server.

#### Note

The procedure described here is available for all platforms as of MySQL 5.7.6. Prior to 5.7.6, use **mysql\_install\_db** on Unix and Unix-like systems (see Section 2.10.1.2, “Initializing the Data Directory Manually Using mysql\_install\_db”). Prior to MySQL 5.7.7, Windows distributions include a data directory with prebuilt tables in the `mysql` database.

The following instructions assume that your current location is the MySQL installation directory, represented here by **BASEDIR**:

```
shell> cd BASEDIR
```

To initialize the data directory, invoke **mysqld** with the `--initialize` or `--initialize-insecure` option, depending on whether you want the server to generate a random initial password for the `'root'@'localhost'` account.

On Windows, use one of these commands:

```
C:\> bin\mysqld --initialize
C:\> bin\mysqld --initialize-insecure
```

On Unix and Unix-like systems, 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, run **mysqld** as `root` and include the `--user` option as shown here:

```
shell> bin/mysqld --initialize --user=mysql
shell> bin/mysqld --initialize-insecure --user=mysql
```

Otherwise, execute the program while logged in as `mysql`, in which case you can omit the `--user` option from the command.

Regardless of platform, use `--initialize` for “secure by default” installation (that is, including generation of a random initial `root` password). In this case, the password is marked as expired and you will need to choose a new one. With the `--initialize-insecure` option, no `root` password is generated; it is assumed that you will assign a password to the account in timely fashion before putting the server into production use.

It might be necessary to specify other options such as `--basedir` or `--datadir` if **mysqld** does not identify the correct locations for the installation directory or data directory. For example (enter the command on one line):

```
shell> bin/mysqld --initialize --user=mysql
        --basedir=/opt/mysql/mysql
        --datadir=/opt/mysql/mysql/data
```

Alternatively, put the relevant option settings in an option file and pass the name of that file to **mysqld**. For Unix and Unix-like systems, suppose that the option file name is `/opt/mysql/mysql/etc/my.cnf`. Put these lines in the file:

```
[mysqld]
basedir=/opt/mysql/mysql
datadir=/opt/mysql/mysql/data
```

Then invoke **mysqld** as follows (enter the command on a single line with the `--defaults-file` option first):

```
shell> bin/mysqld --defaults-file=/opt/mysql/mysql/etc/my.cnf
        --initialize --user=mysql
```

On Windows, suppose that `C:\my.ini` contains these lines:

```
[mysqld]
basedir=C:\\Program Files\\MySQL\\MySQL Server 5.7
datadir=D:\\MySQLdata
```

Then invoke **mysqld** as follows (the `--defaults-file` option must be first):

```
C:\> bin/mysqld --defaults-file=C:\my.ini --initialize
```

When invoked with the `--initialize` or `--initialize-insecure` option, **mysqld** performs the following initialization sequence.

### Note

The server writes any messages to its standard error output. This may be redirected to the error log, so look there if you do not see the messages on your screen. For information about the error log, including where it is located, see Section 6.4.2, “The Error Log”.

On Windows, use the `--console` option to direct messages to the console.

1. The server checks for the existence of the data directory as follows:

- If no data directory exists, the server creates it.
- If a data directory exists and is not empty (that is, it contains files or subdirectories), the server exits after producing an error message:

```
[ERROR] --initialize specified but the data directory exists. Aborting.
```

In this case, remove or rename the data directory and try again.

As of MySQL 5.7.11, an existing data directory is permitted to be nonempty if every entry either has a name that begins with a period (.) or is named using an `--ignore-db-dir` option.

2. Within the data directory, the server creates the `mysql` system database and its tables, including the grant tables, server-side help tables, and time zone tables. For a complete listing and description of the grant tables, see Section 7.2, “The MySQL Access Privilege System”.
3. The server initializes the system tablespace and related data structures needed to manage InnoDB tables.

### Note

After **mysqld** sets up the InnoDB system tablespace, changes to some tablespace characteristics require setting up a whole new instance. This includes the file name of the first file in the system tablespace and the number of undo logs. If you do not want to use the default values, make sure that the settings for the innodb\_data\_file\_path and innodb\_log\_file\_size configuration parameters are in place in the MySQL configuration file before running **mysqld**. Also make sure to specify as necessary other parameters that affect the creation and location of InnoDB files, such as innodb\_data\_home\_dir and innodb\_log\_group\_home\_dir.

If those options are in your configuration file but that file is not in a location that MySQL reads by default, specify the file location using the --defaults-extra-file option when you run **mysqld**.

4. The server creates a 'root'@'localhost' superuser account. The server's action with respect to a password for this account depends on how you invoke it:
  - With --initialize but not --initialize-insecure, the server generates a random password, marks it as expired, and writes a message displaying the password:

```
[Warning] A temporary password is generated for root@localhost:
iTag*AfrH5ej
```

- With --initialize-insecure (either with or without --initialize because --initialize-insecure implies --initialize), the server does not generate a password or mark it expired, and

writes a warning message:

```
Warning] root@localhost is created with an empty password ! Please
consider switching off the --initialize-insecure option.
```

5. The server populates the server-side help tables if content is available (in the `fill_help_tables.sql` file). The server does not populate the time zone tables; to do so, see Section 11.6, “MySQL Server Time Zone Support”.

6. If the `--init-file` option was given to name a file of SQL statements, the server executes the statements in the file. This option enables you to perform custom bootstrapping sequences.

When the server operates in bootstrap mode, some functionality is unavailable that limits the statements permitted in the file. These include statements that relate to account management (such as `CREATE USER` or `GRANT`), replication, and global transaction identifiers.

7. The server exits.

After you initialize the data directory by starting the server with `--initialize` or `--initialize-insecure`, start the server normally (that is, without either of those options) and assign the `'root'@'localhost'` account a new password:

1. Start the server. For instructions, see Section 2.10.2, “Starting the Server”.

2. Connect to the server:

- If you used `--initialize` but not `--initialize-insecure` to initialize the data directory, connect to the server as `root` using the random password that the server generated during the initialization sequence:

```
shell> mysql -u root -p
Enter password: (enter the random root password here)
```

Look in the server error log if you do not know this password.

- If you used `--initialize-insecure` to initialize the data directory, connect to the server as `root` without a password:

```
shell> mysql -u root --skip-password
```

3. After connecting, assign a new `root` password:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

## Note

The data directory initialization sequence performed by the server does not substitute for the actions performed by **mysql\_secure\_installation** or **mysql\_ssl\_rsa\_setup**. See Section 5.4.4, “**mysql\_secure\_installation** — Improve MySQL Installation Security”, and Section 5.4.5, “**mysql\_ssl\_rsa\_setup** — Create SSL/RSA Files”.

---

© 2016, Oracle Corporation and/or its affiliates