

NAME

mysqldump - a database backup program

SYNOPSIS

mysqldump [*options*] [*db_name* [*tbl_name* ...]]

DESCRIPTION

- Performance and Scalability Considerations
- Invocation Syntax
- Option Syntax - Alphabetical Summary

The **mysqldump** client utility performs logical backups, producing a set of SQL statements that can be executed to reproduce the original database object definitions and table data. It dumps one or more MySQL databases for backup or transfer to another SQL server. The **mysqldump** command can also generate output in CSV, other delimited text, or XML format.

mysqldump requires at least the SELECT privilege for dumped tables, SHOW VIEW for dumped views, TRIGGER for dumped triggers, and LOCK TABLES if the **--single-transaction** option is not used. Certain options might require other privileges as noted in the option descriptions.

To reload a dump file, you must have the privileges required to execute the statements that it contains, such as the appropriate CREATE privileges for objects created by those statements.

mysqldump output can include ALTER DATABASE statements that change the database collation. These may be used when dumping stored programs to preserve their character encodings. To reload a dump file containing such statements, the ALTER privilege for the affected database is required.

If you are performing a backup on the server and your tables all are MyISAM tables, you can also use **mysqldump** for this purpose.

Note

A dump made using PowerShell on Windows with output redirection creates a file that has UTF-16 encoding:

```
shell> mysqldump [options] > dump.sql
```

However, UTF-16 is not permitted as a connection character set (see Section 10.4, “Connection Character Sets and Collations”), so the dump file will not load correctly. To work around this issue, use the **--result-file** option, which creates the output in ASCII format:

```
shell> mysqldump [options] --result-file=dump.sql
```

Performance and Scalability Considerations. PP **mysqldump** advantages include the convenience and flexibility of viewing or even editing the output before restoring. You can clone databases for development and DBA work, or produce slight variations of an existing database for testing. It is not intended as a fast or scalable solution for backing up substantial amounts of data. With large data sizes, even if the backup step takes a reasonable time, restoring the data can be very slow because replaying the SQL statements involves disk I/O for insertion, index creation, and so on.

For large-scale backup and restore, a physical backup is more appropriate, to copy the data files in their original format that can be restored quickly:

- If your tables are primarily InnoDB tables, or if you have a mix of InnoDB and MyISAM tables, consider using the **mysqlbackup** command of the MySQL Enterprise Backup product. (Available as part of the Enterprise subscription.) It provides the best performance for InnoDB backups with minimal disruption; it can also back up tables from MyISAM and other storage engines; and it provides a number of convenient options to accommodate different backup scenarios. See Section 25.2, “MySQL Enterprise Backup Overview”.
- If your tables are primarily MyISAM tables, consider using the **mysqldump** instead, for better performance than **mysqldump** of backup and restore operations. See

mysqlhotcopy(1).

mysqldump can retrieve and dump table contents row by row, or it can retrieve the entire content from a table and buffer it in memory before dumping it. Buffering in memory can be a problem if you are dumping large tables. To dump tables row by row, use the **--quick** option (or **--opt**, which enables **--quick**). The **--opt** option (and hence **--quick**) is enabled by default, so to enable memory buffering, use **--skip-quick**.

If you are using a recent version of **mysqldump** to generate a dump to be reloaded into a very old MySQL server, use the **--skip-opt** option instead of the **--opt** or **--extended-insert** option.

For additional information about **mysqldump**, see Section 7.4, “Using mysqldump for Backups”.
Invocation Syntax.PP There are in general three ways to use **mysqldump**—in order to dump a set of one or more tables, a set of one or more complete databases, or an entire MySQL server—as shown here:

```
shell> mysqldump [options] db_name [tbl_name ...]  
shell> mysqldump [options] --databases db_name ...  
shell> mysqldump [options] --all-databases
```

To dump entire databases, do not name any tables following *db_name*, or use the **--databases** or **--all-databases** option.

mysqldump does not dump the INFORMATION_SCHEMA or performance_schema database by default. To dump either of these, name it explicitly on the command line and also use the **--skip-lock-tables** option. You can also name them with the **--databases** option. Before MySQL 5.5 **mysqldump** silently ignores INFORMATION_SCHEMA even if you name it explicitly on the command line.

mysqldump does not dump the performance_schema database.

Before MySQL 5.5.25, **mysqldump** does not dump the general_log or slow_query_log tables for dumps of the mysql database. As of 5.5.25, the dump includes statements to recreate those tables so that they are not missing after reloading the dump file. Log table contents are not dumped.

mysqldump also does not dump the NDB Cluster ndbinfo information database.

To see a list of the options your version of **mysqldump** supports, execute **mysqldump --help**.

Some **mysqldump** options are shorthand for groups of other options:

- Use of **--opt** is the same as specifying **--add-drop-table**, **--add-locks**, **--create-options**, **--disable-keys**, **--extended-insert**, **--lock-tables**, **--quick**, and **--set-charset**. All of the options that **--opt** stands for also are on by default because **--opt** is on by default.
- Use of **--compact** is the same as specifying **--skip-add-drop-table**, **--skip-add-locks**, **--skip-comments**, **--skip-disable-keys**, and **--skip-set-charset** options.

To reverse the effect of a group option, uses its **--skip-xxx** form (**--skip-opt** or **--skip-compact**). It is also possible to select only part of the effect of a group option by following it with options that enable or disable specific features. Here are some examples:

- To select the effect of **--opt** except for some features, use the **--skip** option for each feature. To disable extended inserts and memory buffering, use **--opt --skip-extended-insert --skip-quick**. (Actually, **--skip-extended-insert --skip-quick** is sufficient because **--opt** is on by default.)
- To reverse **--opt** for all features except index disabling and table locking, use **--skip-opt --disable-keys --lock-tables**.

When you selectively enable or disable the effect of a group option, order is important because options are processed first to last. For example, **--disable-keys --lock-tables --skip-opt** would not have the intended effect; it is the same as **--skip-opt** by itself.

mysqldump can retrieve and dump table contents row by row, or it can retrieve the entire content from a table and buffer it in memory before dumping it. Buffering in memory can be a problem if you are dumping large tables. To dump tables row by row, use the **--quick** option (or

--opt, which enables **--quick**). The **--opt** option (and hence **--quick**) is enabled by default, so to enable memory buffering, use **--skip-quick**.

If you are using a recent version of **mysqldump** to generate a dump to be reloaded into a very old MySQL server, you should not use the **--opt** or **--extended-insert** option. Use **--skip-opt** instead.

For additional information about **mysqldump**, see Section 7.4, “Using mysqldump for Backups”. Option Syntax - Alphabetical Summary.PP **mysqldump** supports the following options, which can be specified on the command line or in the [mysqldump] and [client] groups of an option file. For information about option files used by MySQL programs, see Section 4.2.6, “Using Option Files”.

- **--help, -?**

Display a help message and exit.

- **--add-drop-database**

Write a DROP DATABASE statement before each CREATE DATABASE statement. This option is typically used in conjunction with the **--all-databases** or **--databases** option because no CREATE DATABASE statements are written unless one of those options is specified.

- **--add-drop-table**

Write a DROP TABLE statement before each CREATE TABLE statement.

- **--add-drop-trigger**

Write a DROP TRIGGER statement before each CREATE TRIGGER statement.

Note

This option is supported only by **mysqldump** as supplied with NDB Cluster. It is not available when using MySQL Server 5.5.

- **--add-locks**

Surround each table dump with LOCK TABLES and UNLOCK TABLES statements. This results in faster inserts when the dump file is reloaded. See Section 8.2.4.1, “Optimizing INSERT Statements”.

- **--all-databases, -A**

Dump all tables in all databases. This is the same as using the **--databases** option and naming all the databases on the command line.

- **--all-tablespaces, -Y**

Adds to a table dump all SQL statements needed to create any tablespaces used by an NDBCLUSTER table. This information is not otherwise included in the output from **mysqldump**. This option is currently relevant only to NDB Cluster tables.

- **--allow-keywords**

Permit creation of column names that are keywords. This works by prefixing each column name with the table name.

- **--apply-slave-statements**

For a slave dump produced with the **--dump-slave** option, add a STOP SLAVE statement before the CHANGE MASTER TO statement and a START SLAVE statement at the end of the output. This option was added in MySQL 5.5.3.

- **--bind-address=*ip_address***

On a computer having multiple network interfaces, use this option to select which interface to use for connecting to the MySQL server.

This option is supported only in the version of **mysqldump** that is supplied with NDB Cluster. It is not available in standard MySQL Server 5.5 releases.

- **--character-sets-dir=*dir_name***

The directory where character sets are installed. See Section 10.14, “Character Set Configuration”.

- **--comments, -i**

Write additional information in the dump file such as program version, server version, and host. This option is enabled by default. To suppress this additional information, use **--skip-comments**.

- **--compact**

Produce more compact output. This option enables the **--skip-add-drop-table**, **--skip-add-locks**, **--skip-comments**, **--skip-disable-keys**, and **--skip-set-charset** options.

- **--compatible=*name***

Produce output that is more compatible with other database systems or with older MySQL servers. The value of *name* can be `ansi`, `mysql323`, `mysql40`, `postgresql`, `oracle`, `mssql`, `db2`, `maxdb`, `no_key_options`, `no_table_options`, or `no_field_options`. To use several values, separate them by commas. These values have the same meaning as the corresponding options for setting the server SQL mode. See Section 5.1.10, “Server SQL Modes”.

This option does not guarantee compatibility with other servers. It only enables those SQL mode values that are currently available for making dump output more compatible. For example, **--compatible=oracle** does not map data types to Oracle types or use Oracle comment syntax.

- **--complete-insert, -c**

Use complete INSERT statements that include column names.

- **--compress, -C**

Compress all information sent between the client and the server if both support compression.

- **--create-options**

Include all MySQL-specific table options in the CREATE TABLE statements.

- **--databases, -B**

Dump several databases. Normally, **mysqldump** treats the first name argument on the command line as a database name and following names as table names. With this option, it treats all name arguments as database names. CREATE DATABASE and USE statements are included in the output before each new database.

This option may be used to dump the INFORMATION_SCHEMA and performance_schema databases, which normally are not dumped even with the **--all-databases** option. (Also use the **--skip-lock-tables** option.)

- **--debug[=*debug_options*], -# [*debug_options*]**

Write a debugging log. A typical *debug_options* string is `d:t:o,file_name`. The default value is `d:t:o,/tmp/mysqldump.trace`.

- **--debug-check**

Print some debugging information when the program exits.

- **--debug-info**

Print debugging information and memory and CPU usage statistics when the program exits.

- **--default-auth=*plugin***

A hint about the client-side authentication plugin to use. See Section 6.3.6, “Pluggable Authentication”.

This option was added in MySQL 5.5.9.

- **--default-character-set=*charset_name***

Use *charset_name* as the default character set. See Section 10.14, “Character Set Configuration”. If no character set is specified, **mysqldump** uses utf8.

- **--defaults-extra-file=*file_name***

Read this option file after the global option file but (on Unix) before the user option file. If the file does not exist or is otherwise inaccessible, an error occurs. Before MySQL 5.5.8, *file_name* must be the full path name to the file. As of MySQL 5.5.8, the name is interpreted relative to the current directory if given as a relative path name.

- **--defaults-file=*file_name***

Use only the given option file. If the file does not exist or is otherwise inaccessible, an error occurs. Before MySQL 5.5.8, *file_name* must be the full path name to the file. As of MySQL 5.5.8, the name is interpreted relative to the current directory if given as a relative path name.

- **--defaults-group-suffix=*str***

Read not only the usual option groups, but also groups with the usual names and a suffix of *str*. For example, **mysqldump** normally reads the [client] and [mysqldump] groups. If the **--defaults-group-suffix=*_other*** option is given, **mysqldump** also reads the [client_*other*] and [mysqldump_*other*] groups.

- **--delayed-insert**

Write INSERT DELAYED statements rather than INSERT statements.

- **--delete-master-logs**

On a master replication server, delete the binary logs by sending a PURGE BINARY LOGS statement to the server after performing the dump operation. This option automatically enables **--master-data**.

- **--disable-keys, -K**

For each table, surround the INSERT statements with `/*!40000 ALTER TABLE tbl_name DISABLE KEYS */;` and `/*!40000 ALTER TABLE tbl_name ENABLE KEYS */;` statements. This makes loading the dump file faster because the indexes are created after all rows are inserted. This option is effective only for nonunique indexes of MyISAM tables. It has no effect for other tables.

- **--dump-date**

If the **--comments** option is given, **mysqldump** produces a comment at the end of the dump of the following form:

```
-- Dump completed on DATE
```

However, the date causes dump files taken at different times to appear to be different, even if the data are otherwise identical. **--dump-date** and **--skip-dump-date** control whether the date is added to the comment. The default is **--dump-date** (include the date in the comment). **--skip-dump-date** suppresses date printing.

- **--dump-slave[=*value*]**

This option is similar to **--master-data** except that it is used to dump a replication slave server to produce a dump file that can be used to set up another server as a slave that has the same master as the dumped server. It causes the dump output to include a CHANGE MASTER TO statement that indicates the binary log coordinates (file name and position) of the dumped slaves master. These are the master server coordinates from which the slave should start replicating.

--dump-slave causes the coordinates from the master to be used rather than those of the dumped server, as is done by the **--master-data** option. In addition, specifying this option causes the **--master-data** option to be overridden, if used, and effectively ignored.

The option value is handled the same way as for **--master-data** (setting no value or 1 causes a CHANGE MASTER TO statement to be written to the dump, setting 2 causes

the statement to be written but encased in SQL comments) and has the same effect as **--master-data** in terms of enabling or disabling other options and in how locking is handled.

This option causes **mysqldump** to stop the slave SQL thread before the dump and restart it again after.

In conjunction with **--dump-slave**, the **--apply-slave-statements** and **--include-master-host-port** options can also be used.

This option was added in MySQL 5.5.3.

- **--events, -E**

Include Event Scheduler events for the dumped databases in the output. This option requires the EVENT privileges for those databases.

- **--extended-insert, -e**

Write INSERT statements using multiple-row syntax that includes several VALUES lists. This results in a smaller dump file and speeds up inserts when the file is reloaded.

- **--fields-terminated-by=..., --fields-enclosed-by=..., --fields-optionally-enclosed-by=..., --fields-escaped-by=...**

These options are used with the **--tab** option and have the same meaning as the corresponding FIELDS clauses for LOAD DATA INFILE. See Section 13.2.6, “LOAD DATA INFILE Syntax”.

- **--first-slave**

Deprecated. Use **--lock-all-tables** instead. **--first-slave** was removed in MySQL 5.5.3.

- **--flush-logs, -F**

Flush the MySQL server log files before starting the dump. This option requires the RELOAD privilege. If you use this option in combination with the **--all-databases** option, the logs are flushed *for each database dumped*. The exception is when using **--lock-all-tables**, **--master-data**, or (as of MySQL 5.5.21) **--single-transaction**: In this case, the logs are flushed only once, corresponding to the moment that all tables are locked by FLUSH TABLES WITH READ LOCK. If you want your dump and the log flush to happen at exactly the same moment, you should use **--flush-logs** together with **--lock-all-tables**, **--master-data**, or **--single-transaction**.

- **--flush-privileges**

Add a FLUSH PRIVILEGES statement to the dump output after dumping the mysql database. This option should be used any time the dump contains the mysql database and any other database that depends on the data in the mysql database for proper restoration.

- **--force, -f**

Continue even if an SQL error occurs during a table dump.

One use for this option is to cause **mysqldump** to continue executing even when it encounters a view that has become invalid because the definition refers to a table that has been dropped. Without **--force**, **mysqldump** exits with an error message. With **--force**, **mysqldump** prints the error message, but it also writes an SQL comment containing the view definition to the dump output and continues executing.

- **--enable-cleartext-plugin**

Enable the mysql_clear_password cleartext authentication plugin. (See Section 6.5.1.3, “Client-Side Cleartext Pluggable Authentication”.)

This option was added in MySQL 5.5.47.

- **--host=host_name, -h host_name**

Dump data from the MySQL server on the given host. The default host is localhost.

- **--hex-blob**

Dump binary columns using hexadecimal notation (for example, abc becomes 0x616263). The affected data types are BINARY, VARBINARY, the BLOB types, and BIT.

- **--include-master-host-port**

For the CHANGE MASTER TO statement in a slave dump produced with the **--dump-slave** option, add MASTER_HOST and MASTER_PORT options for the host name and TCP/IP port number of the slaves master. This option was added in MySQL 5.5.3.

- **--ignore-table=*db_name.tbl_name***

Do not dump the given table, which must be specified using both the database and table names. To ignore multiple tables, use this option multiple times. This option also can be used to ignore views.

- **--insert-ignore**

Write INSERT IGNORE statements rather than INSERT statements.

- **--lines-terminated-by=...**

This option is used with the **--tab** option and has the same meaning as the corresponding LINES clause for LOAD DATA INFILE. See Section 13.2.6, “LOAD DATA INFILE Syntax”.

- **--lock-all-tables, -x**

Lock all tables across all databases. This is achieved by acquiring a global read lock for the duration of the whole dump. This option automatically turns off **--single-transaction** and **--lock-tables**.

- **--lock-tables, -l**

For each dumped database, lock all tables to be dumped before dumping them. The tables are locked with READ LOCAL to permit concurrent inserts in the case of MyISAM tables. For transactional tables such as InnoDB, **--single-transaction** is a much better option than **--lock-tables** because it does not need to lock the tables at all.

Because **--lock-tables** locks tables for each database separately, this option does not guarantee that the tables in the dump file are logically consistent between databases. Tables in different databases may be dumped in completely different states.

Some options, such as **--opt**, automatically enable **--lock-tables**. If you want to override this, use **--skip-lock-tables** at the end of the option list.

- **--log-error=*file_name***

Log warnings and errors by appending them to the named file. The default is to do no logging.

- **--master-data[=*value*]**

Use this option to dump a master replication server to produce a dump file that can be used to set up another server as a slave of the master. It causes the dump output to include a CHANGE MASTER TO statement that indicates the binary log coordinates (file name and position) of the dumped server. These are the master server coordinates from which the slave should start replicating after you load the dump file into the slave.

If the option value is 2, the CHANGE MASTER TO statement is written as an SQL comment, and thus is informative only; it has no effect when the dump file is reloaded. If the option value is 1, the statement is not written as a comment and takes effect when the dump file is reloaded. If no option value is specified, the default value is 1.

This option requires the RELOAD privilege and the binary log must be enabled.

The **--master-data** option automatically turns off **--lock-tables**. It also turns on **--lock-all-tables**, unless **--single-transaction** also is specified, in which case, a global read lock is acquired only for a short time at the beginning of the dump (see the description for

--single-transaction). In all cases, any action on logs happens at the exact moment of the dump.

It is also possible to set up a slave by dumping an existing slave of the master. In MySQL 5.5.3 and higher, you can create such a dump using the **--dump-slave** option, which overrides **--master-data** and causes it to be ignored if both options are used.

Before MySQL 5.5.3, use the following procedure on the existing slave:

1. Stop the slaves SQL thread and get its current status:

```
mysql> STOP SLAVE SQL THREAD;
```

```
mysql> SHOW SLAVE STATUS;
```

2. From the output of the SHOW SLAVE STATUS statement, the binary log coordinates of the master server from which the new slave should start replicating are the values of the Relay_Master_Log_File and Exec_Master_Log_Pos fields. Denote those values as *file_name* and *file_pos*.

3. Dump the slave server:

```
shell> mysqldump --master-data=2 --all-databases > dumpfile
```

Using **--master-data=2** works only if binary logging has been enabled on the slave. Otherwise, **mysqldump** fails with the error Binlogging on server not active. In this case you must handle any locking issues in another manner, using one or more of **--add-locks**, **--lock-tables**, **--lock-all-tables**, or **--single-transaction**, as required by your application and environment.

4. Restart the slave:

```
mysql> START SLAVE;
```

5. On the new slave, load the dump file:

```
shell> mysql < dumpfile
```

6. On the new slave, set the replication coordinates to those of the master server obtained earlier:

```
mysql> CHANGE MASTER TO
```

```
-> MASTER_LOG_FILE = file_name, MASTER_LOG_POS = file_pos;
```

The CHANGE MASTER TO statement might also need other parameters, such as MASTER_HOST to point the slave to the correct master server host. Add any such parameters as necessary.

- **--no-autocommit**

Enclose the INSERT statements for each dumped table within SET autocommit = 0 and COMMIT statements.

- **--no-create-db, -n**

Suppress the CREATE DATABASE statements that are otherwise included in the output if the **--databases** or **--all-databases** option is given.

- **--no-create-info, -t**

Do not write CREATE TABLE statements that create each dumped table.

Note

This option does *not* exclude statements creating log file groups or tablespaces from **mysqldump** output; however, you can use the **--no-tablespaces** option for this purpose.

- **--no-data, -d**

Do not write any table row information (that is, do not dump table contents). This is useful if you want to dump only the CREATE TABLE statement for the table (for example, to create an empty copy of the table by loading the dump file).

- **--no-defaults**

Do not read any option files. If program startup fails due to reading unknown options from

an option file, **--no-defaults** can be used to prevent them from being read.

- **--no-set-names, -N**

This has the same effect as **--skip-set-charset**.

- **--no-tablespaces, -y**

This option suppresses all CREATE LOGFILE GROUP and CREATE TABLESPACE statements in the output of **mysqldump**.

- **--opt**

This option is shorthand. It is the same as specifying **--add-drop-table --add-locks --create-options --disable-keys --extended-insert --lock-tables --quick --set-charset**. It should give you a fast dump operation and produce a dump file that can be reloaded into a MySQL server quickly.

The --opt option is enabled by default. Use --skip-opt to disable it. See the discussion at the beginning of this section for information about selectively enabling or disabling a subset of the options affected by --opt.

- **--order-by-primary**

Dump each tables rows sorted by its primary key, or by its first unique index, if such an index exists. This is useful when dumping a MyISAM table to be loaded into an InnoDB table, but will make the dump operation take considerably longer.

- **--password[=*password*], -p[*password*]**

The password to use when connecting to the server. If you use the short option form (**-p**), you *cannot* have a space between the option and the password. If you omit the *password* value following the **--password** or **-p** option on the command line, **mysqldump** prompts for one.

Specifying a password on the command line should be considered insecure. See Section 6.1.2.1, “End-User Guidelines for Password Security”. You can use an option file to avoid giving the password on the command line.

- **--pipe, -W**

On Windows, connect to the server using a named pipe. This option applies only if the server supports named-pipe connections.

- **--plugin-dir=*dir_name***

The directory in which to look for plugins. Specify this option if the **--default-auth** option is used to specify an authentication plugin but **mysqldump** does not find it. See Section 6.3.6, “Pluggable Authentication”.

This option was added in MySQL 5.5.9.

- **--port=*port_num*, -P *port_num***

The TCP/IP port number to use for the connection.

- **--print-defaults**

Print the program name and all options that it gets from option files.

- **--protocol={TCP|SOCKET|PIPE|MEMORY}**

The connection protocol to use for connecting to the server. It is useful when the other connection parameters normally would cause a protocol to be used other than the one you want. For details on the permissible values, see Section 4.2.2, “Connecting to the MySQL Server”.

- **--quick, -q**

This option is useful for dumping large tables. It forces **mysqldump** to retrieve rows for a table from the server a row at a time rather than retrieving the entire row set and buffering it in memory before writing it out.

- **--quote-names, -Q**

Quote identifiers (such as database, table, and column names) within ‘ characters. If the ANSI_QUOTES SQL mode is enabled, identifiers are quoted within characters. This option is enabled by default. It can be disabled with **--skip-quote-names**, but this option should be given after any option such as **--compatible** that may enable **--quote-names**.

- **--replace**

Write REPLACE statements rather than INSERT statements.

- **--result-file=***file_name*, **-r** *file_name*

Direct output to the named file. The result file is created and its previous contents overwritten, even if an error occurs while generating the dump.

This option should be used on Windows to prevent newline n characters from being converted to rn carriage return/newline sequences.

- **--routines, -R**

Include stored routines (procedures and functions) for the dumped databases in the output. This option requires the SELECT privilege for the mysql.proc table.

The output generated by using **--routines** contains CREATE PROCEDURE and CREATE FUNCTION statements to create the routines. However, these statements do not include attributes such as the routine creation and modification timestamps, so when the routines are reloaded, they are created with timestamps equal to the reload time.

If you require routines to be created with their original timestamp attributes, do not use **--routines**. Instead, dump and reload the contents of the mysql.proc table directly, using a MySQL account that has appropriate privileges for the mysql database.

Prior to MySQL 5.5.21, this option had no effect when used together with the **--xml** option. (Bug #11760384, Bug #52792)

- **--set-charset**

Write SET NAMES *default_character_set* to the output. This option is enabled by default. To suppress the SET NAMES statement, use **--skip-set-charset**.

- **--shared-memory-base-name=***name*

On Windows, the shared-memory name to use, for connections made using shared memory to a local server. The default value is MYSQL. The shared-memory name is case-sensitive.

The server must be started with the **--shared-memory** option to enable shared-memory connections.

- **--single-transaction**

This option sets the transaction isolation mode to REPEATABLE READ and sends a START TRANSACTION SQL statement to the server before dumping data. It is useful only with transactional tables such as InnoDB, because then it dumps the consistent state of the database at the time when START TRANSACTION was issued without blocking any applications.

When using this option, you should keep in mind that only InnoDB tables are dumped in a consistent state. For example, any MyISAM or MEMORY tables dumped while using this option may still change state.

While a **--single-transaction** dump is in process, to ensure a valid dump file (correct table contents and binary log coordinates), no other connection should use the following statements: ALTER TABLE, CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE. A consistent read is not isolated from those statements, so use of them on a table to be dumped can cause the SELECT that is performed by **mysqldump** to retrieve the table contents to obtain incorrect contents or fail.

The **--single-transaction** option and the **--lock-tables** option are mutually exclusive because LOCK TABLES causes any pending transactions to be committed implicitly.

This option is not supported for NDB Cluster tables; the results cannot be guaranteed to be consistent due to the fact that the NDBCLUSTER storage engine supports only the READ_COMMITTED transaction isolation level. You should always use NDB backup and restore instead.

To dump large tables, combine the **--single-transaction** option with the **--quick** option.

- **--skip-comments**

See the description for the **--comments** option.

- **--skip-opt**

See the description for the **--opt** option.

- **--socket=path, -S path**

For connections to localhost, the Unix socket file to use, or, on Windows, the name of the named pipe to use.

- **--ssl***

Options that begin with **--ssl** specify whether to connect to the server using SSL and indicate where to find SSL keys and certificates. See Section 6.4.2, “Command Options for Encrypted Connections”.

- **--tab=dir_name, -T dir_name**

Produce tab-separated text-format data files. For each dumped table, **mysqldump** creates a *tbl_name.sql* file that contains the CREATE TABLE statement that creates the table, and the server writes a *tbl_name.txt* file that contains its data. The option value is the directory in which to write the files.

Note

This option should be used only when **mysqldump** is run on the same machine as the **mysqld** server. Because the server creates *.txt files in the directory that you specify, the directory must be writable by the server and the MySQL account that you use must have the FILE privilege.

Because **mysqldump** creates *.sql in the same directory, it must be writable by your system login account.

By default, the .txt data files are formatted using tab characters between column values and a newline at the end of each line. The format can be specified explicitly using the **--fields-xxx** and **--lines-terminated-by** options.

Column values are converted to the character set specified by the **--default-character-set** option.

- **--tables**

Override the **--databases** or **-B** option. **mysqldump** regards all name arguments following the option as table names.

- **--triggers**

Include triggers for each dumped table in the output. This option is enabled by default; disable it with **--skip-triggers**.

To be able to dump a tables triggers, you must have the TRIGGER privilege for the table.

- **--tz-utc**

This option enables **TIMESTAMP** columns to be dumped and reloaded between servers in different time zones. **mysqldump** sets its connection time zone to UTC and adds SET TIME_ZONE=+00:00 to the dump file. Without this option, **TIMESTAMP** columns are dumped and reloaded in the time zones local to the source and destination servers, which can cause the values to change if the servers are in different time zones. **--tz-utc** also protects against changes due to daylight saving time. **--tz-utc** is enabled by default. To disable it, use **--skip-tz-utc**.

- **--user=user_name, -u user_name**

The MySQL user name to use when connecting to the server.

- **--verbose, -v**

Verbose mode. Print more information about what the program does.

- **--version, -V**

Display version information and exit.

- **--where=*where_condition*, -w *where_condition***

Dump only rows selected by the given WHERE condition. Quotes around the condition are mandatory if it contains spaces or other characters that are special to your command interpreter.

Examples:

```
--where=user=jimf
-wuserid>1
-wuserid<1
```

- **--xml, -X**

Write dump output as well-formed XML.

NULL, NULL, and Empty Values: For a column named *column_name*, the NULL value, an empty string, and the string value NULL are distinguished from one another in the output generated by this option as follows.

Value:	XML Representation:
NULL (<i>unknown value</i>)	<field name= <i>column_name</i> xsi:nil=true />
	<field name= <i>column_name</i> ></field>
	<field name= <i>column_name</i> >NULL</field>

The output from the **mysql** client when run using the **--xml** option also follows the preceding rules. (See the section called “MYSQL OPTIONS”.)

XML output from **mysqldump** includes the XML namespace, as shown here:

```
shell> mysqldump --xml -u root world City
<?xml version=1.0?>
<mysqldump > -P xmlns:xsi= -- http://www.w3.org/2001/XMLSchema-instance
<database name=world>
<table_structure name=City>
<field Field=ID Type=int(11) Null=NO Key=PRI Extra=auto_increment />
<field Field=Name Type=char(35) Null=NO Key= Default= Extra= />
<field Field=CountryCode Type=char(3) Null=NO Key= Default= Extra= />
<field Field=District Type=char(20) Null=NO Key= Default= Extra= />
<field Field=Population Type=int(11) Null=NO Key= Default=0 Extra= />
<key Table=City Non_unique=0 Key_name=PRIMARY Seq_in_index=1 Column_name=ID
Collation=A Cardinality=4079 Null= Index_type=BTREE Comment= />
<options Name=City Engine=MyISAM Version=10 Row_format=Fixed Rows=4079
Avg_row_length=67 Data_length=273293 Max_data_length=18858823439613951
Index_length=43008 Data_free=0 Auto_increment=4080
Create_time=2007-03-31 01:47:01 Update_time=2007-03-31 01:47:02
Collation=latin1_swedish_ci Create_options= Comment= />
</table_structure>
<table_data name=City>
```

```

<row>
<field name=ID>1</field>
<field name=Name>Kabul</field>
<field name=CountryCode>AFG</field>
<field name=District>Kabol</field>
<field name=Population>1780000</field>
</row>
...
<row>
<field name=ID>4079</field>
<field name=Name>Rafah</field>
<field name=CountryCode>PSE</field>
<field name=District>Rafah</field>
<field name=Population>92020</field>
</row>
</table_data>
</database>
</mysqldump>

```

Prior to MySQL 5.5.21, this option prevented the **--routines** option from working correctly—that is, no stored routines, triggers, or events could be dumped in XML format. (Bug #11760384, Bug #52792)

You can also set the following variables by using **--var_name=value** syntax:

- **max_allowed_packet**

The maximum size of the buffer for client/server communication. The default is 24MB, the maximum is 1GB.

- **net_buffer_length**

The initial size of the buffer for client/server communication. When creating multiple-row INSERT statements (as with the **--extended-insert** or **--opt** option), **mysqldump** creates rows up to **net_buffer_length** bytes long. If you increase this variable, ensure that the MySQL server **net_buffer_length** system variable has a value at least this large.

A common use of **mysqldump** is for making a backup of an entire database:

```
shell> mysqldump db_name > backup-file.sql
```

You can load the dump file back into the server like this:

```
shell> mysql db_name < backup-file.sql
```

Or like this:

```
shell> mysql -e source /path-to-backup/backup-file.sql db_name
```

mysqldump is also very useful for populating databases by copying data from one MySQL server to another:

```
shell> mysqldump --opt db_name | mysql --host=remote_host -C db_name
```

It is possible to dump several databases with one command:

```
shell> mysqldump --databases db_name1 [db_name2 ...] > my_databases.sql
```

To dump all databases, use the **--all-databases** option:

```
shell> mysqldump --all-databases > all_databases.sql
```

For InnoDB tables, **mysqldump** provides a way of making an online backup:

```
shell> mysqldump --all-databases --master-data --single-transaction > all_databases.sql
```

This backup acquires a global read lock on all tables (using FLUSH TABLES WITH READ

LOCK) at the beginning of the dump. As soon as this lock has been acquired, the binary log coordinates are read and the lock is released. If long updating statements are running when the FLUSH statement is issued, the MySQL server may get stalled until those statements finish. After that, the dump becomes lock free and does not disturb reads and writes on the tables. If the update statements that the MySQL server receives are short (in terms of execution time), the initial lock period should not be noticeable, even with many updates.

For point-in-time recovery (also known as “roll-forward,” when you need to restore an old backup and replay the changes that happened since that backup), it is often useful to rotate the binary log (see Section 5.4.4, “The Binary Log”) or at least know the binary log coordinates to which the dump corresponds:

```
shell> mysqldump --all-databases --master-data=2 > all_databases.sql
```

Or:

```
shell> mysqldump --all-databases --flush-logs --master-data=2  
> all_databases.sql
```

The **--master-data** and **--single-transaction** options can be used simultaneously, which provides a convenient way to make an online backup suitable for use prior to point-in-time recovery if tables are stored using the InnoDB storage engine.

For more information on making backups, see Section 7.2, “Database Backup Methods”, and Section 7.3, “Example Backup and Recovery Strategy”.

If you encounter problems backing up views, please read the section that covers restrictions on views which describes a workaround for backing up views when this fails due to insufficient privileges. See Section C.5, “Restrictions on Views”.

COPYRIGHT

Copyright 1997, 2018, Oracle and/or its affiliates. All rights reserved.

This documentation is free software; you can redistribute it and/or modify it only under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License.

This documentation 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 the program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA or see <http://www.gnu.org/licenses/>.

SEE ALSO

For more information, please refer to the MySQL Reference Manual, which may already be installed locally and which is also available online at <http://dev.mysql.com/doc/>.

AUTHOR

Oracle Corporation (<http://dev.mysql.com/>).