It can be set on slaves independently from the master and may be useful to do so to keep slaves synchronized properly.

`--relay-log=`*`filename`*
: This option sets the root name of the relay log file. By default it's *`slave_host_name-relay-bin`*. MySQL will rotate the log files and append a suffix to the file name given with this option. The suffix is generally a seven digit number, counting from 0000001.

`--relay-log-index=`*`filename`*
: This option sets the name of the relay log index file. By default it's *`slave_host_name-relay-bin.index`*.

`--relay-log-info-file=`*`filename`*
: This option sets the name of the file that the slave will use to record information related to the relay log. By default it's *relay-log.info* and is located in the data directory of MySQL.

`--relay_log_purge[={0|1}]`
: This option is used to make the server automatically purge relay logs when it determines they are no longer necessary. The default value of 1 enables it; a value of 0 disables it.

`--replicate-do-db=`*`database`*
: This option tells the slave thread to limit replication to SQL statements executed against the database given, and only when it is the default database. When the user sets the default database to another database, but executes SQL statements affecting the database given with this option, those statements will not be replicated. Additional databases may be specified with multiple instances of this option.

`--replicate-do-table=`*`database.table`*
: This option tells the slave thread to limit replication to SQL statements executed against the table given. Additional tables may be specified with multiple instances of this option.

`--replicate-ignore-db=`*`database`*
: This option skips replication for SQL statements executed against the database given, but only when it is the default database. So when the user sets the default database to another database, but executes SQL statements affecting the database given with this option, those statements will be replicated. Additional databases may be specified with multiple instances of this option.

`--replicate-ignore-table=`*`database.table`*
: This option omits replication of SQL statements executed against the table given. Additional tables may be specified with multiple instances of this option.

`--replicate-rewrite-db='`*`filename->filename`*`'`
: This option tells the slave to change the database with the first name to have the second name (the name after the `->`), but only when the default database on the master is set to the first database.

`--replicate-same-server-id[={0|1}]`
: If this option is set to 1, entries in the binary log with the same `server-id` as the slave will be replicated. This can potentially cause an infinite loop of replication, so it shouldn't be implemented unless necessary and then only for a limited time and purpose. This option is set to 0 by default and is used on the slave server. The option is ignored if `--log-slave-updates` is enabled.

`--replicate-wild-do-table=`*`database.table`*

This option is similar to `--replicate-do-table` except that you may give wildcards (`%` or `_`) for the database and table names. For instance, to match all tables that start with the name *clients*, you would give a value of `clients%`. To literally give a percent sign or an underscore, escape the character with a preceding backslash (i.e., `\%` and `\_`). Additional tables may be specified with multiple instances of this option.

`--replicate-wild-ignore-table=`*`database.table`*

This option is similar to `--replicate-ignore-table` except that you may give wildcards (`%` or `_`) for the database and table names. For instance, to match all tables that start with the name *clients*, you would give a value of `clients%`. To literally give a percent sign or an underscore, escape the character with a preceding backslash (i.e., `\%` and `\_`). Additional tables may be specified with multiple instances of this option.

`--report-host=`*`host`*

Because the master cannot always ascertain the slave's hostname or IP address, use this option to have the slave register with the master and report its hostname or IP address. This information will be returned when `SHOW SLAVE HOSTS` is executed on the master.

`--report-password=`*`value`*

This option sets the password used by the slave to register with the master. If the `--show-slave-auth-info` option is enabled, this information will be returned when `SHOW SLAVE HOSTS` is executed on the master.

`--report-port=`*`value`*

This option sets the port used by the slave to communicate with the master. It should be employed only when a special port is being used or if the server has special tunneling requirements.

`--report-user=`*`value`*

This option sets the username used by the slave to register with the master. If the `--show-slave-auth-info` option is enabled, this information will be returned when `SHOW SLAVE HOSTS` is executed on the master.

`--server-id=`*`value`*

This option ets the local server's server identifier. It must be used on the master as well as each slave, must be unique for each server, and should be set in the options file.

`--show-slave-auth-info`

This option causes the SQL statement `SHOW SLAVE HOSTS` to reveal the slave's username and password if the slave was started with the `--report-user` and the `--report-password` options.

`--slave_compressed_protocol[={0|1}]`

If set to 1, this option instructs the slave to compress data passed between it and the master, if they support compression. The default is 0.

`--slave_load_tmpdir=`*`value`*

This option specifies the directory where the slave stores temporary files used by the `LOAD DATA INFILE` statement.

`--slave-net-timeout=`*`value`*

This option specifies the number of seconds before a slave connection times out and the slave attempts to reconnect. See the options `--master-connect-retry` and `--master-retry-count` earlier in this chapter, as they relate to this option.

`--slave-skip-errors=error_nbr,...|all`
: By default, replication stops on the slave when an error occurs. This option instructs the slave not to terminate replication for specific errors. Error numbers for the errors should be given in a comma-separated list. You may specify all errors by giving the value of `all`. This option generally should not be used, and the value of `all` in particular should probably never be used.

`--sql-slave-skip-counter=number`
: When the slave begins to re-execute commands that the master executed, this option causes the slave to skip the first *number* events from the master's log.

`--skip-slave-start`
: If this option is enabled, the master server won't automatically start the slaves when it's restarted. Instead, you will have to enter the `START SLAVE` statement on each slave to start it.

`--slave_transaction_retries=value`
: This option specifies the number of times the slave should try to execute a transaction before returning an error if the transaction fails because of problems related to InnoDB or NDB settings. For InnoDB, this applies if there is a deadlock or if the transaction takes more time than is allowed by `innodb_lock_wait_timeout`. For NDB, this applies if the transaction takes more time than is allowed by `TransactionDeadlockDetectionTimeout` or `TransactionInactiveTimeout`. The default value of this option is 10.

Storage engine specific options

An alphabetical list follows of `mysqld` server options recognized by particular storage engines (formerly known as table types). The options are grouped into subsections based on the storage engines: "MyISAM," "InnoDB," and "Other storage engine options," which include MEMORY, MERGE, and NDB (MySQL Cluster).

Older versions of MySQL offered BDB options that are not covered in this book because MySQL no longer supports the BDB storage engine. See the documentation on MySQL's web site for information on BDB options if you're still using BDB tables. It's recommended that you migrate those tables to another storage engine. For a list of storage engines and to see their status on your server, enter `SHOW ENGINES`.

Here are a couple of related options that aren't used for a particular storage engine:

`--default-storage-engine=engine`
: This option specifies the default storage engine. MyISAM is the default unless changed with this option. The server variable associated with this option is `storage_engine`. This option is synonymous with the `--default-table-type` option.

`--default-table-type=engine`
: This option is synonymous with `--default-table-engine`.

MyISAM

These options are related to the MyISAM storage engine, which is typically the default storage engine for MySQL. To determine the default storage engine, enter `SHOW VARIABLES LIKE 'storage_engine';` on the server. You can change the default storage engine with the `--default-storage-engine` option:

`--bulk_insert_buffer_size=value`
See "Performance optimization" later in this chapter.

`--ft_boolean_syntax=value`
This option sets the operators that may be used for FULLTEXT searches of TEXT columns in MyISAM tables. The default operators are: +, -, >, <, (,), ~, *, :, "", &, and |.

`--ft_max_word_len=value`
This option sets the maximum length of a word for which a FULLTEXT search of a table may be made. After setting this option, rebuild the FULLTEXT index by executing `REPAIR TABLE table QUICK;` on the server.

`--ft_min_word_len=value`
Use this option to set the minimum length of a word for which a FULLTEXT search of a table may be made. After setting this option, rebuild the FULLTEXT index by executing `REPAIR TABLE table QUICK;` on the server.

`--ft_query_expansion_limit=value`
This option sets the maximum number of matches for FULLTEXT searches that can be made when using the `WITH QUERY EXPANSION` clause.

`--ft_stopword_file=filename`
This option specifies a text file containing stopwords, which are words not to be considered in FULLTEXT searches. Comments should not be included in this file, only stopwords. A list of words is built into MySQL by default.

`--keep_files_on_create[={0|1}]`
If for some reason a file with the prefix *.MYD* or *.MYI* is located in the data directory of MySQL, but wasn't placed there by the server, and a new table is created with the same name as the prefix of the files, MyISAM will overwrite the files. However, if this option is set to 1, the files won't be overwritten and an error will be returned instead. This option was added as of version 5.1.23 of MySQL.

`--myisam_block_size=value`
This option sets the block size in bytes for index pages in MyISAM.

`--myisam_data_pointer_size=value`
This option sets the default pointer size in bytes for MyISAM tables when tables are created without the `MAX_ROWS` option of the `CREATE TABLE` statement. The default value is 6; valid values range from 2 to 7.

`--myisam_max_extra_sort_file_size=value`
This option is deprecated as of version 5.1 of MySQL.

`--myisam_max_sort_file_size=value`
This option sets the maximum file size in bytes of the temporary file used by MyISAM when recreating a table's index (i.e., when running the `ALTER TABLE`, `LOAD DATA INFILE`, or `REPAIR TABLE` statements). Any space in excess of this value that may be required will be handled in the key cache. The default value is 2 GB.

`--myisam-recover[=value,...]`
This option sets the MyISAM storage engine's recovery mode so that all MyISAM tables will be automatically checked and repaired if needed when the server starts. The choices of settings are `BACKUP` (makes backups of recovered tables that were changed), `DEFAULT` (disables this option), `FORCE` (runs recovery regardless of the risk

of losing data), or QUICK (doesn't check rows for tables without any deletions). Multiple choices may be given in a comma-separated list.

`--myisam_repair_threads[={0|1}]`
With this option enabled, when repairing a table's index each index will be sorted in its own thread. This will potentially increase the speed of the repair process. However, this option is still in beta testing mode. Its default value is 1, enabling the option.

`--myisam_sort_buffer_size=`*value*
This option sets the size of the buffer used for sorting indexes in a MyISAM table. The maximum value for this option is 4 GB, but on 64-bit operating systems, as of version 5.1.23 a larger buffer size may be possible. The variable associated with this option is used when the ALTER TABLE, CREATE INDEX, or REPAIR TABLE statements are executed.

`--myisam_stats_method={nulls_equal|nulls_unequal}`
When aggregate or statistical functions are used, MyISAM has to decide how to treat NULL values for indexes. If this option is set to `nulls_equal`, all NULL values will be considered equal and their associated columns will be grouped together. If `nulls_unequal` is given, each row will be considered a separate and distinct value and they won't be grouped together.

`--myisam_use_mmap`
This option instructs MyISAM to use memory mapping on the underlying operating system when reading from and writing to tables.

InnoDB

These options are related to the InnoDB storage engine, a transactional storage engine:

`--innodb`
This option enables support for the InnoDB storage engine. It is enabled by default. Run the SHOW STORAGE ENGINES; statement on the server to see which storage engines are enabled.

`--innodb_additional_mem_pool_size=`*value*
This option sets the size in bytes of the memory pool used by InnoDB for storing the data dictionary and other internal data structure information. The default value is 1 MB. If this option does not allocate enough memory, InnoDB will write warning messages to the error log.

`--innodb_autoextend_increment=`*value*
This option sets the size in megabytes of increments made to the size of a tablespace in InnoDB when it is automatically extended. The default value is 8 (i.e., 8 MB).

`--innodb_autoinc_lock_mode={0|1|2}`
This option sets the locking mode used when the storage engine generates automatically incremented values. Possible values are 0 (*traditional* mode), 1 (*consecutive* mode), and 2 (*interleaved* mode). The differences are described in the MySQL online manual. In general, processing can get faster under some circumstances as the value of this option gets higher, but results may not always be safe. This option is available as of version 5.1.22 of MySQL.

`--innodb_buffer_pool_awe_mem_mb=value`
: On 32-bit Windows systems, Address Windowing Extensions (AWE) may be available for making use of more than the normal 4 GB memory limit. On such a server, you can use this option to set the amount of AWE memory in megabytes that InnoDB will use for its buffer pool. This option allows for a value of 0 to 63,000. A value of 0 disables it. To take advantage of AWE, you need to recompile MySQL.

`--innodb_buffer_pool_size=value`
: This option sets the size in bytes of the memory buffer used by InnoDB for caching data and indexes.

`--innodb_checksums`
: With this option, which is enabled by default, checksum validation is used on pages read from the filesystem. This provides greater assurance that when data was retrieved there wasn't a problem due to corrupted files or hardware-related trouble. Use the `--skip-innodb-checksums` option to disable it.

`--innodb_commit_concurrency=value`
: This option sets the maximum number of threads that may commit transactions simultaneously. A value of 0 removes the limit on concurrent commits.

`--innodb_data_file_path=path:size...`
: This option allows you to increase the storage space for InnoDB datafiles by specifying names and sizes of datafiles within the directory given with the `--innodb_data_home_dir` option. Each *size* is a number followed by `M` for megabytes or `G` for gigabytes. The minimum total of the file sizes should be 10 MB. If no size is given, a 10 MB datafile with autoextending capability will be used by default. For most operating systems, there is a 4 GB maximum limit.

`--innodb_data_home_dir=path`
: This option specifies the base directory for InnoDB datafiles. If not used, the default will be the data directory for MySQL.

`--innodb_doublewrite`
: This option, enabled by default, causes InnoDB to write the data it receives twice. First it writes data to a buffer, then it writes the data to the filesystem, then it compares the data for integrity. To disable this behavior, use the `--skip-innodb_doublewrite` option.

`--innodb_fast_shutdown[={0|1|2}]`
: This option determines the general procedures that InnoDB follows when shutting down the storage engine. If it is set to 0, the process will go much slower (from minutes to hours longer): it will involve a full purge and a merge of the insert buffer. If this option is set to the default of 1, the process is disabled. If it's set to 2, InnoDB will flush its logs and shut down rapidly. When it's restarted, a crash recovery will be conducted. This option is not allowed on NetWare systems.

`--innodb_file_io_threads=value`
: This option sets the number of file I/O threads permitted. The default value is 4. Changing this on Unix-type systems has no effect. On Windows systems, however, performance may be improved with a higher value.

`--innodb_file_per_table`
: InnoDB uses a shared tablespace by default. When this option is enabled, a separate *.idb* file will be created for each new table to be used for data and indexes instead of using the shared tablespace. By default this is disabled.

`--innodb_flush_log_at_trx_commit={0|1|2}`
: This option determines the procedure for flushing and writing to logs along with transaction commits. If it's set to a value of 0, the log buffer is written to the log file and the log is flushed every second, but not at a transaction commit. If it's set to the default of 1, the log buffer is written to the log file and the log is flushed at every transaction commit. If it's set to 2, the log buffer is written to the log file at each transaction commit and the log is flushed every second without reference to the actual commit. It's recommended generally that this option be left at the default value of 1 and that `--sync_binlog` also be set to 1 to enable it.

`--innodb_flush_method={fdatasync | O_DIRECT | O_DSYNC}`
: This option sets the method of synchronizing data and flushing logs with InnoDB. The default value of `fdatasync` instructs InnoDB to use the operating system's `fsync( )` call to synchronize datafiles and log files. The value of `O_DIRECT` has the server use `O_DIRECT` for opening datafiles and `fsync( )` to synchronize datafiles and log files. This value is available only for Linux, FreeBSD, and Solaris systems. `O_DSYNC` has the server use `O_SYNCH` for opening and flushing log files, but uses `fsync( )` to flush datafiles.

`--innodb_force_recovery=`*`level`*
: This option puts InnoDB in crash recovery mode. The allowable values are 1 through 6. Each level includes all previous levels. Level 1 indicates that the server should continue running even if it finds corrupt pages. Level 2 prevents the main thread from running a purge operation if it would cause the server to crash. A value of 3 prevents transaction rollbacks from being run after the recovery is finished. A setting of 4 prevents operations from the insert buffer from running if they would cause the server to crash. Level 5 causes InnoDB not to consider undo logs when starting and to consider all transactions to have been committed. Finally, level 6 instructs the server not to perform a log roll-forward during the recovery.

`--innodb_lock_wait_timeout=`*`value`*
: This option sets the maximum number of seconds that InnoDB can wait to get a lock on a table before it gives up and rolls back a transaction. The default value is 50.

`--innodb_locks_unsafe_for_binlog`
: To achieve something like row-level locking, InnoDB locks the key for a row. This will also generally prevent other users from writing to the space next to the row that has its key locked. Setting this option to a value of 1 disables this extra protection. Setting it to the default value of 0 protects that next key.

`--innodb_log_arch_dir=`*`value`*
: This option sets the file path where completed log files should be archived. Generally, it should be set to the same directory as the option `--innodb_log_group_home_dir`. Archiving is generally not used, as it's not needed or used for recovery.

`--innodb_log_archive[={0|1}]`
: A value of 1 instructs InnoDB to archive log files. By default, it's set to 0 because it's no longer used.

`--innodb_log_buffer_size=`*`value`*
: This option sets the size in bytes of InnoDB's log buffer. InnoDB writes from the buffer to the log file. The default value is 1 MB.

`--innodb_log_file_size=value`
: This option sets the size in bytes of the log file in a log group to use with InnoDB. The default value is 5 MB. Larger values for this option make recovery slower. The total of all log files normally cannot be more than 4 GB.

`--innodb_log_files_in_group=value`
: This option determines the number of log files in a log group. The default is 2. Log files are written to in a circular manner.

`--innodb_log_group_home_dir[=path]`
: This option sets the file path for InnoDB log files. By default, InnoDB creates two log files in the data directory of MySQL called *ib_logfile0* and *ib_logfile1*.

`--innodb_max_dirty_pages_pct=value`
: In this context, dirty pages are pages that are in the buffer pool but are not yet written to the datafiles. Use this option to set the percentage of dirty pages that may be allowed in the buffer pool. The value given can range from 0 to 100; the default is 90.

`--innodb_max_purge_lag=value`
: This option is related to delays caused by purge operations that are running slowly or are backed up, thus holding up SQL statements that change data. Set the value to the number of such statements that may be delayed during purge operations. The default value of 0 instructs InnoDB not to delay them at all.

`--innodb_mirrored_log_groups=value`
: This option sets the number of mirrored log groups that InnoDB should maintain. By default, this is set to 1 and is usually sufficient.

`--innodb_open_files=value`
: This option sets the maximum number of *.idb* files that may be open at one time. The minimum value is 10; the default is 300. This option applies only when multiple tablespaces are used.

`--innodb-safe-binlog`
: This option ensures consistency between the contents of InnoDB tables and the binary log.

`--innodb_status_file`
: This option has InnoDB keep a status file of the results of the `SHOW ENGINE INNODB STATUS` statement. It writes to the file occasionally. The file is named *innodb_status.pid* and is usually located in the data directory of MySQL.

`--innodb_support_xa`
: This option enables support for a two-phase commit for XA transactions. It's enabled and set to 1 by default. A value of 0 disables it and can sometimes improve performance if the system doesn't use XA transactions.

`--innodb_sync_spin_loops=value`
: This option sets the number of times a thread in InnoDB will wait for a mutex to be free. Once this is exceeded, the thread will be suspended.

`--innodb_table_locks[={0|1}]`
: When enabled (i.e., set to 1), this option causes InnoDB to internally lock a table if the `LOCK TABLE` statement is run and `AUTOCOMMIT` is set to 0.

`--innodb_thread_concurrency=value`
: This option sets the maximum number of threads that can concurrently use InnoDB. Additional threads that try to access InnoDB tables are put into wait mode. The value

can be from 0 to 1,000. Before version 5.1.12 of MySQL, any value over 20 was the same as unlimited. A value of 0 disables the waiting behavior and allows unlimited concurrent threads.

`--innodb_thread_sleep_delay=`*`microseconds`*
: This option sets the number of microseconds that a thread may sleep before being put on a queue. The default value is 10,000; 0 disables sleep.

`--skip-innodb`
: This option disables the InnoDB storage engine.

`--skip-innodb-checksums`
: By default, InnoDB uses checksum validation on pages read from the filesystem (see `--innodb-checksums` earlier in this section). This option disables this behavior.

`--skip-innodb-doublewrite`
: By default, InnoDB writes to a buffer before writing to the filesystem (see `--innodb-doublewrite` earlier in this section). This option disables this behavior.

`--timed_mutexes[={0|1}]`
: When this option is set to 1, the server stores the amount of time InnoDB threads waits for mutexes. The default value of 0 disables this option.

Other storage engine options

These options are recognized by storage engines not previously listed. This section includes MEMORY and NDB specific options for the `mysqld` daemon:

`--max_heap_table_size=`*`value`*
: This option sets the maximum number of rows in a MEMORY table. It applies only to tables created or altered after it's set.

`--ndbcluster`
: This option enables support for the NDB Cluster storage engine.

`--ndb-connectstring=`*`string`*
: This option specifies the connect string that the NDB storage engine uses to create its place in a cluster.

`--skip-merge`
: This option disables the MERGE storage engine. It was added in version 5.1.12 of MySQL.

`--skip-ndbcluster`
: This option disables the NDB Cluster storage engine.

mysqld_multi

mysqld_multi

- `mysqld_multi [`*`options`*`] {start|stop|report} [`*`server_id`*`]`

This option runs multiple MySQL servers on different socket files and ports. To set up multiple servers, you must create a section for each server in a configuration file (e.g., */etc/my.cnf*). The naming scheme for each section must be `[mysqld`*`n`*`]`, where *n* is a different number for each server. You must enter options separately for each server in its

own section, even when servers use the same options. At a minimum, each server should use a different socket file or a different TCP/IP port. You should also use different data directories for each server. The directory should be accessible to the operating system user who started the utility. It should not be the *root* user, though, as this would be a security vulnerability. To see an example of how a configuration file might be set up for multiple servers, enter the following from the command line:

```
mysqld_multi --example
```

Once you have configured multiple servers, you can enter something like the following from the command line to start a server:

```
mysqld_multi start 3
```

This line would start server number 3, listed in the configuration file as `[mysqld3]`. By entering `report` for the first argument, you can obtain the status on the server. For starting and stopping the server, this script uses the `mysqladmin` utility. Here is an alphabetical list of options specific to `mysqld_multi` that you can enter from the command line, along with a brief explanation of each:

`--config-file=`*`filename`*
: This option specifies an alternative server configuration file. As of version 5.1.18 of MySQL, though, this option has been deprecated and is treated like `--defaults-extra-file`.

`--example`
: This option displays a sample configuration file.

`--help`
: This option displays basic help information.

`--log=`*`filename`*
: This option sets the name of the log file. The default is */tmp/mysqld_multi.log*.

`--mysqladmin=`*`filename`*
: This option points to the executable file of the `mysqladmin` utility to invoke.

`--mysqld=`*`filename`*
: This option specifies the MySQL daemon to start, either `mysqld` or `mysqld_safe`. If this is `mysqld`, you should add the `--pid-file` option of `mysqld` so that each server will have a separate process identifier file. If this option is set to `mysqld_safe`, you probably should include the options `ledir` and `mysqld` as they relate to `mysqld_safe`. You would include them in the options file under the server group for the server started by `mysqld_multi`.

`--no-log`
: This option instructs the utility not to save messages to a log, but to send them to `stdout` instead.

`--password=`*`password`*
: This option provides the password for using `mysqladmin`.

`--silent`
: This option disables warning messages from the utility.

`--tcp-ip`
: This option sends this utility's commands to the server using a TCP/IP socket instead of a Unix-domain socket.

`--user=user`
: This option provides the username for using `mysqladmin`. The same user must be used for all servers and must have the `SHUTDOWN` privilege on all of them.

`--version`
: This option displays the version of the utility.

mysqld_safe

mysqld_safe

```
mysqld_safe [options]
```

`mysqld_safe` is recommended utility for starting the MySQL server because the server is restarted automatically if it dies unexpectedly. The utility is available on Unix and Novell NetWare systems.

Although options may be entered from the command line, they should be included in the options file (e.g., *my.cnf*) under the heading `[mysqld_safe]`. Options specific to `mysqld_safe` should not be passed on the command line because they will be passed to the `mysqld` server, which will try to interpret them. Therefore, options are shown here as they would appear in the configuration file, without initial hyphens. `mysqld_safe` can also accept options for the `mysqld` server, but the configuration file is also better for these because it ensures they will be passed to the daemon when it's reloaded after a crash:

`autoclose`
: On Novell NetWare systems, when `mysqld_safe` closes, the related screen does not close automatically without user interaction. Use this option to have the screen close automatically.

`basedir=path`
: This option is necessary and is used to specify the path to the directory where MySQL files are installed.

`core-file-size=value`
: This option sets the maximum size set for the core file to create if the daemon dies.

`datadir=path`
: This option specifies the directory that contains datafiles (i.e., table files).

`defaults-extra-file=filename`
: This option specifies an additional options file to use after the default file is read. When used at the command line, this has to be the first option, except that `--defaults-file` must precede it if used.

`defaults-file=filename`
: This option specifies the default options file for the server; it can be used to substitute special options for the normal default options files. When used at the command line, this has to be the first option given.

`err-log=filename`
: This option specifies the path to the error log for error messages outside the daemon, such as errors when starting.

`ledir=`*`path`*
: This option is necessary for running `mysqld_safe`. It specifies the path where the daemons may be found.

`log-error[=`*`filename`*`]`
: This option enables logging of error messages and server startup messages, optionally specifying a log file. The default log file is *`host.err`* in MySQL's data directory, where *`host`* is the host's name.

`mysqld=`*`daemon`*
: This option is required when using a binary distribution and the data directory for MySQL is not in the location originally set by the distribution. With it you specify which daemon to start (i.e., `mysqld`). This daemon program must be in the same directory given with the `ledir` option.

`mysqld-version=[max]`
: This option specifies which daemon to use by providing the suffix of the daemon's name. A value of `max` starts `mysqld-max`, whereas a blank value ensures `mysqld` is started.

`nice=`*`number`*
: This option employs the `nice` utility to give scheduling priority to the value given.

`no-defaults`
: This option instructs the script not to refer to configuration files for options. When used at the command line, this has to be the first option given.

`open-files-limit=`*`number`*
: This option limits the number of files the daemon may open. Only *root* may use this option.

`pid-file=`*`filename`*
: This option specifies the file that will store the server's process identifier.

`port=port`
: This option specifies the TCP/IP port number to which `mysqld_safe` should listen for incoming connections. Unless started by the *root* filesystem user, the port number should be 1024 or higher.

`skip-kill-mysqld`
: When `mysqld_safe` is started on a Linux system, if this option is not used, any `mysqld` processes that are running will be terminated by it. Use this option to allow existing servers to stay up.

`skip-syslog`
: This option causes the daemon not to log errors to the system's `syslog` facility. The MySQL-specific log will still be written. This option is available as of version 5.1.20 of MySQL. See `syslog` below for more information related to this option.

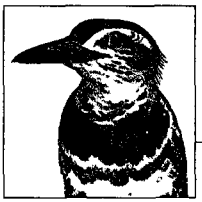
`socket=`*`filename`*
: This option provides the name of the server's socket file for local connections.

`syslog`
: On operating systems that support the `logger` program, this option instructs the daemon to log error messages to the related `syslog`. This option is available as of version 5.1.20 of MySQL. See `skip-syslog` above for more information related to this option.

`syslog-tag`
: When writing error messages to `syslog`, this option marks each message with `mysqld` or `mysqld_safe`, depending on the source of the error. This option is available as of version 5.1.21 of MySQL. See `syslog` and `skip-syslog` previously for more information related to this option.

`timezone=`*`zone`*
: This option sets the environment variable `TZ` for the timezone of the server.

`user=user`
: This option specifies the username or user ID number for the user that starts the server.



16

Command-Line Utilities

This chapter describes the utilities that you can use to administer the MySQL server and data. Some interact with the server, and others manipulate MySQL's datafiles directly. Others can be used to make backups of data (e.g., `mysqldump`). The utilities are listed here in alphabetical order.

Some of these utilities are provided with MySQL and are typically installed in a standard directory for executables so that they are automatically on the user's command path. Other utilities have to be downloaded and installed from MySQL AB's site or from a third-party site.

comp_err

`comp_err` *source destination*

This utility compiles text files that contain mappings of error codes into a format used by MySQL. This is particularly useful for creating error code messages in spoken languages for which error message files do not already exist. You can also use it to modify error messages to your own wording. To do this, just edit the appropriate *errmsg.txt* file in its default directory. For English messages on Unix systems, the source text file and the compiled system file are found typically in */usr/share/mysql/english*. The following demonstrates how to compile a text file containing error messages in Pig Latin:

```
comp_err /usr/share/mysql/piglatin/errmsg.txt \
         /usr/share/mysql/piglatin/errmsg.sys
```

To make the new set of error messages the default set, add the following line to the MySQL configuration file (e.g., *my.cnf* or *my.ini*, depending on your system) under the `[mysqld]` section:

```
language=/usr/share/mysql/piglatin
```

Notice that only the directory is given and not the filename.

Here is a list of options available for this utility in alphabetical order:

`--charset=`*`path`*`, -C` *`path`*
: This option specifies the path to the character set files. The default directory is */usr/local/mysql/sql/share/charsets*, adjusted for the server's installation location.

`--debug[=`*`options`*`], -#` *`options`*`]`
: This option logs debugging information. The set of options used by default is `'d:t:o,logname'`. See Table 16-1 at the end of the list of options under the `mysql dump` utility for an explanation of these flags and others that may be used.

`--debug-info, -T`
: This option writes debugging information and CPU and memory usage information to the log after the utility ends.

`--header_file=`*`filename`*`, -H` *`filename`*
: This option specifies the error header file. By default, it's *mysqld_error.h*.

`--in_file=`*`filename`*`, -F` *`filename`*
: This option specifies the input file. By default, it's */usr/local/mysql/sql/share/errmsg.txt*, adjusted for the server's installation location.

`--name_file=`*`filename`*`, -N` *`filename`*
: This option specifies the error file. By default, it's *mysqld_ername.h*.

`--out_dir=`*`path`*`, -D` *`path`*
: This option specifies the output directory. By default, it's */usr/local/mysql/sql/share*, adjusted for the server's installation location.

`--out_file=`*`filename`*`, -O` *`filename`*
: This option specifies the output file. By default, it's *errmsg.sys*.

`--statefile=`*`filename`*`, -S` *`filename`*
: This option specifies the SQLSTATE header file to be generated. By default, it's *sql_state.h*.

`--version, -V`
: This option returns the version of the utility.

make_binary_distribution

```
make_binary_distribution
```

This utility creates a binary distribution of MySQL from the source code. This can be useful, for instance, to a developer who has modified the source code for her needs and wants to make a customized binary version for her associates to use. Executing the script from the directory containing the modified source code generates a GNU zipped TAR file for distribution.

msql2mysql

```
msql2mysql program.c
```

This utility converts C API function calls querying the mSQL database, in programs written in C, to the MySQL equivalent functions. The only argument is the name of the source to convert. This utility does not create a copy of the source file. Instead, it converts the given source file itself. Therefore, you should make a backup of the source before

issuing the command. This utility isn't always effective in converting all mSQL functions, so manual inspection of the code and testing may be required after a conversion. Note that the `replace` utility is used by `msql2mysql`.

my_print_defaults

`my_print_defaults` *`options filename`*

This utility parses a configuration file, converting key/value pairs into command-line equivalent options. For instance, a line from the *my.cnf* file that reads `basedir=/data/mysql` will be converted to `--basedir=/data/mysql`. To export the MySQL daemon (i.e., *mysqld*) section of *my.cnf* file, enter the following from the command line (the output follows):

```
my_print_defaults --config-file=/etc/my.cnf mysqld
--basedir=/data/mysql
--datadir=/data/mysql
--socket=/tmp/mysql.sock
--tmpdir=/tmp
--log-bin=/data/mysql/logs/log-bin
```

Notice that only the `mysqld` section is parsed and that the header `[mysqld]` and the blank lines are not included in the output. Also, each key/value pair is printed on a separate line. To parse more than one section, you can list additional section names at the end of the command line, separated by spaces.

Here is a list of options available for this utility in alphabetical order:

`--config-file=`*`filename`*`, --defaults-file=`*`filename`*`, -c` *`filename`*
: This option instructs the utility to read only the given configuration or options file (i.e., *my.cnf* or *my.ini*).

`--debug[=`*`options`*`], -# [`*`options`*`]`
: This option logs debugging information. The set of options used by default is `'d:t:o,logname'`. See Table 16-1 at the end of the list of options under the `mysqldump` utility for an explanation of these flags and others that may be used.

`--defaults-extra-file=`*`filename`*`, --extra-file=`*`filename`*`, -e` *`filename`*
: This option instructs the utility to read the given configuration or options file in addition to the default options file (i.e., *my.cnf* or *my.ini*).

`--defaults-group-suffix=`*`suffix`*`, -g` *`suffix`*
: This option instructs the utility to read the options for the groups with the given suffix (e.g., `_clients`).

`--help, -?`
: This option displays basic help information.

`--no-defaults, -n`
: This option indicates that no options file should be used.

`--verbose, -v`
: This option displays more information from the utility.

`--version, -V`
: This option returns the version of the utility.

myisam_ftdump

`myisam_ftdump` *`options table index_nbr`*

This utility displays information related to FULLTEXT indexes on MyISAM tables. It must be run from the server. For the table name, you can give either the name of the table or the name of the table's index file with its path (e.g., */data/mysql/clients.MYI*). The third argument for this utility is the index number. To determine the index number, execute `SHOW INDEX FROM` *`table`*; for the table you want to examine. In the results, the `Non_unique` field will contain the index numbers:

```
myisam_ftdump --stats /data/mysql/russell_dyer/articles.MYI 1

Total rows: 98
Total words: 38517
Unique words: 9961
Longest word: 33 chars (mysql_opt_use_embedded_connection)
Median length: 7
Average global weight: 3.826532
Most common word: 83 times, weight: -1.710790 (make)
```

Here is a list of options available for this utility in alphabetical order:

`--count, -c`
: This option will display a list of all words found in the specified index of the given table with a count of the number of occurrences of each word, along with its weighting in the index.

`--dump, -d`
: This option is used to dump the index, word weighting, and data offsets.

`--length, -l`
: This option returns the distribution length.

`--stats, -s`
: With this option, you can see some statistical information on the index. If no options are given with the utility, this option is assumed.

`--help, -h, -?`
: This option displays basic help information.

`--verbose, -v`
: This option is meant to display more information, but it seems to have no effect on the results at this time.

myisamchk

`myisamchk` *`options table`*`[.MYI][...]`

This utility checks, repairs, and optimizes MyISAM tables. It works with the table files directly and does not require interaction with the MySQL server. Therefore, it may be necessary to specify the path along with the table or table names in the second argument. Also, tables that are being checked should be locked or the MySQL server daemon should be stopped. This utility works with the index files for the tables, so the suffix *.MYI* may be given for table names to prevent it from attempting to analyze other files. Omitting the suffix (e.g., *work_req* instead of *work_req.**) will have the same effect as giving a

specific suffix (*work_req.MYI*). To check all of the tables in a database, use the wildcard (i.e., **.MYI*). Here is a basic example of how you can use `myisamchk` to check one table:

```
myisamchk /data/mysql/workrequests/requests
Checking MyISAM file: /data/mysql/workrequests/requests
Data records:     531   Deleted blocks:       0
myisamchk: warning: 3 clients is using or hasn't closed the table properly
- check file-size
- check key delete-chain
- check record delete-chain
- check index reference
- check data record references index: 1
- check record links
MyISAM-table '/data/mysql/workrequests/requests' is usable but should be
   fixed
```

No options are specified here, so the default of `--check` is used. Notice that `myisamchk` detected a problem with the table. To fix this problem, you can run the utility again, but with the `--recover` option like so:

```
myisamchk --recover /data/mysql/workrequests/requests
- recovering (with sort) MyISAM-table
 '/data/mysql/workrequests/requests'
Data records: 531
- Fixing index 1
```

The following sections list the options available with `myisamchk`.

myisamchk check options

`--check, -c`
: This option checks tables for errors.

`--check-only-changed, -C`
: This option checks only tables that have changed since the last check.

`--extend-check, -e`
: This option checks tables thoroughly. Use it only in extreme cases.

`--fast, -F`
: Use this option to have the utility check only tables that haven't been closed properly.

`--force, -f`
: This option repairs tables that report errors during check mode. It restarts the utility with the `--recover` option if any errors occur.

`--information, -i`
: This option displays statistical information about tables being checked.

`--medium-check, -m`
: This option checks tables more thoroughly than `--check`, but not as thoroughly as `--extend-check`.

`--read-only, -T`
: This option tells the utility not to mark tables with status information so that tables may be used by the utility during its check. Tables are not marked as checked when using this option.

`--update-state, -U`
: This option has the utility update tables to indicate when they were checked and mark them as crashed if any errors are found.

myisamchk repair options

`--backup, -B`
: This option makes copies of datafiles (*table.MYD*), naming them *table-date-time.BAK*.

`--character-sets-dir=path`
: This option sets the directory where character sets are located.

`--correct-checksum`
: This option corrects a table's checksum information.

`--data-file-length=number, -D number`
: This option sets the maximum length of a datafile for rebuilding a full datafile.

`--extend-check, -e`
: This option instructs the utility to attempt to recover all rows, including intentionally deleted ones.

`--force, -f`
: This option instructs the utility to ignore error messages and to overwrite temporary files.

`--keys-used=bitfield, -k bitfield`
: This option instructs the utility to have MyISAM updates use only specific keys for faster data inserts.

`--max-record-length=number`
: This option tells the utility to skip rows larger than the length specified if there is not enough memory.

`--no-symlinks, -l`
: This option instructs the utility not to follow symbolic links at the filesystem level.

`--parallel-recover, -p`
: This option is the same as the `--recover` option, but it creates all keys in parallel using different threads.

`--quick, -q`
: This option repairs only indexes, not datafiles, of uncorrupted tables.

`-qq`
: This option repairs only indexes and updates datafiles only when duplicates are found.

`--recover, -r`
: Use this option to recover a table that has been corrupted. You might also try increasing the variable `sort_buffer_size` with this option. If this option does not work, try `--safe-recover`.

`--safe-recover, -o`
: Use this option if `--recover` fails. It also repairs rows that the `--sort-recover` option cannot handle (e.g., duplicate values for unique keys).

`--set-character-set=set`
: This option specifies the character set to use.

`--set-collation=set`
: This option specifies the collation to use with the utility when sorting table indexes. Execute `SHOW COLLATION;` on the server to retrieve a list of collations that may be used with this option.

`--sort-recover, -n`
: This option instructs the utility to use the sort buffer regardless of whether the temporary file would be too large based on default limits.

`--tmpdir=path, -t path`
: This option specifies the directory used by the utility for temporary files. Multiple directories may be given in a colon-separated list on Unix systems and a semicolon-separated list on Windows systems. By default, this utility uses the value for the environmental variable `TMPDIR`.

`--unpack, -u`
: This option unpacks tables that were packed with the `myisampack` utility.

Other myisamchk options

`--analyze, -a`
: This option optimizes the use of keys in tables. It can help with some joins. Use the `--description` and the `--verbose` options to show the calculated distribution.

`--block-search=offset, -b offset`
: This option searches for a row based on a given offset.

`--description, -d`
: This option displays information about the table.

`--set-auto-increment[=value], -A [value]`
: This option sets the value of an auto-increment column for the next row created. If no value is given, the next value above the highest value found for the column is used.

`--sort-index, -S`
: This option sorts indexes.

`--sort-records=index, -R index`
: This option sorts rows based on the index given.

Global myisamchk options

`--debug[=options], -# [options]`
: This option logs debugging information. The set of options used by default is `'d:t:o,logname'`. See Table 16-1 at the end of the list of options under the `mysqldump` utility for an explanation of these flags and others that may be used.

`--character-sets-dir=path`
: This option specifies the directory containing character sets.

`--help, -?`
: This option displays basic help information.

`--silent, -s`
: This option displays only print error messages. With `-ss` even less information will be displayed.

`--sort-index, -S`
: This option sorts indexes.

`--sort-records=`*value*, `-R` *value*
: This option sorts records based on the index given.

`--tmpdir=`*path*, `-t` *path*
: This option sets the path for temporary files. Additional paths may be given in a colon-separated list.

`--verbose`, `-v`
: This option displays more information. Additional vs (e.g., `-vv`) will provide more information.

`--version`, `-V`
: This option displays the version of the utility.

`--wait`, `-w`
: This option instructs the utility to wait before proceeding if the table is locked.

myisamlog

`myisamlog` *options* [*filename* [*table* ...]]

This utility scans and extracts information from the *myisam.log* file, which logs debugging messages for the MyISAM table handler. The name of the log file may be given. Also, the command can list specific tables to limit scanning to these tables. To activate the log, add the following line to the MySQL server configuration file (e.g., *my.cnf*) under the `[server]` section or the `[mysqld]` section:

```
log-isam=/data/mysql/logs/myisam.log
```

Here is a list of options available for this utility in alphabetical order:

`-?`, `-I`
: This option displays basic help information.

`-c` *number*
: This option limits the output to *number* commands.

`-D`
: Use this option with a server that was compiled with debugging in effect.

`-F` *path*
: This option provides the file path to use. The path should end with a trailing slash.

`-f` *files*
: This option sets the maximum number of open files allowed.

`-i`
: This option displays additional information.

`-o` *offset*
: This option specifies where in the log to begin the scan.

`-P`
: This option displays information about processes.

`-p` *number*
: This option removes the given number of components from the front of the path.

`-R`
: This option displays the current record position.

`-r`
: This option displays recovery activities.

`-u`
: This option displays update activities.

`-V`
: This option displays the version of the utility.

`-v`
: This option displays more information. Additional vs (e.g., `-vv`) will increase the amount of information.

`-w`
: This option displays file write activities.

myisampack

`myisampack` *options /path/table*`[.MYI]`

This utility creates compressed, read-only tables in order to reduce table sizes and to increase retrieval speed. For the table, give the path and table name. Optionally, you can include the *.MYI* file extension with the table name. When reading compressed tables, MySQL decompresses the data in memory. To decompress tables packed with `myisampack`, use `myisamchk` with the `--unpack` option.

Tables that are compressed and later decompressed should be reindexed using `myisamchk`.

A sample run of this utility is:

```
myisamchk --verbose /data/mysql/testing/courses.MYI
```

Here is a list of options available for this utility in alphabetical order:

`--backup, -b`
: This option has the utility create a backup of the given table (*table.OLD*).

`--character-sets-dir=`*filepath*
: This option specifies the directory containing the character sets the utility should use for sorting data.

`--debug[=`*options*`], -# [`*options*`]`
: This option logs debugging information. The set of options used by default is `'d:t:o,logname'`. See Table 16-1 at the end of the list of options under the `mysqldump` utility for an explanation of these flags and others that may be used.

`--force, -f`
: This option forces a compressed table to be created even if the results are larger than the original, and to overwrite a temporary table (*table.TMD*) if it exists.

`--help, -?`
: This option displays basic help information.

`--join=`*table*`, -j` *table*
: This option instructs the utility to join the tables given into one compressed table. The table structures must be identical.

`--packlength=`*bytes*`, -p` *bytes*
: This option sets the size of the pointers for records to the number of bytes given (1, 2, or 3).

`--silent, -s`
: This option suppresses all information except error messages.

`--tmp_dir=path, -T path`
: This option specifies the directory in which to write temporary tables.

`--test, -t`
: This option has the utility test the compression process without actually compressing the table.

`--verbose, -v`
: This option displays information about the compression process.

`--version, -V`
: This option displays the version of the utility.

`--wait, -w`
: This option instructs the utility to wait before compressing if the table is locked by another client or utility.

mysql_convert_table_format

`mysql_convert_table_format options database`

This utility converts all tables in a given database from one storage engine to another. By default it converts them to MyISAM. The program requires that Perl and the Perl DBI module and `DBD::mysql` be installed on the system where it's executed.

mysql_convert_table_format options

`--force`
: This option instructs the utility to keep running despite errors.

`--host=host, -h host`
: This option specifies the host on which to connect and to convert tables.

`--help, -?`
: This option displays help information about the utility.

`--password=password, -p password`
: This option provides the password of the user logging into the server.

`--port=port`
: This option specifies the port on which to connect to the server. The default is 3306.

`--socket=filename, -S filename`
: This option provides the name of the server's socket file.

`--type=engine`
: This option specifies the storage engine to which to convert tables. If not given, MyISAM is assumed.

`--user=user, -u user`
: This option provides the username for logging into the server.

`--verbose`
: This option displays more information from the utility.

`--version`
: This option returns the version of the utility.

mysql_find_rows

`mysql_find_rows` *options filename*

This utility searches a text file containing SQL statements (e.g., a dump file generated by `mysqldump`) for a given pattern and prints the SQL statements it finds. Multiple files may be specified in a comma-separated list:

```
mysql_find_rows --regexp='Graham Greene' < backup.sql > greene_sql_
   statements.txt
```

In this example, the utility will search the dump file *backup.sql* (the redirect for the input is optional) for occurrences of the name of the writer *Graham Greene*. It will write the results—the SQL statements it finds that contain that text—to the *greene_sql_statements.txt* file because of the redirect (i.e., `>`). Otherwise, the results would be displayed on the screen. When creating a dump file that you want to search with this utility, you may want to avoid the `--extended-insert` option (or use `--skip-opt` to disable it) because that option leaves a single `INSERT` statement in the dump file for all rows in the entire table. Your search with this utility would then show all rows for the table, as it returns the whole SQL statement containing the search pattern.

mysql_find_rows options

`--help`
: This option displays help information about the utility.

`--regexp=`*pattern*
: This option specifies the pattern on which the utility is to search the given text file. The pattern is usually entered between quotes. If the option is not given, then the utility will search for `SET` and `USE` statements.

`--rows=`*number*
: This option limits the number of rows of the results. It will return the first rows that it finds, up to the number given.

`--skip-use-db`
: This option instructs the utility not to search for `USE` statements, which it searches for by default.

`--start_row=`*number*
: This option returns rows starting after the given number of rows.

mysql_fix_extensions

`mysql_fix_extensions` *path*

This utility converts the file extensions of the names of MyISAM table files from uppercase to lowercase. The names of MyISAM table files typically end with *.frm*, *.MYD*, and *.MYI*. This utility changes the names of the last two types to *.myd* and *.myi*, respectively. This utility may be necessary when moving database files from servers running on an operating system that is case-insensitive (e.g., Windows) to one that is case-sensitive (e.g., Linux). You need to give the path to the directory for the data, that is, the directory where the database subdirectories are located.