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 'mysqld_safe.1'

$ man mysqld_safe.1

MYSQLD_SAFE(1) MySQL Database System MYSQLD_SAFE(1)

NAME

mysqld_safe - MySQL server startup script

SYNOPSIS

mysqld_safe options

DESCRIPTION

mysqld_safe is the recommended way to start a mysqld server on Unix. mysqld_safe adds some safety features such as restarting the server when an error occurs and logging runtime information to an error log. A description of error logging is given later in this section.

Note

For some Linux platforms, MySQL installation from RPM or Debian packages includes systemd support for managing MySQL server startup and shutdown. On these platforms, mysqld_safe is not installed because it is unnecessary. For more information, see Section 2.5.9, ?Managing MySQL Server with systemd?.

One implication of the non-use of mysqld_safe on platforms that use systemd for server management is that use of [mysqld_safe] or [safe_mysqld] sections in option files is not supported and might lead to unexpected behavior.

mysqld_safe tries to start an executable named mysqld. To override the default behavior and specify explicitly the name of the server you want to run, specify a --mysqld or --mysqld-version option to mysqld_safe. You can also use --ledir to indicate the directory where mysqld_safe should look for the server.

Many of the options to mysqld_safe are the same as the options to mysqld. See

Section 5.1.7, ?Server Command Options?.

Options unknown to mysqld_safe are passed to mysqld if they are specified on the command line, but ignored if they are specified in the [mysqld_safe] group of an option file. See Section 4.2.2.2, ?Using Option Files?.

mysqld_safe reads all options from the [mysqld], [server], and [mysqld_safe] sections in option files. For example, if you specify a [mysqld] section like this, mysqld_safe finds and uses the --log-error option:

```
[mysqld]
log-error=error.log
```

For backward compatibility, mysqld_safe also reads [safe_mysqld] sections, but to be current you should rename such sections to [mysqld_safe].

mysqld_safe accepts options on the command line and in option files, as described in the following table. For information about option files used by MySQL programs, see Section 4.2.2.2, ?Using Option Files?.

? --help

```
????????????????????????????????
?Command-Line Format ? --help ?
????????????????????????????????
```

Display a help message and exit.

? --basedir=dir_name

```
????????????????????????????????????????????
?Command-Line Format ? --basedir=dir_name ?
????????????????????????????????????????????
?Type                ? Directory name     ?
????????????????????????????????????????????
```

The path to the MySQL installation directory.

? --core-file-size=size

```
???????????????????????????????????????????????
?Command-Line Format ? --core-file-size=size ?
???????????????????????????????????????????????
?Type                ? String                ?
???????????????????????????????????????????????
```

The size of the core file that mysqld should be able to create. The option value is passed to ulimit -c.

Note

The innodb_buffer_pool_in_core_file variable can be used to reduce the size of core files on operating systems that support it. For more information, see Section 15.8.3.7, ?Excluding Buffer Pool Pages from Core Files?.

? --datadir=dir_name

```
???????????????????????????????????????????
?Command-Line Format ? --datadir=dir_name ?
???????????????????????????????????????????
?Type                ? Directory name     ?
???????????????????????????????????????????
```

The path to the data directory.

? --defaults-extra-file=file_name

```
????????????????????????????????????????????????????????
?Command-Line Format ? --defaults-extra-file=file_name ?
????????????????????????????????????????????????????????
?Type                ? File name                       ?
????????????????????????????????????????????????????????
```

Read this option file in addition to the usual option files. If the file does not exist or is otherwise inaccessible, the server exits with an error. If file_name is not an absolute path name, it is interpreted relative to the current directory. This must be the first option on the command line if it is used.

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, the server exits with an error. If file_name is not an absolute path name, it is interpreted relative to the current directory. This must be the first option on the command line if it is used.

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

? --ledir=dir_name

```
?????????????????????????????????????????
?Command-Line Format ? --ledir=dir_name ?
?????????????????????????????????????????
?Type                ? Directory name   ?
?????????????????????????????????????????
```

If mysqld_safe cannot find the server, use this option to indicate the path name to the directory where the server is located.

This option is accepted only on the command line, not in option files. On platforms that use systemd, the value can be specified in the value of MYSQLD_OPTS. See Section 2.5.9, ?Managing MySQL Server with systemd?.

? --log-error=file_name

```
??????????????????????????????????????????????
?Command-Line Format ? --log-error=file_name ?
??????????????????????????????????????????????
?Type                ? File name             ?
??????????????????????????????????????????????
```

Write the error log to the given file. See Section 5.4.2, ?The Error Log?.

? --mysqld-safe-log-timestamps

```
???????????????????????????????????????????
?Command-Line Format ? --mysqld-safe-log- ?
?                    ? timestamps=type    ?
???????????????????????????????????????????
?Type                ? Enumeration        ?
???????????????????????????????????????????
?Default Value       ? utc                ?
???????????????????????????????????????????
?Valid Values        ?                    ?
?                    ?          system    ?
?                    ?                    ?
?                    ?           hyphen  ?
?                    ?                    ?
?                    ?           legacy  ?
??????????????????????????????????????????
```

This option controls the format for timestamps in log output produced by mysqld_safe.

The following list describes the permitted values. For any other value, mysqld_safe

logs a warning and uses UTC format.

? UTC, utc

ISO 8601 UTC format (same as --log_timestamps=UTC for the server). This is the

default.

? SYSTEM, system

ISO 8601 local time format (same as --log_timestamps=SYSTEM for the server).

? HYPHEN, hyphen

YY-MM-DD h:mm:ss format, as in mysqld_safe for MySQL 5.6.

? LEGACY, legacy

YYMMDD hh:mm:ss format, as in mysqld_safe prior to MySQL 5.6.

? --malloc-lib=[lib_name]

```
???????????????????????????????????????????????
?Command-Line Format ? --malloc-lib=[lib-name] ?
???????????????????????????????????????????????
?Type                ? String                  ?
???????????????????????????????????????????????
```

The name of the library to use for memory allocation instead of the system malloc()

library. The option value must be one of the directories /usr/lib, /usr/lib64,

/usr/lib/i386-linux-gnu, or /usr/lib/x86_64-linux-gnu.

The --malloc-lib option works by modifying the LD_PRELOAD environment value to affect

dynamic linking to enable the loader to find the memory-allocation library when mysqld

runs:

? If the option is not given, or is given without a value (--malloc-lib=),

LD_PRELOAD is not modified and no attempt is made to use tcmalloc.

? Prior to MySQL 8.0.21, if the option is given as --malloc-lib=tcmalloc,

mysqld_safe looks for a tcmalloc library in /usr/lib. If tmalloc is found, its

path name is added to the beginning of the LD_PRELOAD value for mysqld. If tcmalloc is not found, mysqld_safe aborts with an error.

As of MySQL 8.0.21, tcmalloc is not a permitted value for the --malloc-lib option.

? If the option is given as --malloc-lib=/path/to/some/library, that full path is added to the beginning of the LD_PRELOAD value. If the full path points to a nonexistent or unreadable file, mysqld_safe aborts with an error.

? For cases where mysqld_safe adds a path name to LD_PRELOAD, it adds the path to the beginning of any existing value the variable already has.

Note

On systems that manage the server using systemd, mysqld_safe is not available. Instead, specify the allocation library by setting LD_PRELOAD in /etc/sysconfig/mysql.

Linux users can use the libtcmalloc_minimal.so library on any platform for which a tcmalloc package is installed in /usr/lib by adding these lines to the my.cnf file:

```
[mysqld_safe]
malloc-lib=tcmalloc
```

To use a specific tcmalloc library, specify its full path name. Example:

```
[mysqld_safe]
malloc-lib=/opt/lib/libtcmalloc_minimal.so
```

? --mysqld=prog_name

??

?Command-Line Format ? --mysqld=file_name ?

??

?Type ? File name ?

??

The name of the server program (in the ledir directory) that you want to start. This option is needed if you use the MySQL binary distribution but have the data directory outside of the binary distribution. If mysqld_safe cannot find the server, use the --ledir option to indicate the path name to the directory where the server is located.

This option is accepted only on the command line, not in option files. On platforms that use systemd, the value can be specified in the value of MYSQLD_OPTS. See Section 2.5.9, ?Managing MySQL Server with systemd?.

? --mysqld-version=suffix

???

```
?Command-Line Format ? --mysqld-version=suffix ?
???????????????????????????????????????????????
?Type               ? String                  ?
???????????????????????????????????????????????
```

This option is similar to the --mysqld option, but you specify only the suffix for the server program name. The base name is assumed to be mysqld. For example, if you use --mysqld-version=debug, mysqld_safe starts the mysqld-debug program in the ledir directory. If the argument to --mysqld-version is empty, mysqld_safe uses mysqld in the ledir directory.

This option is accepted only on the command line, not in option files. On platforms that use systemd, the value can be specified in the value of MYSQLD_OPTS. See Section 2.5.9, ?Managing MySQL Server with systemd?.

? --nice=priority

```
???????????????????????????????????????
?Command-Line Format ? --nice=priority ?
???????????????????????????????????????
?Type               ? Numeric         ?
???????????????????????????????????????
```

Use the nice program to set the server's scheduling priority to the given value.

? --no-defaults

```
?????????????????????????????????????
?Command-Line Format ? --no-defaults ?
?????????????????????????????????????
?Type               ? String        ?
?????????????????????????????????????
```

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. This must be the first option on the command line if it is used.

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

? --open-files-limit=count

```
????????????????????????????????????????????????
?Command-Line Format ? --open-files-limit=count ?
```

??

?Type ? String ?

??

The number of files that mysqld should be able to open. The option value is passed to ulimit -n.

Note

You must start mysqld_safe as root for this to function properly.

? --pid-file=file_name

??

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

??

?Type ? File name ?

??

The path name that mysqld should use for its process ID file.

? --plugin-dir=dir_name

???

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

???

?Type ? Directory name ?

???

The path name of the plugin directory.

? --port=port_num

???????????????????????????????????????

?Command-Line Format ? --port=number ?

???????????????????????????????????????

?Type ? Numeric ?

???????????????????????????????????????

The port number that the server should use when listening for TCP/IP connections. The port number must be 1024 or higher unless the server is started by the root operating system user.

? --skip-kill-mysqld

??

?Command-Line Format ? --skip-kill-mysqld ?

??

Do not try to kill stray mysqld processes at startup. This option works only on Linux.

? --socket=path

??

?Command-Line Format ? --socket=file_name ?

??

?Type ? File name ?

??

The Unix socket file that the server should use when listening for local connections.

? --syslog, --skip-syslog

??????????????????????????????????

?Command-Line Format ? --syslog ?

??????????????????????????????????

?Deprecated ? Yes ?

??????????????????????????????????

???????????????????????????????????????

?Command-Line Format ? --skip-syslog ?

???????????????????????????????????????

?Deprecated ? Yes ?

???????????????????????????????????????

--syslog causes error messages to be sent to syslog on systems that support the logger program. --skip-syslog suppresses the use of syslog; messages are written to an error log file.

When syslog is used for error logging, the daemon.err facility/severity is used for all log messages.

Using these options to control mysqld logging is deprecated. To write error log output to the system log, use the instructions at Section 5.4.2.8, ?Error Logging to the System Log?. To control the facility, use the server log_syslog_facility system variable.

? --syslog-tag=tag

??

?Command-Line Format ? --syslog-tag=tag ?

??

?Deprecated ? Yes ?

???

For logging to syslog, messages from mysqld_safe and mysqld are written with identifiers of mysqld_safe and mysqld, respectively. To specify a suffix for the identifiers, use --syslog-tag=tag, which modifies the identifiers to be mysqld_safe-tag and mysqld-tag.

Using this option to control mysqld logging is deprecated. Use the server log_syslog_tag system variable instead. See Section 5.4.2.8, ?Error Logging to the System Log?.

? --timezone=timezone

??

?Command-Line Format ? --timezone=timezone ?

??

?Type ? String ?

??

Set the TZ time zone environment variable to the given option value. Consult your operating system documentation for legal time zone specification formats.

? --user={user_name|user_id}

???

?Command-Line Format ? --user={user_name|user_id} ?

???

?Type ? String ?

???

?Type ? Numeric ?

???

Run the mysqld server as the user having the name user_name or the numeric user ID user_id. (?User? in this context refers to a system login account, not a MySQL user listed in the grant tables.)

If you execute mysqld_safe with the --defaults-file or --defaults-extra-file option to name an option file, the option must be the first one given on the command line or the option file is not used. For example, this command does not use the named option file:

```
mysql> mysqld_safe --port=port_num --defaults-file=file_name
```

Instead, use the following command:

```
mysql> mysqld_safe --defaults-file=file_name --port=port_num
```

The mysqld_safe script is written so that it normally can start a server that was installed from either a source or a binary distribution of MySQL, even though these types of distributions typically install the server in slightly different locations. (See Section 2.1.5, ?Installation Layouts?.) mysqld_safe expects one of the following conditions to be true:

? The server and databases can be found relative to the working directory (the directory from which mysqld_safe is invoked). For binary distributions, mysqld_safe looks under its working directory for bin and data directories. For source distributions, it looks for libexec and var directories. This condition should be met if you execute mysqld_safe from your MySQL installation directory (for example, /usr/local/mysql for a binary distribution).

? If the server and databases cannot be found relative to the working directory, mysqld_safe attempts to locate them by absolute path names. Typical locations are /usr/local/libexec and /usr/local/var. The actual locations are determined from the values configured into the distribution at the time it was built. They should be correct if MySQL is installed in the location specified at configuration time.

Because mysqld_safe tries to find the server and databases relative to its own working directory, you can install a binary distribution of MySQL anywhere, as long as you run mysqld_safe from the MySQL installation directory:

```
cd mysql_installation_directory
bin/mysqld_safe &
```

If mysqld_safe fails, even when invoked from the MySQL installation directory, specify the --ledir and --datadir options to indicate the directories in which the server and databases are located on your system.

mysqld_safe tries to use the sleep and date system utilities to determine how many times per second it has attempted to start. If these utilities are present and the attempted starts per second is greater than 5, mysqld_safe waits 1 full second before starting again. This is intended to prevent excessive CPU usage in the event of repeated failures. (Bug #11761530, Bug #54035)

When you use mysqld_safe to start mysqld, mysqld_safe arranges for error (and notice) messages from itself and from mysqld to go to the same destination.

There are several mysqld_safe options for controlling the destination of these messages:

? --log-error=file_name: Write error messages to the named error file.

? --syslog: Write error messages to syslog on systems that support the logger program.

? --skip-syslog: Do not write error messages to syslog. Messages are written to the default error log file (host_name.err in the data directory), or to a named file if the --log-error option is given.

If none of these options is given, the default is --skip-syslog.

When mysqld_safe writes a message, notices go to the logging destination (syslog or the error log file) and stdout. Errors go to the logging destination and stderr.

Note

Controlling mysqld logging from mysqld_safe is deprecated. Use the server's native syslog support instead. For more information, see Section 5.4.2.8, ?Error Logging to the System Log?.

COPYRIGHT

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/).