



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'mysqlpump.1'

\$ man mysqlpump.1

MYSQLPUMP(1) MySQL Database System MYSQLPUMP(1)

NAME

mysqlpump - a database backup program

SYNOPSIS

mysqlpump [options] [db_name [tbl_name ...]]

DESCRIPTION

- ? mysqlpump Invocation Syntax
- ? mysqlpump Option Summary
- ? mysqlpump Option Descriptions
- ? mysqlpump Object Selection
- ? mysqlpump Parallel Processing
- ? mysqlpump Restrictions

The mysqlpump 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.

Note

mysqlpump is deprecated as of MySQL 8.0.34; expect it to be removed in a future version of MySQL. You can use such MySQL programs as mysqldump and MySQL Shell to perform logical backups, dump databases, and similar tasks instead.

Tip

Consider using the MySQL Shell dump utilities[1], which provide parallel dumping with multiple threads, file compression, and progress information display, as well as cloud features such as Oracle Cloud Infrastructure Object Storage streaming, and MySQL

HeatWave Service compatibility checks and modifications. Dumps can be easily imported into a MySQL Server instance or a MySQL HeatWave Service DB System using the MySQL Shell load dump utilities[2]. Installation instructions for MySQL Shell can be found here[3].

mysqlpump features include:

- ? Parallel processing of databases, and of objects within databases, to speed up the dump process
- ? Better control over which databases and database objects (tables, stored programs, user accounts) to dump
- ? Dumping of user accounts as account-management statements (CREATE USER, GRANT) rather than as inserts into the mysql system database
- ? Capability of creating compressed output
- ? Progress indicator (the values are estimates)
- ? For dump file reloading, faster secondary index creation for InnoDB tables by adding indexes after rows are inserted

Note

mysqlpump uses MySQL features introduced in MySQL 5.7, and thus assumes use with MySQL 5.7 or higher.

mysqlpump 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. The SELECT privilege on the mysql system database is required to dump user definitions. 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.

Note

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

```
mysqlpump [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 cannot be loaded correctly. To work around this issue, use the --result-file option, which creates the

output in ASCII format:

```
mysqlpump [options] --result-file=dump.sql
```

mysqlpump Invocation Syntax

By default, mysqlpump dumps all databases (with certain exceptions noted in [mysqlpump Restrictions](#)). To specify this behavior explicitly, use the `--all-databases` option:

```
mysqlpump --all-databases
```

To dump a single database, or certain tables within that database, name the database on the command line, optionally followed by table names:

```
mysqlpump db_name
```

```
mysqlpump db_name tbl_name1 tbl_name2 ...
```

To treat all name arguments as database names, use the `--databases` option:

```
mysqlpump --databases db_name1 db_name2 ...
```

By default, mysqlpump does not dump user account definitions, even if you dump the mysql system database that contains the grant tables. To dump grant table contents as logical definitions in the form of CREATE USER and GRANT statements, use the `--users` option and suppress all database dumping:

```
mysqlpump --exclude-databases=% --users
```

In the preceding command, % is a wildcard that matches all database names for the `--exclude-databases` option.

mysqlpump supports several options for including or excluding databases, tables, stored programs, and user definitions. See [mysqlpump Object Selection](#).

To reload a dump file, execute the statements that it contains. For example, use the mysql client:

```
mysqlpump [options] > dump.sql
```

```
mysql < dump.sql
```

The following discussion provides additional mysqlpump usage examples.

To see a list of the options mysqlpump supports, issue the command `mysqlpump --help`.

mysqlpump Option Summary

mysqlpump supports the following options, which can be specified on the command line or in the `[mysqlpump]` and `[client]` groups of an option file. (Prior to MySQL 8.0.20, mysqlpump read the `[mysql_dump]` group rather than `[mysqlpump]`. As of 8.0.20, `[mysql_dump]` is still accepted but is deprecated.) For information about option files used by MySQL programs, see [Section 4.2.2.2, ?Using Option Files?](#). [mysqlpump Option Descriptions](#)

? --help, -?

??

?Command-Line Format ? --help ?

??

Display a help message and exit.

? --add-drop-database

??

?Command-Line Format ? --add-drop-database ?

??

Write a DROP DATABASE statement before each CREATE DATABASE statement.

Note

In MySQL 8.0, the mysql schema is considered a system schema that cannot be dropped by end users. If --add-drop-database is used with --all-databases or with --databases where the list of schemas to be dumped includes mysql, the dump file contains a DROP DATABASE `mysql` statement that causes an error when the dump file is reloaded.

Instead, to use --add-drop-database, use --databases with a list of schemas to be dumped, where the list does not include mysql.

? --add-drop-table

??

?Command-Line Format ? --add-drop-table ?

??

Write a DROP TABLE statement before each CREATE TABLE statement.

? --add-drop-user

??

?Command-Line Format ? --add-drop-user ?

??

Write a DROP USER statement before each CREATE USER statement.

? --add-locks

??

?Command-Line Format ? --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 8.2.5.1, "Optimizing INSERT Statements".

This option does not work with parallelism because INSERT statements from different tables can be interleaved and UNLOCK TABLES following the end of the inserts for one table could release locks on tables for which inserts remain.

--add-locks and --single-transaction are mutually exclusive.

? --all-databases, -A

??

?Command-Line Format ? --all-databases ?

??

Dump all databases (with certain exceptions noted in mysqlpump Restrictions). This is the default behavior if no other is specified explicitly.

--all-databases and --databases are mutually exclusive.

Note

See the --add-drop-database description for information about an incompatibility of that option with --all-databases.

Prior to MySQL 8.0, the --routines and --events options for mysqldump and mysqlpump were not required to include stored routines and events when using the --all-databases option: The dump included the mysql system database, and therefore also the mysql.proc and mysql.event tables containing stored routine and event definitions. As of MySQL 8.0, the mysql.event and mysql.proc tables are not used. Definitions for the corresponding objects are stored in data dictionary tables, but those tables are not dumped. To include stored routines and events in a dump made using --all-databases, use the --routines and --events options explicitly.

? --bind-address=ip_address

??

?Command-Line Format ? --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.

? --character-sets-dir=path

??

?Command-Line Format ? --character-sets-dir=dir_name ?

??

?Type ? Directory name ?

??

The directory where character sets are installed. See Section 10.15, ?Character Set Configuration?.

? --column-statistics

??

?Command-Line Format ? --column-statistics ?

??

?Type ? Boolean ?

??

?Default Value ? OFF ?

??

Add ANALYZE TABLE statements to the output to generate histogram statistics for dumped tables when the dump file is reloaded. This option is disabled by default because histogram generation for large tables can take a long time.

? --complete-insert

??

?Command-Line Format ? --complete-insert ?

??

Write complete INSERT statements that include column names.

? --compress, -C

??

?Command-Line Format ? --compress[={OFF|ON}] ?

??

?Deprecated ? 8.0.18 ?

??

?Type ? Boolean ?

??

?Default Value ? OFF ?

??

Compress all information sent between the client and the server if possible. See Section 4.2.8, ?Connection Compression Control?.

As of MySQL 8.0.18, this option is deprecated. Expect it to be removed in a future version of MySQL. See the section called "Configuring Legacy Connection Compression".

? --compress-output=algorithm

??

?Command-Line Format ? --compress-output=algorithm ?

??

?Type ? Enumeration ?

??

?Valid Values ? ?

? ? LZ4 ?

? ? ?

? ? ZLIB ?

??

By default, mysqlpump does not compress output. This option specifies output compression using the specified algorithm. Permitted algorithms are LZ4 and ZLIB. To uncompress compressed output, you must have an appropriate utility. If the system commands lz4 and openssl zlib are not available, MySQL distributions include lz4_decompress and zlib_decompress utilities that can be used to decompress mysqlpump output that was compressed using the --compress-output=LZ4 and --compress-output=ZLIB options. For more information, see lz4_decompress(1), and zlib_decompress(1).

? --compression-algorithms=value

??

?Command-Line Format ? --compression-algorithms=value ?

??

?Introduced ? 8.0.18 ?

??

?Type ? Set ?

??

?Default Value ? uncompressed ?

??

?Valid Values ? ?

? ? zlib ?

? ? ?

? ? zstd ?
? ? ?
? ? uncompressed ?

??

The permitted compression algorithms for connections to the server. The available algorithms are the same as for the protocol_compression_algorithms system variable.

The default value is uncompressed.

For more information, see Section 4.2.8, "Connection Compression Control".

This option was added in MySQL 8.0.18.

? --databases, -B

??

?Command-Line Format ? --databases ?

??

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

--all-databases and --databases are mutually exclusive.

Note

See the --add-drop-database description for information about an incompatibility of that option with --databases.

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

??

?Command-Line Format ? --debug[=debug_options] ?

??

?Type ? String ?

??

?Default Value ? d:t:O,/tmp/mysqlpump.trace ?

??

Write a debugging log. A typical debug_options string is d:t:o,file_name. The default is d:t:O,/tmp/mysqlpump.trace.

This option is available only if MySQL was built using WITH_DEBUG. MySQL release binaries provided by Oracle are not built using this option.

? --debug-check

??

?Command-Line Format ? --debug-check ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

Print some debugging information when the program exits.

This option is available only if MySQL was built using WITH_DEBUG. MySQL release binaries provided by Oracle are not built using this option.

? --debug-info, -T

??

?Command-Line Format ? --debug-info ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

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

This option is available only if MySQL was built using WITH_DEBUG. MySQL release binaries provided by Oracle are not built using this option.

? --default-auth=plugin

??

?Command-Line Format ? --default-auth=plugin ?

??

?Type ? String ?

??

A hint about which client-side authentication plugin to use. See Section 6.2.17, Pluggable Authentication?.

? --default-character-set=charset_name

??

?Command-Line Format ? --default-character- ?

? ? set=charset_name ?

??

?Type ? String ?

??

?Default Value ? utf8 ?

??

Use charset_name as the default character set. See Section 10.15, ?Character Set Configuration?. If no character set is specified, mysqlpump uses utf8mb4.

? --default-parallelism=N

??

?Command-Line Format ? --default-parallelism=N ?

??

?Type ? Integer ?

??

?Default Value ? 2 ?

??

The default number of threads for each parallel processing queue. The default is 2. The --parallel-schemas option also affects parallelism and can be used to override the default number of threads. For more information, see mysqlpump Parallel Processing. With --default-parallelism=0 and no --parallel-schemas options, mysqlpump runs as a single-threaded process and creates no queues.

With parallelism enabled, it is possible for output from different databases to be interleaved.

? --defaults-extra-file=file_name

??

?Command-Line Format ? --defaults-extra-file=file_name ?

??

?Type ? 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.

If file_name is not an absolute path name, it is interpreted relative to the current

directory.

For additional information about this and other option-file options, see Section 4.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --defaults-file=file_name

??

?Command-Line Format ? --defaults-file=file_name ?

??

?Type ? File name ?

??

Use only the given option file. If the file does not exist or is otherwise inaccessible, an error occurs. If file_name is not an absolute path name, it is interpreted relative to the current directory.

Exception: Even with --defaults-file, client programs read .mylogin.cnf.

For additional information about this and other option-file options, see Section 4.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --defaults-group-suffix=str

??

?Command-Line Format ? --defaults-group-suffix=str ?

??

?Type ? String ?

??

Read not only the usual option groups, but also groups with the usual names and a suffix of str. For example, mysqlpump normally reads the [client] and [mysqlpump] groups. If this option is given as --defaults-group-suffix=_other, mysqlpump also reads the [client_other] and [mysqlpump_other] groups.

For additional information about this and other option-file options, see Section 4.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --defer-table-indexes

??

?Command-Line Format ? --defer-table-indexes ?

??

?Type ? Boolean ?

??

?Default Value ? TRUE ?

??

In the dump output, defer index creation for each table until after its rows have been loaded. This works for all storage engines, but for InnoDB applies only for secondary indexes.

This option is enabled by default; use --skip-defer-table-indexes to disable it.

? --events

??

?Command-Line Format ? --events ?

??

?Type ? Boolean ?

??

?Default Value ? TRUE ?

??

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

The output generated by using --events contains CREATE EVENT statements to create the events.

This option is enabled by default; use --skip-events to disable it.

? --exclude-databases=db_list

??

?Command-Line Format ? --exclude-databases=db_list ?

??

?Type ? String ?

??

Do not dump the databases in db_list, which is a list of one or more comma-separated database names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --exclude-events=event_list

??

?Command-Line Format ? --exclude-events=event_list ?

??

?Type ? String ?

??

Do not dump the databases in event_list, which is a list of one or more comma-separated event names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --exclude-routines=routine_list

??

?Command-Line Format ? --exclude-routines=routine_list ?

??

?Type ? String ?

??

Do not dump the events in routine_list, which is a list of one or more comma-separated routine (stored procedure or function) names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --exclude-tables=table_list

??

?Command-Line Format ? --exclude-tables=table_list ?

??

?Type ? String ?

??

Do not dump the tables in table_list, which is a list of one or more comma-separated table names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --exclude-triggers=trigger_list

??

?Command-Line Format ? --exclude-triggers=trigger_list ?

??

?Type ? String ?

??

Do not dump the triggers in trigger_list, which is a list of one or more comma-separated trigger names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --exclude-users=user_list

??

?Command-Line Format ? --exclude-users=user_list ?

??

?Type ? String ?

??

Do not dump the user accounts in user_list, which is a list of one or more comma-separated account names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --extended-insert=N

??

?Command-Line Format ? --extended-insert=N ?

??

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. The option value indicates the number of rows to include in each INSERT statement. The default is 250. A value of 1 produces one INSERT statement per table row.

? --get-server-public-key

??

?Command-Line Format ? --get-server-public-key ?

??

?Type ? Boolean ?

??

Request from the server the public key required for RSA key pair-based password exchange. This option applies to clients that authenticate with the caching_sha2_password authentication plugin. For that plugin, the server does not send the public key unless requested. This option is ignored for accounts that do not authenticate with that plugin. It is also ignored if RSA-based password exchange is not used, as is the case when the client connects to the server using a secure connection.

If --server-public-key-path=file_name is given and specifies a valid public key file, it takes precedence over --get-server-public-key.

For information about the caching_sha2_password plugin, see Section 6.4.1.2, ?Caching SHA-2 Pluggable Authentication?.

? --hex-blob

??

?Command-Line Format ? --hex-blob ?

??

Dump binary columns using hexadecimal notation (for example, 'abc' becomes 0x616263).

The affected data types are BINARY, VARBINARY, BLOB types, BIT, all spatial data types, and other non-binary data types when used with the binary character set.

? --host=host_name, -h host_name

??

?Command-Line Format ? --host ?

??

Dump data from the MySQL server on the given host.

? --include-databases=db_list

??

?Command-Line Format ? --include-databases=db_list ?

??

?Type ? String ?

??

Dump the databases in db_list, which is a list of one or more comma-separated database names. The dump includes all objects in the named databases. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --include-events=event_list

??

?Command-Line Format ? --include-events=event_list ?

??

?Type ? String ?

??

Dump the events in event_list, which is a list of one or more comma-separated event names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --include-routines=routine_list

??

?Command-Line Format ? --include-routines=routine_list ?

??

?Type ? String ?

??

Dump the routines in routine_list, which is a list of one or more comma-separated routine (stored procedure or function) names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --include-tables=table_list

??

?Command-Line Format ? --include-tables=table_list ?

??

?Type ? String ?

??

Dump the tables in table_list, which is a list of one or more comma-separated table names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --include-triggers=trigger_list

??

?Command-Line Format ? --include-triggers=trigger_list ?

??

?Type ? String ?

??

Dump the triggers in trigger_list, which is a list of one or more comma-separated trigger names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --include-users=user_list

??

?Command-Line Format ? --include-users=user_list ?

??

?Type ? String ?

??

Dump the user accounts in user_list, which is a list of one or more comma-separated user names. Multiple instances of this option are additive. For more information, see mysqlpump Object Selection.

? --insert-ignore

??

?Command-Line Format ? --insert-ignore ?

??

Write INSERT IGNORE statements rather than INSERT statements.

? --log-error-file=file_name

??

?Command-Line Format ? --log-error-file=file_name ?

??

?Type ? File name ?

??

Log warnings and errors by appending them to the named file. If this option is not given, mysqlpump writes warnings and errors to the standard error output.

? --login-path=name

??

?Command-Line Format ? --login-path=name ?

??

?Type ? String ?

??

Read options from the named login path in the .mylogin.cnf login path file. A ?login path? is an option group containing options that specify which MySQL server to connect to and which account to authenticate as. To create or modify a login path file, use the mysql_config_editor utility. See mysql_config_editor(1).

For additional information about this and other option-file options, see Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --max-allowed-packet=N

??

?Command-Line Format ? --max-allowed-packet=N ?

??

?Type ? Numeric ?

??

?Default Value ? 25165824 ?

??

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

the maximum is 1GB.

? --net-buffer-length=N

??

?Command-Line Format ? --net-buffer-length=N ?

??

?Type ? Numeric ?

??

?Default Value ? 1047552 ?

??

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

? --no-create-db

??

?Command-Line Format ? --no-create-db ?

??

Suppress any CREATE DATABASE statements that might otherwise be included in the output.

? --no-create-info, -t

??

?Command-Line Format ? --no-create-info ?

??

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

? --no-defaults

??

?Command-Line Format ? --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.

The exception is that the .mylogin.cnf file is read in all cases, if it exists. This permits passwords to be specified in a safer way than on the command line even when

--no-defaults is used. To create .mylogin.cnf, use the mysql_config_editor utility.

See mysql_config_editor(1).

For additional information about this and other option-file options, see Section 4.2.2.3, "Command-Line Options that Affect Option-File Handling".

? --parallel-schemas=[N:]db_list

??

?Command-Line Format ? --parallel- ?

? ? schemas=[N:]schema_list ?

??

?Type ? String ?

??

Create a queue for processing the databases in db_list, which is a list of one or more comma-separated database names. If N is given, the queue uses N threads. If N is not given, the --default-parallelism option determines the number of queue threads.

Multiple instances of this option create multiple queues. mysqlpump also creates a default queue to use for databases not named in any --parallel-schemas option, and for dumping user definitions if command options select them. For more information, see mysqlpump Parallel Processing.

? --password[=password], -p[password]

??

?Command-Line Format ? --password[=password] ?

??

?Type ? String ?

??

The password of the MySQL account used for connecting to the server. The password value is optional. If not given, mysqlpump prompts for one. If given, there must be no space between --password= or -p and the password following it. If no password option is specified, the default is to send no password.

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

To explicitly specify that there is no password and that mysqlpump should not prompt for one, use the --skip-password option.

? --password1[=pass_val] The password for multifactor authentication factor 1 of the MySQL account used for connecting to the server. The password value is optional. If not given, mysqlpump prompts for one. If given, there must be no space between --password1= and the password following it. If no password option is specified, the default is to send no password.

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

To explicitly specify that there is no password and that mysqlpump should not prompt for one, use the --skip-password1 option.

--password1 and --password are synonymous, as are --skip-password1 and --skip-password.

? --password2[=pass_val] The password for multifactor authentication factor 2 of the MySQL account used for connecting to the server. The semantics of this option are similar to the semantics for --password1; see the description of that option for details.

? --password3[=pass_val] The password for multifactor authentication factor 3 of the MySQL account used for connecting to the server. The semantics of this option are similar to the semantics for --password1; see the description of that option for details.

? --plugin-dir=dir_name

??

?Command-Line Format ? --plugin-dir=dir_name ?

??

?Type ? Directory 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 mysqlpump does not find it. See Section 6.2.17, "Pluggable Authentication".

? --port=port_num, -P port_num

??

?Command-Line Format ? --port=port_num ?

??

?Type ? Numeric ?
??
?Default Value ? 3306 ?
??

For TCP/IP connections, the port number to use.

? --print-defaults
??
?Command-Line Format ? --print-defaults ?
??

Print the program name and all options that it gets from option files.
For additional information about this and other option-file options, see
Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --protocol={TCP|SOCKET|PIPE|MEMORY}
??
?Command-Line Format ? --protocol=type ?
??
?Type ? String ?
??
?Default Value ? [see text] ?
??
?Valid Values ? ?

- ? ? TCP ?
- ? ? ?
- ? ? SOCKET ?
- ? ? ?
- ? ? PIPE ?
- ? ? ?
- ? ? MEMORY ?

??
The transport protocol to use for connecting to the server. It is useful when the
other connection parameters normally result in use of a protocol other than the one
you want. For details on the permissible values, see Section 4.2.7, ?Connection
Transport Protocols?.

? --replace

??

?Command-Line Format ? --replace ?

??

Write REPLACE statements rather than INSERT statements.

? --result-file=file_name

??

?Command-Line Format ? --result-file=file_name ?

??

?Type ? 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 \r\n carriage return/newline sequences.

? --routines

??

?Command-Line Format ? --routines ?

??

?Type ? Boolean ?

??

?Default Value ? TRUE ?

??

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

The output generated by using --routines contains CREATE PROCEDURE and CREATE FUNCTION statements to create the routines.

This option is enabled by default; use --skip-routines to disable it.

? --server-public-key-path=file_name

??

?Command-Line Format ? --server-public-key- ?

? ? path=file_name ?

??

?Type ? File name ?

??

The path name to a file in PEM format containing a client-side copy of the public key required by the server for RSA key pair-based password exchange. This option applies to clients that authenticate with the sha256_password or caching_sha2_password authentication plugin. This option is ignored for accounts that do not authenticate with one of those plugins. It is also ignored if RSA-based password exchange is not used, as is the case when the client connects to the server using a secure connection. If --server-public-key-path=file_name is given and specifies a valid public key file, it takes precedence over --get-server-public-key.

For sha256_password, this option applies only if MySQL was built using OpenSSL. For information about the sha256_password and caching_sha2_password plugins, see Section 6.4.1.3, ?SHA-256 Pluggable Authentication?, and Section 6.4.1.2, ?Caching SHA-2 Pluggable Authentication?.

? --set-charset

??

?Command-Line Format ? --set-charset ?

??

Write SET NAMES default_character_set to the output.

This option is enabled by default. To disable it and suppress the SET NAMES statement, use --skip-set-charset.

? --set-gtid-purged=value

??

?Command-Line Format ? --set-gtid-purged=value ?

??

?Type ? Enumeration ?

??

?Default Value ? AUTO ?

??

?Valid Values ? ?

? ? OFF ?

? ? ?

? ? ON ?

? ? ?

? ? AUTO ?

??

This option enables control over global transaction ID (GTID) information written to the dump file, by indicating whether to add a SET @@GLOBAL.gtid_purged statement to the output. This option may also cause a statement to be written to the output that disables binary logging while the dump file is being reloaded.

The following table shows the permitted option values. The default value is AUTO.

??

?Value ? Meaning ?

??

?OFF ? Add no SET statement to the ?

? ? output. ?

??

?ON ? Add a SET statement to the ?

? ? output. An error occurs if ?

? ? GTIDs are not ?

? ? enabled on the server. ?

??

?AUTO ? Add a SET statement to the ?

? ? output if GTIDs are ?

? ? enabled on the ?

? ? server. ?

??

The --set-gtid-purged option has the following effect on binary logging when the dump file is reloaded:

? --set-gtid-purged=OFF: SET @@SESSION.SQL_LOG_BIN=0; is not added to the output.

? --set-gtid-purged=ON: SET @@SESSION.SQL_LOG_BIN=0; is added to the output.

? --set-gtid-purged=AUTO: SET @@SESSION.SQL_LOG_BIN=0; is added to the output if GTIDs are enabled on the server you are backing up (that is, if AUTO evaluates to ON).

? --single-transaction

??

?Command-Line Format ? --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 mysqlpump to retrieve the table contents to obtain incorrect contents or fail.

--add-locks and --single-transaction are mutually exclusive.

? --skip-definer

??

?Command-Line Format ? --skip-definer ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

Omit DEFINER and SQL SECURITY clauses from the CREATE statements for views and stored programs. The dump file, when reloaded, creates objects that use the default DEFINER and SQL SECURITY values. See Section 25.6, ?Stored Object Access Control?.

? --skip-dump-rows, -d

??

?Command-Line Format ? --skip-dump-rows ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

Do not dump table rows.

? --skip-generated-invisible-primary-key

??

?Command-Line Format ? --skip-generated-invisible-

? ? primary-key ?

??

?Introduced ? 8.0.30 ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

This option is available beginning with MySQL 8.0.30, and causes generated invisible primary keys (GIPKs) to be excluded from the dump. See Section 13.1.20.11, ?Generated Invisible Primary Keys?, for more information about GIPKs and GIPK mode.

? --socket=path, -S path

??

?Command-Line Format ? --socket={file_name|pipe_name} ?

??

?Type ? String ?

??

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

On Windows, this option applies only if the server was started with the named_pipe system variable enabled to support named-pipe connections. In addition, the user making the connection must be a member of the Windows group specified by the named_pipe_full_access_group system variable.

? --ssl* Options that begin with --ssl specify whether to connect to the server using encryption and indicate where to find SSL keys and certificates. See the section called ?Command Options for Encrypted Connections?.

? --ssl-fips-mode={OFF|ON|STRICT}

??

?Command-Line Format ? --ssl-fips-mode={OFF|ON|STRICT} ?

??

?Deprecated ? 8.0.34 ?

??

?Type ? Enumeration ?

??

?Default Value ? OFF ?

??

?Valid Values ? ?

? ? OFF ?

? ? ?

? ? ON ?

? ? ?

? ? STRICT ?

??

Controls whether to enable FIPS mode on the client side. The --ssl-fips-mode option differs from other --ssl-xxx options in that it is not used to establish encrypted connections, but rather to affect which cryptographic operations to permit. See Section 6.8, "FIPS Support".

These --ssl-fips-mode values are permitted:

? OFF: Disable FIPS mode.

? ON: Enable FIPS mode.

? STRICT: Enable "strict" FIPS mode.

Note

If the OpenSSL FIPS Object Module is not available, the only permitted value for --ssl-fips-mode is OFF. In this case, setting --ssl-fips-mode to ON or STRICT causes the client to produce a warning at startup and to operate in non-FIPS mode.

As of MySQL 8.0.34, this option is deprecated. Expect it to be removed in a future version of MySQL.

? --tls-ciphersuites=ciphersuite_list

??

?Command-Line Format ? --tls- ?

? ? ciphersuites=ciphersuite_list ?

??

?Introduced ? 8.0.16 ?

??

?Type ? String ?

??

The permissible ciphersuites for encrypted connections that use TLSv1.3. The value is a list of one or more colon-separated ciphersuite names. The ciphersuites that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 6.3.2, 'Encrypted Connection TLS Protocols and Ciphers?.

This option was added in MySQL 8.0.16.

? --tls-version=protocol_list

??

?Command-Line Format ? --tls-version=protocol_list ?

??

?Type ? String ?

??

?Default Value (? 8.0.16) ?

? ? TLSv1,TLSv1.1,TLSv1.2,TLSv1.3 ?

? ? (OpenSSL 1.1.1 or ?

? ? higher) ?

? ? ?

? ? TLSv1,TLSv1.1,TLSv1.2 ?

? ? (otherwise) ?

??

?Default Value (? 8.0.15) ? TLSv1,TLSv1.1,TLSv1.2 ?

??

The permissible TLS protocols for encrypted connections. The value is a list of one or more comma-separated protocol names. The protocols that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 6.3.2, 'Encrypted Connection TLS Protocols and Ciphers?.

? --triggers

??

?Command-Line Format ? --triggers ?

??

?Type ? Boolean ?

??

?Default Value ? TRUE ?

??

Include triggers for each dumped table in the output.

This option is enabled by default; use --skip-triggers to disable it.

? --tz-utc

??

?Command-Line Format ? --tz-utc ?

??

This option enables TIMESTAMP columns to be dumped and reloaded between servers in different time zones. mysqlpump 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.

This option is enabled by default; use --skip-tz-utc to disable it.

? --user=user_name, -u user_name

??

?Command-Line Format ? --user=user_name ?

??

?Type ? String ?

??

The user name of the MySQL account to use for connecting to the server.

If you are using the Rewriter plugin with MySQL 8.0.31 or later, you should grant this user the SKIP_QUERY_REWRITE privilege.

? --users

??

?Command-Line Format ? --users ?

??

?Type ? Boolean ?

??

?Default Value ? FALSE ?

??

Dump user accounts as logical definitions in the form of CREATE USER and GRANT statements.

User definitions are stored in the grant tables in the mysql system database. By default, mysqlpump does not include the grant tables in mysql database dumps. To dump the contents of the grant tables as logical definitions, use the --users option and suppress all database dumping:

```
mysqlpump --exclude-databases=% --users
```

? --version, -V

??

?Command-Line Format ? --version ?

??

Display version information and exit.

? --watch-progress

??

?Command-Line Format ? --watch-progress ?

??

?Type ? Boolean ?

??

?Default Value ? TRUE ?

??

Periodically display a progress indicator that provides information about the completed and total number of tables, rows, and other objects.

This option is enabled by default; use --skip-watch-progress to disable it.

? --zstd-compression-level=level

??

?Command-Line Format ? --zstd-compression-level=# ?

??

?Introduced ? 8.0.18 ?

??

?Type ? Integer ?

??

The compression level to use for connections to the server that use the zstd compression algorithm. The permitted levels are from 1 to 22, with larger values indicating increasing levels of compression. The default zstd compression level is 3.

The compression level setting has no effect on connections that do not use zstd compression.

For more information, see Section 4.2.8, "Connection Compression Control".

This option was added in MySQL 8.0.18.

mysqlpump Object Selection

mysqlpump has a set of inclusion and exclusion options that enable filtering of several object types and provide flexible control over which objects to dump:

? --include-databases and --exclude-databases apply to databases and all objects within them.

? --include-tables and --exclude-tables apply to tables. These options also affect triggers associated with tables unless the trigger-specific options are given.

? --include-triggers and --exclude-triggers apply to triggers.

? --include-routines and --exclude-routines apply to stored procedures and functions. If a routine option matches a stored procedure name, it also matches a stored function of the same name.

? --include-events and --exclude-events apply to Event Scheduler events.

? --include-users and --exclude-users apply to user accounts.

Any inclusion or exclusion option may be given multiple times. The effect is additive.

Order of these options does not matter.

The value of each inclusion and exclusion option is a list of comma-separated names of the appropriate object type. For example:

--exclude-databases=test,world

--include-tables=customer,invoice

Wildcard characters are permitted in the object names:

? % matches any sequence of zero or more characters.

? _ matches any single character.

For example, --include-tables=t%,__tmp matches all table names that begin with t and all five-character table names that end with tmp.

For users, a name specified without a host part is interpreted with an implied host of %.

For example, u1 and u1@% are equivalent. This is the same equivalence that applies in MySQL generally (see Section 6.2.4, "Specifying Account Names").

Inclusion and exclusion options interact as follows:

- ? By default, with no inclusion or exclusion options, `mysqlpump` dumps all databases (with certain exceptions noted in `mysqlpump` Restrictions).
- ? If inclusion options are given in the absence of exclusion options, only the objects named as included are dumped.
- ? If exclusion options are given in the absence of inclusion options, all objects are dumped except those named as excluded.
- ? If inclusion and exclusion options are given, all objects named as excluded and not named as included are not dumped. All other objects are dumped.

If multiple databases are being dumped, it is possible to name tables, triggers, and routines in a specific database by qualifying the object names with the database name. The following command dumps databases `db1` and `db2`, but excludes tables `db1.t1` and `db2.t2`:

```
mysqlpump --include-databases=db1,db2 --exclude-tables=db1.t1,db2.t2
```

The following options provide alternative ways to specify which databases to dump:

- ? The `--all-databases` option dumps all databases (with certain exceptions noted in `mysqlpump` Restrictions). It is equivalent to specifying no object options at all (the default `mysqlpump` action is to dump everything).
- `--include-databases=%` is similar to `--all-databases`, but selects all databases for dumping, even those that are exceptions for `--all-databases`.
- ? The `--databases` option causes `mysqlpump` to treat all name arguments as names of databases to dump. It is equivalent to an `--include-databases` option that names the same databases.

`mysqlpump` Parallel Processing

`mysqlpump` can use parallelism to achieve concurrent processing. You can select concurrency between databases (to dump multiple databases simultaneously) and within databases (to dump multiple objects from a given database simultaneously).

By default, `mysqlpump` sets up one queue with two threads. You can create additional queues and control the number of threads assigned to each one, including the default queue:

- ? `--default-parallelism=N` specifies the default number of threads used for each queue.

In the absence of this option, `N` is 2.

The default queue always uses the default number of threads. Additional queues use the default number of threads unless you specify otherwise.

? `--parallel-schemas=[N:]db_list` sets up a processing queue for dumping the databases named in `db_list` and optionally specifies how many threads the queue uses. `db_list` is a list of comma-separated database names. If the option argument begins with `N:`, the queue uses `N` threads. Otherwise, the `--default-parallelism` option determines the number of queue threads.

Multiple instances of the `--parallel-schemas` option create multiple queues.

Names in the database list are permitted to contain the same `%` and `_` wildcard characters supported for filtering options (see `mysqlpump` Object Selection).

`mysqlpump` uses the default queue for processing any databases not named explicitly with a `--parallel-schemas` option, and for dumping user definitions if command options select them.

In general, with multiple queues, `mysqlpump` uses parallelism between the sets of databases processed by the queues, to dump multiple databases simultaneously. For a queue that uses multiple threads, `mysqlpump` uses parallelism within databases, to dump multiple objects from a given database simultaneously. Exceptions can occur; for example, `mysqlpump` may block queues while it obtains from the server lists of objects in databases.

With parallelism enabled, it is possible for output from different databases to be interleaved. For example, `INSERT` statements from multiple tables dumped in parallel can be interleaved; the statements are not written in any particular order. This does not affect reloading because output statements qualify object names with database names or are preceded by `USE` statements as required.

The granularity for parallelism is a single database object. For example, a single table cannot be dumped in parallel using multiple threads.

Examples:

```
mysqlpump --parallel-schemas=db1,db2 --parallel-schemas=db3
```

`mysqlpump` sets up a queue to process `db1` and `db2`, another queue to process `db3`, and a default queue to process all other databases. All queues use two threads.

```
mysqlpump --parallel-schemas=db1,db2 --parallel-schemas=db3
          --default-parallelism=4
```

This is the same as the previous example except that all queues use four threads.

```
mysqlpump --parallel-schemas=5:db1,db2 --parallel-schemas=3:db3
```

The queue for db1 and db2 uses five threads, the queue for db3 uses three threads, and the default queue uses the default of two threads.

As a special case, with `--default-parallelism=0` and no `--parallel-schemas` options, `mysqlpump` runs as a single-threaded process and creates no queues. `mysqlpump` Restrictions `mysqlpump` does not dump the `performance_schema`, `ndbinfo`, or `sys` schema by default. To dump any of these, name them explicitly on the command line. You can also name them with the `--databases` or `--include-databases` option.

`mysqlpump` does not dump the `INFORMATION_SCHEMA` schema.

`mysqlpump` does not dump InnoDB `CREATE TABLESPACE` statements.

`mysqlpump` dumps user accounts in logical form using `CREATE USER` and `GRANT` statements (for example, when you use the `--include-users` or `--users` option). For this reason, dumps of the `mysql` system database do not by default include the grant tables that contain user definitions: `user`, `db`, `tables_priv`, `columns_priv`, `procs_priv`, or `proxies_priv`. To dump any of the grant tables, name the `mysql` database followed by the table names:

```
mysqlpump mysql user db ...
```

COPYRIGHT

Copyright ? 1997, 2023, Oracle and/or its affiliates.

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/>.

NOTES

1. MySQL Shell dump utilities

<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-dump-instance-schema.html>

2. MySQL Shell load dump utilities

<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-load-dump.html>

3. here

<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html>

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/>).

MySQL 8.0

11/27/2023

MYSQLPUMP(1)