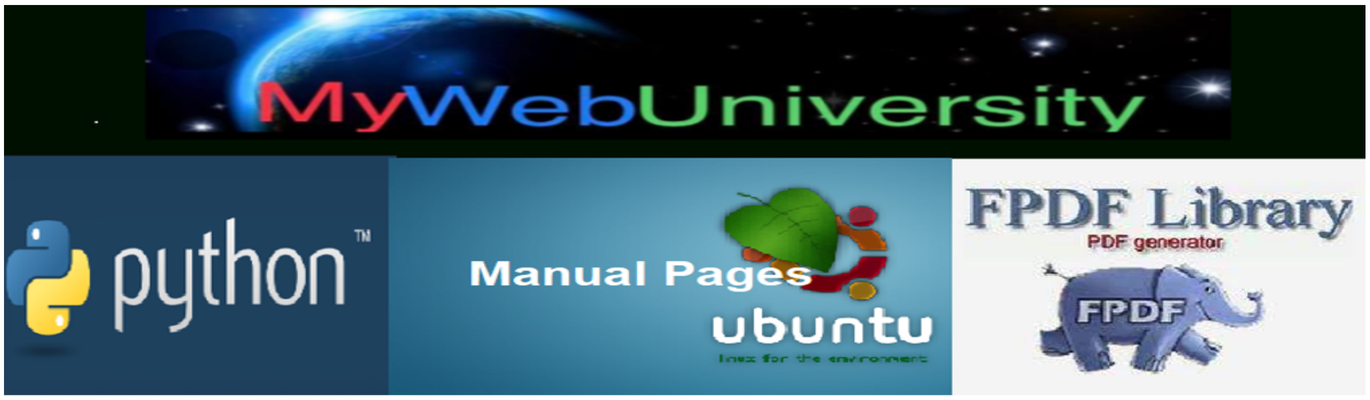

Rocky Enterprise Linux 9.2 Manual Pages on command 'mysql_config_editor.1'

$ man mysql_config_editor.1

MYSQL_CONFIG_EDITOR(1) MySQL Database System MYSQL_CONFIG_EDITOR(1)

NAME

mysql_config_editor - configure authentication information for connecting to MySQL server

SYNOPSIS

mysql_config_editor options command

DESCRIPTION

The mysql_config_editor utility enables you to store authentication credentials in an obfuscated login path file named .mylogin.cnf. The file location is the %APPDATA%\MySQL directory on Windows and the current user's home directory on non-Windows systems. The file can be read later by MySQL client programs to obtain authentication credentials for connecting to MySQL Server.

The unobfuscated format of the .mylogin.cnf login path file consists of option groups, similar to other option files. Each option group in .mylogin.cnf is called a ?login path,? which is a group that permits only certain options: host, user, password, port and socket. Think of a login path option group as a set of options that specify which MySQL server to connect to and which account to authenticate as. Here is an unobfuscated example:

```
[client]
user = mydefaultname
password = mydefaultpass
host = 127.0.0.1
[mypath]
user = myothername
password = myotherpass
```

```
    host = localhost
```

When you invoke a client program to connect to the server, the client uses .mylogin.cnf in conjunction with other option files. Its precedence is higher than other option files, but less than options specified explicitly on the client command line. For information about the order in which option files are used, see Section 4.2.2.2, ?Using Option Files?.

To specify an alternate login path file name, set the MYSQL_TEST_LOGIN_FILE environment variable. This variable is recognized by mysql_config_editor, by standard MySQL clients (mysql, mysqladmin, and so forth), and by the mysql-test-run.pl testing utility.

Programs use groups in the login path file as follows:

? mysql_config_editor operates on the client login path by default if you specify no --login-path=name option to indicate explicitly which login path to use.

? Without a --login-path option, client programs read the same option groups from the login path file that they read from other option files. Consider this command:

```
mysql
```

By default, the mysql client reads the [client] and [mysql] groups from other option files, so it reads them from the login path file as well.

? With a --login-path option, client programs additionally read the named login path from the login path file. The option groups read from other option files remain the same. Consider this command:

```
mysql --login-path=mypath
```

The mysql client reads [client] and [mysql] from other option files, and [client], [mysql], and [mypath] from the login path file.

? Client programs read the login path file even when the --no-defaults option is used, unless --no-login-paths is set. This permits passwords to be specified in a safer way than on the command line even if --no-defaults is present.

mysql_config_editor obfuscates the .mylogin.cnf file so it cannot be read as cleartext, and its contents when unobfuscated by client programs are used only in memory. In this way, passwords can be stored in a file in non-cleartext format and used later without ever needing to be exposed on the command line or in an environment variable.

mysql_config_editor provides a print command for displaying the login path file contents, but even in this case, password values are masked so as never to appear in a way that other users can see them.

The obfuscation used by mysql_config_editor prevents passwords from appearing in

.mylogin.cnf as cleartext and provides a measure of security by preventing inadvertent password exposure. For example, if you display a regular unobfuscated my.cnf option file on the screen, any passwords it contains are visible for anyone to see. With .mylogin.cnf, that is not true, but the obfuscation used is not likely to deter a determined attacker and you should not consider it unbreakable. A user who can gain system administration privileges on your machine to access your files could unobfuscate the .mylogin.cnf file with some effort.

The login path file must be readable and writable to the current user, and inaccessible to other users. Otherwise, mysql_config_editor ignores it, and client programs do not use it, either.

Invoke mysql_config_editor like this:

```
    mysql_config_editor [program_options] command [command_options]
```

If the login path file does not exist, mysql_config_editor creates it.

Command arguments are given as follows:

? program_options consists of general mysql_config_editor options.

? command indicates what action to perform on the .mylogin.cnf login path file. For example, set writes a login path to the file, remove removes a login path, and print displays login path contents.

? command_options indicates any additional options specific to the command, such as the login path name and the values to use in the login path.

The position of the command name within the set of program arguments is significant. For example, these command lines have the same arguments, but produce different results:

```
    mysql_config_editor --help set
    mysql_config_editor set --help
```

The first command line displays a general mysql_config_editor help message, and ignores the set command. The second command line displays a help message specific to the set command.

Suppose that you want to establish a client login path that defines your default connection parameters, and an additional login path named remote for connecting to the MySQL server the host remote.example.com. You want to log in as follows:

? By default, to the local server with a user name and password of localuser and localpass

? To the remote server with a user name and password of remoteuser and remotepass

To set up the login paths in the .mylogin.cnf file, use the following set commands. Enter each command on a single line, and enter the appropriate passwords when prompted:

```
   $> mysql_config_editor set --login-path=client
            --host=localhost --user=localuser --password
   Enter password: enter password "localpass" here
   $> mysql_config_editor set --login-path=remote
            --host=remote.example.com --user=remoteuser --password
   Enter password: enter password "remotepass" here
```

mysql_config_editor uses the client login path by default, so the --login-path=client option can be omitted from the first command without changing its effect.

To see what mysql_config_editor writes to the .mylogin.cnf file, use the print command:

```
   $> mysql_config_editor print --all
   [client]
   user = localuser
   password = *****
   host = localhost
   [remote]
   user = remoteuser
   password = *****
   host = remote.example.com
```

The print command displays each login path as a set of lines beginning with a group header indicating the login path name in square brackets, followed by the option values for the login path. Password values are masked and do not appear as cleartext.

If you do not specify --all to display all login paths or --login-path=name to display a named login path, the print command displays the client login path by default, if there is one.

As shown by the preceding example, the login path file can contain multiple login paths. In this way, mysql_config_editor makes it easy to set up multiple ?personalities? for connecting to different MySQL servers, or for connecting to a given server using different accounts. Any of these can be selected by name later using the --login-path option when you invoke a client program. For example, to connect to the remote server, use this command:

```
   mysql --login-path=remote
```

Here, mysql reads the [client] and [mysql] option groups from other option files, and the [client], [mysql], and [remote] groups from the login path file.

To connect to the local server, use this command:

```
mysql --login-path=client
```

Because mysql reads the client and mysql login paths by default, the --login-path option does not add anything in this case. That command is equivalent to this one:

```
mysql
```

Options read from the login path file take precedence over options read from other option files. Options read from login path groups appearing later in the login path file take precedence over options read from groups appearing earlier in the file.

mysql_config_editor adds login paths to the login path file in the order you create them, so you should create more general login paths first and more specific paths later. If you need to move a login path within the file, you can remove it, then recreate it to add it to the end. For example, a client login path is more general because it is read by all client programs, whereas a mysqldump login path is read only by mysqldump. Options specified later override options specified earlier, so putting the login paths in the order client, mysqldump enables mysqldump-specific options to override client options.

When you use the set command with mysql_config_editor to create a login path, you need not specify all possible option values (host name, user name, password, port, socket). Only those values given are written to the path. Any missing values required later can be specified when you invoke a client path to connect to the MySQL server, either in other option files or on the command line. Any options specified on the command line override those specified in the login path file or other option files. For example, if the credentials in the remote login path also apply for the host remote2.example.com, connect to the server on that host like this:

```
mysql --login-path=remote --host=remote2.example.com
```

mysql_config_editor General Options

mysql_config_editor supports the following general options, which may be used preceding any command named on the command line. For descriptions of command-specific options, see mysql_config_editor Commands and Command-Specific Options.

? --help, -?

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

?Command-Line Format ? --help ?

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

Display a general help message and exit.

To see a command-specific help message, invoke mysql_config_editor as follows, where command is a command other than help:

```
mysql_config_editor command --help
```

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

```
???????????????????????????????????????????????????
?Command-Line Format ? --debug[=debug_options] ?
???????????????????????????????????????????????????
?Type                ? String                 ?
???????????????????????????????????????????????????
?Default Value       ? d:t:o                  ?
???????????????????????????????????????????????????
```

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

? --verbose, -v

```
????????????????????????????????????
?Command-Line Format ? --verbose ?
????????????????????????????????????
```

Verbose mode. Print more information about what the program does. This option may be helpful in diagnosing problems if an operation does not have the effect you expect.

? --version, -V

```
????????????????????????????????????
?Command-Line Format ? --version ?
????????????????????????????????????
```

Display version information and exit.

mysql_config_editor Commands and Command-Specific Options

This section describes the permitted mysql_config_editor commands, and, for each one, the command-specific options permitted following the command name on the command line.

In addition, mysql_config_editor supports general options that can be used preceding any command. For descriptions of these options, see mysql_config_editor General Options.

mysql_config_editor supports these commands:

? help

Display a general help message and exit. This command takes no following options.

To see a command-specific help message, invoke mysql_config_editor as follows, where command is a command other than help:

```
mysql_config_editor command --help
```

? print [options]

Print the contents of the login path file in unobfuscated form, with the exception that passwords are displayed as *****.

The default login path name is client if no login path is named. If both --all and --login-path are given, --all takes precedence.

The print command permits these options following the command name:

? --help, -?

Display a help message for the print command and exit.

To see a general help message, use mysql_config_editor --help.

? --all

Print the contents of all login paths in the login path file.

? --login-path=name, -G name

Print the contents of the named login path.

? remove [options]

Remove a login path from the login path file, or modify a login path by removing options from it.

This command removes from the login path only such options as are specified with the --host, --password, --port, --socket, and --user options. If none of those options are given, remove removes the entire login path. For example, this command removes only the user option from the mypath login path rather than the entire mypath login path:

```
mysql_config_editor remove --login-path=mypath --user
```

This command removes the entire mypath login path:

```
mysql_config_editor remove --login-path=mypath
```

The remove command permits these options following the command name:

? --help, -?

Display a help message for the remove command and exit.

To see a general help message, use mysql_config_editor --help.

    ? --host, -h

      Remove the host name from the login path.

    ? --login-path=name, -G name

      The login path to remove or modify. The default login path name is client if this option is not given.

    ? --password, -p

      Remove the password from the login path.

    ? --port, -P

      Remove the TCP/IP port number from the login path.

    ? --socket, -S

      Remove the Unix socket file name from the login path.

    ? --user, -u

      Remove the user name from the login path.

    ? --warn, -w

      Warn and prompt the user for confirmation if the command attempts to remove the default login path (client) and --login-path=client was not specified. This option is enabled by default; use --skip-warn to disable it.

? reset [options]

  Empty the contents of the login path file.

  The reset command permits these options following the command name:

    ? --help, -?

      Display a help message for the reset command and exit.

      To see a general help message, use mysql_config_editor --help.

? set [options]

  Write a login path to the login path file.

  This command writes to the login path only such options as are specified with the --host, --password, --port, --socket, and --user options. If none of those options are given, mysql_config_editor writes the login path as an empty group.

  The set command permits these options following the command name:

    ? --help, -?

      Display a help message for the set command and exit.

      To see a general help message, use mysql_config_editor --help.

    ? --host=host_name, -h host_name

The host name to write to the login path.

? --login-path=name, -G name

The login path to create. The default login path name is client if this option is not given.

? --password, -p

Prompt for a password to write to the login path. After mysql_config_editor displays the prompt, type the password and press Enter. To prevent other users from seeing the password, mysql_config_editor does not echo it.

To specify an empty password, press Enter at the password prompt. The resulting login path written to the login path file includes a line like this:

```
password =
```

? --port=port_num, -P port_num

The TCP/IP port number to write to the login path.

? --socket=file_name, -S file_name

The Unix socket file name to write to the login path.

? --user=user_name, -u user_name

The user name to write to the login path.

? --warn, -w

Warn and prompt the user for confirmation if the command attempts to overwrite an existing login path. This option is enabled by default; use --skip-warn to disable it.

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/).