

MySQL Connector/Python

MySQL Connector/Python

Abstract

This manual describes how to install, configure, and develop database applications using MySQL Connector/Python, a self-contained Python driver for communicating with MySQL servers.

For release notes detailing the changes in each release of Connector/Python, see [MySQL Connector/Python Release Notes](#).

Document generated on: 2013-02-20 (revision: 34355)

Table of Contents

Preface and Legal Notices	vii
1. MySQL Connector/Python	1
2. Connector/Python Versions	3
3. Connector/Python Installation	5
3.1. Installing Connector/Python Source Distribution on Linux, UNIX, or OS X	5
3.2. Installing Connector/Python Source Distribution on Microsoft Windows	6
3.3. Verifying Your Connector/Python Installation	6
4. Connector/Python Coding Examples	7
4.1. Connecting to MySQL Using Connector/Python	7
4.2. Creating Tables Using Connector/Python	8
4.3. Inserting Data Using Connector/Python	10
4.4. Querying Data Using Connector/Python	11
5. Connector/Python Tutorials	13
5.1. Tutorial: Raise employee's salary using a buffering cursor	13
6. Connector/Python Connection Arguments	15
7. Connector/Python API Reference	19
7.1. Errors and Exceptions	21
7.1.1. Module <code>errorcode</code>	22
7.1.2. Exception <code>errors.Error</code>	22
7.1.3. Exception <code>errors.Warning</code>	23
7.1.4. Exception <code>errors.InterfaceError</code>	23
7.1.5. Exception <code>errors.DatabaseError</code>	23
7.1.6. Exception <code>errors.InternalError</code>	23
7.1.7. Exception <code>errors.OperationalError</code>	23
7.1.8. Exception <code>errors.ProgrammingError</code>	24
7.1.9. Exception <code>errors.IntegrityError</code>	24
7.1.10. Exception <code>errors.DataError</code>	24
7.1.11. Exception <code>errors.NotSupportedError</code>	24
7.1.12. Function <code>errors.custom_error_exception(error=None, exception=None)</code>	24
7.2. Class <code>connection.MySQLConnection</code>	25
7.2.1. Constructor <code>connection.MySQLConnection(**kwargs)</code>	25
7.2.2. Method <code>MySQLConnection.close()</code>	25
7.2.3. Method <code>MySQLConnection.config(**kwargs)</code>	25
7.2.4. Method <code>MySQLConnection.connect(**kwargs)</code>	25
7.2.5. Method <code>MySQLConnection.commit()</code>	25
7.2.6. Method <code>MySQLConnection.cursor(buffered=None, raw=None, cursor_class=None)</code>	26
7.2.7. Method <code>MySQLConnection.cmd_change_user(username='', password='', database='', charset=33)</code>	26
7.2.8. Method <code>MySQLConnection.cmd_debug()</code>	26
7.2.9. Method <code>MySQLConnection.cmd_init_db(database)</code>	26
7.2.10. Method <code>MySQLConnection.cmd_ping()</code>	26
7.2.11. Method <code>MySQLConnection.cmd_process_info()</code>	27
7.2.12. Method <code>MySQLConnection.cmd_process_kill(mysql_pid)</code>	27
7.2.13. Method <code>MySQLConnection.cmd_quit()</code>	27
7.2.14. Method <code>MySQLConnection.cmd_query(statement)</code>	27
7.2.15. Method <code>MySQLConnection.cmd_query_iter(statement)</code>	27
7.2.16. Method <code>MySQLConnection.cmd_refresh(options)</code>	28
7.2.17. Method <code>MySQLConnection.cmd_shutdown()</code>	28
7.2.18. Method <code>MySQLConnection.cmd_statistics()</code>	28

7.2.19. Method <code>MySQLConnection.disconnect()</code>	28
7.2.20. Method <code>MySQLConnection.get_rows(count=None)</code>	28
7.2.21. Method <code>MySQLConnection.get_row()</code>	28
7.2.22. Method <code>MySQLConnection.get_server_info()</code>	29
7.2.23. Method <code>MySQLConnection.get_server_version()</code>	29
7.2.24. Method <code>MySQLConnection.is_connected()</code>	29
7.2.25. Method <code>MySQLConnection.isset_client_flag(flag)</code>	29
7.2.26. Method <code>MySQLConnection.ping(attempts=1, delay=0)</code>	29
7.2.27. Method <code>MySQLConnection.reconnect(attempts=1, delay=0)</code>	29
7.2.28. Method <code>MySQLConnection.rollback()</code>	30
7.2.29. Method <code>MySQLConnection.set_charset_collation(charset=None, collation=None)</code>	30
7.2.30. Method <code>MySQLConnection.set_client_flags(flags)</code>	30
7.2.31. Property <code>MySQLConnection.autocommit</code>	31
7.2.32. Property <code>MySQLConnection.charset_name</code>	31
7.2.33. Property <code>MySQLConnection.collation_name</code>	31
7.2.34. Property <code>MySQLConnection.connection_id</code>	31
7.2.35. Property <code>MySQLConnection.database</code>	31
7.2.36. Property <code>MySQLConnection.get_warnings</code>	31
7.2.37. Property <code>MySQLConnection.raise_on_warnings</code>	32
7.2.38. Property <code>MySQLConnection.server_host</code>	32
7.2.39. Property <code>MySQLConnection.server_port</code>	32
7.2.40. Property <code>MySQLConnection.sql_mode</code>	32
7.2.41. Property <code>MySQLConnection.time_zone</code>	33
7.2.42. Property <code>MySQLConnection.unix_socket</code>	33
7.2.43. Property <code>MySQLConnection.user</code>	33
7.3. Class <code>cursor.MySQLCursor</code>	33
7.3.1. Constructor <code>cursor.MySQLCursor</code>	33
7.3.2. Method <code>MySQLCursor.callproc(procname, args=())</code>	33
7.3.3. Method <code>MySQLCursor.close()</code>	34
7.3.4. Method <code>MySQLCursor.execute(operation, params=None, multi=False)</code>	34
7.3.5. Method <code>MySQLCursor.executemany(operation, seq_params)</code>	35
7.3.6. Method <code>MySQLCursor.fetchall()</code>	35
7.3.7. Method <code>MySQLCursor.fetchmany(size=1)</code>	35
7.3.8. Method <code>MySQLCursor.fetchone()</code>	36
7.3.9. Method <code>MySQLCursor.fetchwarnings()</code>	36
7.3.10. Method <code>MySQLCursor.stored_results()</code>	36
7.3.11. Property <code>MySQLCursor.column_names</code>	37
7.3.12. Property <code>MySQLCursor.statement</code>	37
7.3.13. Property <code>MySQLCursor.with_rows</code>	37
7.4. Class <code>cursor.MySQLCursorBuffered</code>	37
7.5. Class <code>constants.ClientFlag</code>	38
7.6. Class <code>constants.FieldType</code>	38
7.7. Class <code>constants.SQLMode</code>	38
7.8. Class <code>constants.CharacterSet</code>	39
7.9. Class <code>constants.RefreshOption</code>	39
A. Licenses for Third-Party Components	41
A.1. <code>dtoa.c</code> License	42
A.2. Editline Library (<code>libedit</code>) License	42
A.3. <code>FindGTest.cmake</code> License	44
A.4. Fred Fish's Dbug Library License	45
A.5. <code>getarg</code> License	46
A.6. GNU General Public License Version 2.0, June 1991	47

A.7. GNU Lesser General Public License Version 2.1, February 1999	52
A.8. GNU Libtool License	60
A.9. GNU Readline License	61
A.10. Google Controlling Master Thread I/O Rate Patch License	61
A.11. Google Perftools (TCMalloc utility) License	62
A.12. Google SMP Patch License	62
A.13. lib_sql.cc License	63
A.14. Libaio License	63
A.15. libevent License	63
A.16. Linux-PAM License	65
A.17. md5 (Message-Digest Algorithm 5) License	66
A.18. nt_servc (Windows NT Service class library) License	66
A.19. OpenPAM License	67
A.20. Percona Multiple I/O Threads Patch License	67
A.21. Red HAT RPM Spec File License	68
A.22. RegEX-Spencer Library License	68
A.23. RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License	69
A.24. Richard A. O'Keefe String Library License	69
A.25. SHA-1 in C License	70
A.26. zlib License	70

Preface and Legal Notices

This manual describes how to install, configure, and develop database applications using MySQL Connector/Python, the a self-contained Python driver for communicating with MySQL servers.

Legal Notices

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. MySQL is a trademark of Oracle Corporation and/or its affiliates, and shall not be used without Oracle's express written authorization. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

This documentation is in prerelease status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement

only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle or as specifically provided below. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

For more information on the terms of this license, or for details on how the MySQL documentation is built and produced, please visit [MySQL Contact & Questions](#).

For additional licensing information, including licenses for third-party libraries used by MySQL products, see [Preface and Legal Notices](#).

For help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#) where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the [MySQL Documentation Library](#).

Chapter 1. MySQL Connector/Python

MySQL Connector/Python allows Python programs to access MySQL databases, using an API that is compliant with the [Python DB API version 2.0](#). It is written in pure Python and does not have any dependencies except for the [Python Standard Library](#).

For release notes detailing the changes in each release of Connector/Python, see [MySQL Connector/Python Release Notes](#).

MySQL Connector/Python includes support for:

- Almost all features provided by MySQL Server up to and including MySQL Server version 5.5.
- Converting parameter values back and forth between Python and MySQL data types, for example Python `datetime` and MySQL `DATETIME`. You can turn automatic conversion on for convenience, or off for optimal performance.
- All MySQL extensions to standard SQL syntax.
- Protocol compression, which enables compressing the data stream between the client and server.
- Connections using TCP/IP sockets and on Unix using Unix sockets.
- Secure TCP/IP connections using SSL.
- Self-contained driver. Connector/Python does not require the MySQL client library or any Python modules outside the standard library.

MySQL Connector/Python supports from Python version 2.4 through 2.7, and Python 3.1 and later. Note that Connector/Python does not support the old MySQL Server authentication methods, which means that MySQL versions prior to 4.1 will not work.

Chapter 2. Connector/Python Versions

MySQL Connector/Python v1.0.x series went going through a series of beta releases, leading to the first generally available (GA) version 1.0.7. Any development releases prior to general availability are not supported now that the GA version is released.

The following table summarizes the available Connector/Python versions:

Table 2.1. Connector/Python Version Reference

Connector/Python Version	MySQL Server Versions	Python Versions	Support Status for Connector
1.0	5.6, 5.5 (5.1, 5.0, 4.1)	2.7, 2.6 (2.5, 2.4); 3.1 and later	Recommended version

Note

MySQL server and Python versions within brackets are known to work with Connector/Python, but are not officially supported. Bugs might not get fixed for those versions.

Chapter 3. Connector/Python Installation

Table of Contents

3.1. Installing Connector/Python Source Distribution on Linux, UNIX, or OS X	5
3.2. Installing Connector/Python Source Distribution on Microsoft Windows	6
3.3. Verifying Your Connector/Python Installation	6

Connector/Python runs on any platform where Python is installed. Python comes pre-installed on almost any Linux distribution or UNIX-like system such as Apple Mac OS X and FreeBSD. On Microsoft Windows systems, you can install Python using the installer found on the [Python Download website](#).

Connector/Python is a pure Python implementation of the MySQL Client/Server protocol, meaning it does not require any other MySQL client libraries or other components. It also has no third-party dependencies. If you need SSL support, verify that your Python installation has been compiled using the [OpenSSL](#) libraries.

The installation of Connector/Python is similar on every platform and follows the standard [Python Distribution Utilities](#) or [Distutils](#). Some platforms have specific packaging, for example RPM, and, when made available, the installation of these will be covered in this manual.

Python terminology regarding distributions:

- **Source Distribution** is a distribution that contains only source files and is generally platform independent.
- **Built Distribution** can be regarded as a binary package. It contains both sources and platform-independent bytecode.

3.1. Installing Connector/Python Source Distribution on Linux, UNIX, or OS X

On UNIX-like systems such as Linux distributions, Solaris, Apple Mac OS X, and FreeBSD, you can download Connector/Python as a `tar` archive from <http://dev.mysql.com/downloads/connector/python/>.

To install Connector/Python from the `.tar.gz` file, download the latest version and follow these steps:

```
shell> gunzip mysql-connector-python-1.0.6b1.tar.gz
shell> tar xf mysql-connector-python-1.0.6b1.tar
shell> cd mysql-connector-python-1.0.6b1
shell> sudo python setup.py install
```

On UNIX-like systems, Connector/Python gets installed in the default location `/prefix/lib/pythonX.Y/site-packages/`, where `prefix` is the location where Python was installed and `X.Y` is the version of Python. See [How installation works](#) in the Python manual.

If you are not sure where Connector/Python was installed, do the following to retrieve the location:

```
shell> python
>>> from distutils.sysconfig import get_python_lib
>>> print get_python_lib()           # Python v2.x
/Library/Python/2.7/site-packages
>>> print(get_python_lib())         # Python v3.x
/Library/Frameworks/Python.framework/Versions/3.1/lib/python3.1/site-packages
```

Note

The above example shows the default installation location on Mac OS X 10.7.

3.2. Installing Connector/Python Source Distribution on Microsoft Windows

On Microsoft Windows systems, you can download Connector/Python as a [zip](http://dev.mysql.com/downloads/connector/python/) archive from <http://dev.mysql.com/downloads/connector/python/>.

Make sure that the Python executable is available in the Windows `%PATH%` setting. For more information about installation and configuration of Python on Windows, see the section [Using Python on Windows](#) in the Python documentation.

To install Connector/Python from the `.zip` file, download the latest version and follow these steps:

1. Unpack the downloaded `zip` archive into a directory of your choice. For example, into the folder `C:\mysql-connector\`. Use the appropriate unzip command for your system, for example, `unzip`, `pkunzip`, and so on.
2. Start a console window (or a DOS window) and change to the folder where you unpacked the Connector/Python `zip` archive.

```
shell> cd C:\mysql-connector\
```

3. Once inside the Connector/Python folder, do the following:

```
shell> python setup.py install
```

On Windows, Connector/Python gets installed in the default location `C:\PythonX.Y\Lib\site-packages\` where `X.Y` is the Python version you used to install the connector.

If you are not sure where Connector/Python ended up, do the following to retrieve the location where packages get installed:

```
shell> python
>>> from distutils.sysconfig import get_python_lib
>>> print get_python_lib()           # Python v2.x
>>> print(get_python_lib())         # Python v3.x
```

3.3. Verifying Your Connector/Python Installation

To test that your Connector/Python installation is working and is able to connect to a MySQL database server, you can run a very simple program where you substitute the login credentials and host information of the MySQL server. See [Section 4.1, "Connecting to MySQL Using Connector/Python"](#) for an example.

Chapter 4. Connector/Python Coding Examples

Table of Contents

4.1. Connecting to MySQL Using Connector/Python	7
4.2. Creating Tables Using Connector/Python	8
4.3. Inserting Data Using Connector/Python	10
4.4. Querying Data Using Connector/Python	11

These coding examples illustrate how to develop Python applications and scripts which connect to a MySQL Server using MySQL Connector/Python.

4.1. Connecting to MySQL Using Connector/Python

The `connect()` constructor is used for creating a connection to the MySQL server and returns a `MySQLConnection` object.

The following example shows how to connect to the MySQL server:

```
import mysql.connector
cnx = mysql.connector.connect(user='scott', password='tiger',
                             host='127.0.0.1',
                             database='employees')
cnx.close()
```

See [Chapter 6, Connector/Python Connection Arguments](#) for all possible connection arguments.

It is also possible to create connection objects using the `connection.MySQLConnection()` class. Both methods, using the `connect()` constructor, or the class directly, are valid and functionally equal, but using `connector()` is preferred and will be used in most examples in this manual.

To handle connection errors, use the `try` statement and catch all errors using the `errors.Error` exception:

```
import mysql.connector
from mysql.connector import errorcode
try:
    cnx = mysql.connector.connect(user='scott',
                                 database='testt')
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong your username or password")
    elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exists")
    else:
        print(err)
else:
    cnx.close()
```

If you have lots of connection arguments, it's best to keep them in a dictionary and use the `**`-operator. Here is an example:

```
import mysql.connector
config = {
    'user': 'scott',
    'password': 'tiger',
    'host': '127.0.0.1',
    'database': 'employees',
    'raise_on_warnings': True,
```

```

}
cnx = mysql.connector.connect(**config)
cnx.close()

```

4.2. Creating Tables Using Connector/Python

All [DDL](#) (Data Definition Language) statements are executed using a handle structure known as a cursor. The following examples show how to create the tables of the `employees` database. You will need them for the other examples.

In a MySQL server, tables are very long-lived objects, and are often accessed by multiple applications written in different languages. You might typically work with tables that are already set up, rather than creating them within your own application. Avoid setting up and dropping tables over and over again, as that is an expensive operation. The exception is [temporary tables](#), which can be created and dropped quickly within an application.

```

from __future__ import print_function
import mysql.connector
from mysql.connector import errorcode
DB_NAME = 'employees'
TABLES = {}
TABLES['employees'] = (
    "CREATE TABLE `employees` ("
    "  `emp_no` int(11) NOT NULL AUTO_INCREMENT,"
    "  `birth_date` date NOT NULL,"
    "  `first_name` varchar(14) NOT NULL,"
    "  `last_name` varchar(16) NOT NULL,"
    "  `gender` enum('M','F') NOT NULL,"
    "  `hire_date` date NOT NULL,"
    "  PRIMARY KEY (`emp_no`)"
    ") ENGINE=InnoDB")
TABLES['departments'] = (
    "CREATE TABLE `departments` ("
    "  `dept_no` char(4) NOT NULL,"
    "  `dept_name` varchar(40) NOT NULL,"
    "  PRIMARY KEY (`dept_no`), UNIQUE KEY `dept_name` (`dept_name`)"
    ") ENGINE=InnoDB")
TABLES['salaries'] = (
    "CREATE TABLE `salaries` ("
    "  `emp_no` int(11) NOT NULL,"
    "  `salary` int(11) NOT NULL,"
    "  `from_date` date NOT NULL,"
    "  `to_date` date NOT NULL,"
    "  PRIMARY KEY (`emp_no`,`from_date`), KEY `emp_no` (`emp_no`),"
    "  CONSTRAINT `salaries_ibfk_1` FOREIGN KEY (`emp_no`) "
    "    REFERENCES `employees` (`emp_no`) ON DELETE CASCADE"
    ") ENGINE=InnoDB")
TABLES['dept_emp'] = (
    "CREATE TABLE `dept_emp` ("
    "  `emp_no` int(11) NOT NULL,"
    "  `dept_no` char(4) NOT NULL,"
    "  `from_date` date NOT NULL,"
    "  `to_date` date NOT NULL,"
    "  PRIMARY KEY (`emp_no`,`dept_no`), KEY `emp_no` (`emp_no`),"
    "  KEY `dept_no` (`dept_no`),"
    "  CONSTRAINT `dept_emp_ibfk_1` FOREIGN KEY (`emp_no`) "
    "    REFERENCES `employees` (`emp_no`) ON DELETE CASCADE,"
    "  CONSTRAINT `dept_emp_ibfk_2` FOREIGN KEY (`dept_no`) "
    "    REFERENCES `departments` (`dept_no`) ON DELETE CASCADE"
    ") ENGINE=InnoDB")
TABLES['dept_manager'] = (
    "CREATE TABLE `dept_manager` ("
    "  `dept_no` char(4) NOT NULL,"
    "  `emp_no` int(11) NOT NULL,"

```

```

" `from_date` date NOT NULL,"
" `to_date` date NOT NULL,"
" PRIMARY KEY (`emp_no`,`dept_no`),"
" KEY `emp_no` (`emp_no`),"
" KEY `dept_no` (`dept_no`),"
" CONSTRAINT `dept_manager_ibfk_1` FOREIGN KEY (`emp_no`) "
"   REFERENCES `employees` (`emp_no`) ON DELETE CASCADE,"
" CONSTRAINT `dept_manager_ibfk_2` FOREIGN KEY (`dept_no`) "
"   REFERENCES `departments` (`dept_no`) ON DELETE CASCADE"
") ENGINE=InnoDB"
TABLES['titles'] = (
"CREATE TABLE `titles` ("
"  `emp_no` int(11) NOT NULL,"
"  `title` varchar(50) NOT NULL,"
"  `from_date` date NOT NULL,"
"  `to_date` date DEFAULT NULL,"
"  PRIMARY KEY (`emp_no`,`title`,`from_date`), KEY `emp_no` (`emp_no`),"
"  CONSTRAINT `titles_ibfk_1` FOREIGN KEY (`emp_no`)"
"    REFERENCES `employees` (`emp_no`) ON DELETE CASCADE"
") ENGINE=InnoDB"

```

The above code shows how we are storing the `CREATE` statements in a Python dictionary called `TABLES`. We also define the database in a global variable called `DB_NAME`, which allows you to easily use a different schema.

```

cnx = mysql.connector.connect(user='scott')
cursor = cnx.cursor()

```

A single MySQL server can contain multiple `databases`. Typically, you specify the database to switch to when connecting to the MySQL server. This example does not connect to the database upon connection, so that it can make sure the database exists, and create it if not.

```

def create_database(cursor):
    try:
        cursor.execute(
            "CREATE DATABASE {} DEFAULT CHARACTER SET 'utf8'".format(DB_NAME))
    except mysql.connector.Error as err:
        print("Failed creating database: {}".format(err))
        exit(1)
try:
    cnx.database = DB_NAME
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_BAD_DB_ERROR:
        create_database(cursor)
        cnx.database = DB_NAME
    else:
        print(err)
        exit(1)

```

We first try to change to a particular database using the `database` property of the connection object `cnx`. If there is an error, we examine the error number to check if the database does not exist. If so, we call the `create_database` function to create it for us.

On any other error, the application exits and displays the error message.

```

for name, ddl in TABLES.iteritems():
    try:
        print("Creating table {}: ".format(name), end='')
        cursor.execute(ddl)
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_TABLE_EXISTS_ERROR:
            print("already exists.")
        else:
            print(err.errmsg)
    else:

```

```
        print("OK")
    cursor.close()
    cnx.close()
```

After we successfully created or changed to the target database, we create the tables by iterating over the items of the `TABLES` dictionary.

We handle the error when the table already exists by simply notifying the user that it was already there. Other errors are printed, but we simply continue creating tables. (We show how to handle the “table already exists” condition for illustration purposes. In a real application, we would typically avoid the error condition entirely by using the `IF NOT EXISTS` clause of the `CREATE TABLE` statement.)

The output would be something like this:

```
Creating table employees: already exists.
Creating table salaries: already exists.
Creating table titles: OK
Creating table departments: already exists.
Creating table dept_manager: already exists.
Creating table dept_emp: already exists.
```

To populate the employees tables, use the dump files of the [Employee Sample Database](#). Note that you only need the data dump files that you will find in an archive named like `employees_db-dump-files-1.0.5.tar.bz2`. After downloading the dump files, do the following from the command line, adding connection options to the `mysql` commands if necessary:

```
shell> tar xzf employees_db-dump-files-1.0.5.tar.bz2
shell> cd employees_db
shell> mysql employees < load_employees.dump
shell> mysql employees < load_titles.dump
shell> mysql employees < load_departments.dump
shell> mysql employees < load_salaries.dump
shell> mysql employees < load_dept_emp.dump
shell> mysql employees < load_dept_manager.dump
```

4.3. Inserting Data Using Connector/Python

Inserting or updating data is also done using the handler structure known as a cursor. When you use a transactional storage engine such as [InnoDB](#) (which is the default in MySQL 5.5 and later), you must [commit](#) the data after a sequence of `INSERT`, `DELETE`, and `UPDATE` statements.

In this example we show how to insert new data. The second `INSERT` depends on the value of the newly created [primary key](#) of the first. We are also demonstrating how to use extended formats. The task is to add a new employee starting to work tomorrow with a salary set to 50000.

Note

The following example uses tables created in the example [Section 4.2, “Creating Tables Using Connector/Python”](#). The `AUTO_INCREMENT` column option for the primary key of the `employees` table is important to ensure reliable, easily searchable data.

```
from __future__ import print_function
from datetime import date, datetime, timedelta
import mysql.connector
cnx = mysql.connector.connect(user='scott', database='employees')
cursor = cnx.cursor()
tomorrow = datetime.now().date() + timedelta(days=1)
add_employee = ("INSERT INTO employees "
               "(first_name, last_name, hire_date, gender, birth_date) "
```

```

        "VALUES (%s, %s, %s, %s, %s)")
add_salary = ("INSERT INTO salaries "
             "(emp_no, salary, from_date, to_date) "
             "VALUES (%(emp_no)s, %(salary)s, %(from_date)s, %(to_date)s)")
data_employee = ('Geert', 'Vanderkelen', tomorrow, 'M', date(1977, 6, 14))
# Insert new employee
cursor.execute(add_employee, data_employee)
emp_no = cursor.lastrowid
# Insert salary information
data_salary = {
    'emp_no': emp_no,
    'salary': 50000,
    'from_date': tomorrow,
    'to_date': date(9999, 1, 1),
}
cursor.execute(add_salary, data_salary)
# Make sure data is committed to the database
cnx.commit()
cursor.close()
cnx.close()

```

We first open a connection to the MySQL server and store the [connection object](#) in the variable `cnx`. We then create a new cursor, by default a [MySQLCursor](#) object, using the connection's `cursor()` method.

We could calculate tomorrow by calling a database function, but for clarity we do it in Python using the [datetime](#) module.

Both `INSERT` statements are stored in the variables called `add_employee` and `add_salary`. Note that the second `INSERT` statement uses extended Python format codes.

The information of the new employee is stored in the tuple `data_employee`. The query to insert the new employee is executed and we retrieve the newly inserted value for the column `emp_no` using the `lastrowid` property of the cursor object.

Next, we insert the new salary for the new employee. We are using the `emp_no` variable in the dictionary holding the data. This dictionary is passed to the `execute()` method of the cursor object.

Since by default Connector/Python turns [autocommit](#) off, and MySQL 5.5 and later uses transactional [InnoDB](#) tables by default, it is necessary to commit your changes using the connection's `commit()` method. You could also [roll back](#) using the `rollback()` method.

4.4. Querying Data Using Connector/Python

The following example shows how to [query](#) data using a cursor created using the connection's `cursor()` method. The data returned is formatted and printed on the console.

The task is to select all employees hired in the year 1999 and print their names with their hire date to the console.

```

import datetime
import mysql.connector
cnx = mysql.connector.connect(user='scott', database='employees')
cursor = cnx.cursor()
query = ("SELECT first_name, last_name, hire_date FROM employees "
        "WHERE hire_date BETWEEN %s AND %s")
hire_start = datetime.date(1999, 1, 1)
hire_end = datetime.date(1999, 12, 31)
cursor.execute(query, (hire_start, hire_end))
for (first_name, last_name, hire_date) in cursor:
    print("{} , {} was hired on {:d %b %Y}".format(
        last_name, first_name, hire_date))
cursor.close()

```

```
cnx.close()
```

We first open a connection to the MySQL server and store the [connection object](#) in the variable `cnx`. We then create a new cursor, by default a [MySQLCursor](#) object, using the connection's `cursor()` method.

In the preceding example, we store the `SELECT` statement in the variable `query`. Note that we are using unquoted `%s`-markers where dates should have been. Connector/Python converts `hire_start` and `hire_end` from Python types to a data type that MySQL understands and adds the required quotes. In this case, it replaces the first `%s` with `'1999-01-01'`, and the second with `'1999-12-31'`.

We then execute the operation stored in the `query` variable using the `execute()` method. The data used to replace the `%s`-markers in the query is passed as a tuple: `(hire_start, hire_end)`.

After executing the query, the MySQL server is ready to send the data. The result set could be zero rows, one row, or 100 million rows. Depending on the expected volume, you can use different techniques to process this result set. In this example, we use the `cursor` object as an iterator. The first column in the row will be stored in the variable `first_name`, the second in `last_name`, and the third in `hire_date`.

We print the result, formatting the output using Python's built-in `format()` function. Note that `hire_date` was converted automatically by Connector/Python to a Python `datetime.date` object. This means that we can easily format the date in a more human-readable form.

The output should be something like this:

```
..
Wilharm, LiMin was hired on 16 Dec 1999
Wielonsky, Lalit was hired on 16 Dec 1999
Kamble, Dannz was hired on 18 Dec 1999
DuBourdieu, Zhongwei was hired on 19 Dec 1999
Fujisawa, Rosita was hired on 20 Dec 1999
..
```

Chapter 5. Connector/Python Tutorials

Table of Contents

5.1. Tutorial: Raise employee's salary using a buffering cursor	13
---	----

These tutorials illustrate how to develop Python applications and scripts that connect to a MySQL database server using MySQL Connector/Python.

5.1. Tutorial: Raise employee's salary using a buffering cursor

The following example script will give a long-overdue raise effective tomorrow to all employees who joined in the year 2000 and are still with the company.

We are using buffered cursors to iterate through the selected employees. This way we do not have to fetch the rows in a new variables, but can instead use the cursor as an iterator.

Note that the script is an example; there are other ways of doing this simple task.

```
from __future__ import print_function
from decimal import Decimal
from datetime import datetime, date, timedelta
import mysql.connector
# Connect with the MySQL Server
cnx = mysql.connector.connect(user='scott', database='employees')
# Get two buffered cursors
curA = cnx.cursor(buffered=True)
curB = cnx.cursor(buffered=True)
# Query to get employees who joined in a period defined by two dates
query = (
    "SELECT s.emp_no, salary, from_date, to_date FROM employees AS e "
    "LEFT JOIN salaries AS s USING (emp_no) "
    "WHERE to_date = DATE('9999-01-01') "
    "AND e.hire_date BETWEEN DATE(%s) AND DATE(%s)"
)
# UPDATE and INSERT statements for the old and new salary
update_old_salary = (
    "UPDATE salaries SET to_date = %s "
    "WHERE emp_no = %s AND from_date = %s"
)
insert_new_salary = (
    "INSERT INTO salaries (emp_no, from_date, to_date, salary) "
    "VALUES (%s, %s, %s, %s)"
)
# Select the employees getting a raise
curA.execute(query, (date(2000, 1, 1), date(2001, 1, 1)))
# Iterate through the result of curA
for (emp_no, salary, from_date, to_date) in curA:
    # Update the old and insert the new salary
    new_salary = int(round(salary * Decimal('1.15')))
    curB.execute(update_old_salary, (tomorrow, emp_no, from_date))
    curB.execute(insert_new_salary,
                 (emp_no, tomorrow, date(9999, 1, 1), new_salary))
# Commit the changes
cnx.commit()
cnx.close()
```

Chapter 6. Connector/Python Connection Arguments

The following lists the arguments that can be used to initiate a connection with the MySQL server using either:

- Function `mysql.connector.connect()`
- Class `mysql.connector.MySQLConnection()`

Table 6.1. Connection Arguments for Connector/Python

Argument Name	Default	Description
<code>user</code> (<code>username*</code>)		The username used to authenticate with the MySQL Server.
<code>password</code> (<code>passwd*</code>)		The password to authenticate the user with the MySQL Server.
<code>database</code> (<code>db*</code>)		Database name to use when connecting with the MySQL Server.
<code>host</code>	127.0.0.1	Hostname or IP address of the MySQL Server.
<code>port</code>	3306	TCP/IP port of the MySQL Server. Must be an integer.
<code>unix_socket</code>		The location of the Unix socket file.
<code>use_unicode</code>	True	Whether to use Unicode or not.
<code>charset</code>	utf8	Which MySQL character set to use.
<code>collation</code>	utf8_general_ci	Which MySQL collation to use.
<code>autocommit</code>	False	Whether to <code>autocommit</code> transactions.
<code>time_zone</code>		Set the <code>time_zone</code> session variable at connection.
<code>sql_mode</code>		Set the <code>sql_mode</code> session variable at connection.
<code>get_warnings</code>	False	Whether to fetch warnings.
<code>raise_on_warnings</code>	False	Whether to raise an exception on warnings.
<code>connection_timeout</code> (<code>connect_timeout*</code>)		Timeout for the TCP and Unix socket connections.
<code>client_flags</code>		MySQL client flags.
<code>buffered</code>	False	Whether cursor object fetches the result immediately after executing query.
<code>raw</code>	False	Whether MySQL results are returned as-is, rather than converted to Python types.
<code>ssl_ca</code>		File containing the SSL certificate authority.
<code>ssl_cert</code>		File containing the SSL certificate file.
<code>ssl_key</code>		File containing the SSL key.
<code>force_ipv6</code>	False	When set to <code>True</code> , uses IPv6 when an address resolves to both IPv4 and IPv6. By default, IPv4 is used in such cases.
<code>dsn</code>		Not supported (raises <code>NotSupportedError</code> when used).

* Synonymous argument name, available only for compatibility with other Python MySQL drivers. Oracle recommends not to use these alternative names.

MySQL Authentication

Authentication with MySQL will use `username` and `password`. Note that MySQL Connector/Python does not support the old, insecure password protocols of MySQL versions prior to 4.1.

When the `database` parameter is given, the current database is set to the given value. To later change the database, execute the MySQL `USE` command or set the `database` property of the `MySQLConnection` instance.

By default, Connector/Python tries to connect to a MySQL server running on `localhost` using TCP/IP. The `host` argument defaults to IP address 127.0.0.1 and `port` to 3306. Unix sockets are supported by setting `unix_socket`. Named pipes on the Windows platform are not supported.

Character Encoding

Strings coming from MySQL are by default returned as Python Unicode literals. To change this behavior, set `use_unicode` to `False`. You can change the character setting for the client connection through the `charset` argument. To change the character set after connecting to MySQL, set the `charset` property of the `MySQLConnection` instance. This technique is preferred over using the MySQL `SET NAMES` statement directly. Similar to the `charset` property, you can set the `collation` for the current MySQL session.

Transactions

Transactions are not automatically committed; call the `commit()` method of the `MySQLConnection` instance within your application after doing a set of related insert, update, and delete operations. For data consistency and high throughput for write operations, it is best to leave the `autocommit` configuration option turned off when using `InnoDB` or other transactional tables.

Time Zones

The time zone can be set per connection using the `time_zone` argument. This is useful if the MySQL server is set, for example, to UTC and `TIMESTAMP` values should be returned by MySQL converted to the `PST` time zone.

SQL Modes

MySQL supports so called SQL Modes, which will change the behavior of the server globally or per connection. For example, to have warnings raised as errors, set `sql_mode` to `TRADITIONAL`. For more information, see [Server SQL Modes](#).

Troubleshooting and Error Handling

Warnings generated by queries are fetched automatically when `get_warnings` is set to `True`. You can also immediately raise an exception by setting `raise_on_warnings` to `True`. Consider using the MySQL `sql_mode` setting for turning warnings into errors.

To set a timeout value for connections, use `connection_timeout`.

Enabling and Disabling Features through Client Flags

MySQL uses `client flags` to enable or disable features. Using the `client_flags` argument, you have control of what is set. To find out what flags are available, use the following:

```
from mysql.connector.constants import ClientFlag
print '\n'.join(ClientFlag.get_full_info())
```

If `client_flags` is not specified (that is, it is zero), defaults are used for MySQL v4.1 and later. If you specify an integer greater than 0, make sure all flags are set. A better way to set and unset flags is to use a list. For example, to set `FOUND_ROWS`, but disable the default `LONG_FLAG`:

```
flags = [ClientFlag.FOUND_ROWS, -ClientFlag.LONG_FLAG]
mysql.connector.connect(client_flags=flags)
```

Buffering for Result Sets

By default, MySQL Connector/Python does not buffer or pre-fetch results. This means that after a query is executed, your program is responsible of fetching the data. This avoids using excessive memory when queries return large result sets. If you know that the result set is small enough to handle all at once, fetching the results immediately by setting `buffered` to `True`. It is also possible to set this per cursor (see cursor manual).

Type Conversions

MySQL types will be converted automatically to Python types. For example, a `DATETIME` column becomes a `datetime.datetime` object. When conversion should be done differently, for example to get better performance, set `raw` to `True`.

Connecting through SSL

Using SSL connections is possible when your [Python installation supports SSL](#), that is, when it is compiled against the OpenSSL libraries. When you provide the arguments `ssl_ca`, `ssl_key` and `ssl_cert`, the connection switches to SSL, and the `client_flags` option includes the `ClientFlag.SSL` value automatically. You can use this in combination with the `compressed` argument set to `True`.

```
# Note (Example is valid for Python v2 and v3)
from __future__ import print_function
import sys
#sys.path.insert(0, 'python{0}/'.format(sys.version_info[0]))
import mysql.connector
from mysql.connector.constants import ClientFlag
config = {
    'user': 'ssluser',
    'password': 'asecret',
    'host': '127.0.0.1',
    'client_flags': [ClientFlag.SSL],
    'ssl_ca': '/opt/mysql/ssl/ca-cert.pem',
    'ssl_cert': '/opt/mysql/ssl/client-cert.pem',
    'ssl_key': '/opt/mysql/ssl/client-key.pem',
}
cnx = mysql.connector.connect(**config)
cur = cnx.cursor(buffered=True)
cur.execute("SHOW STATUS LIKE 'Ssl_cipher'")
print(cur.fetchone())
cur.close()
cnx.close()
```

Compatibility with Other Connection Interfaces

`passwd`, `db` and `connect_timeout` are valid for compatibility with other MySQL interfaces and are respectively the same as `password`, `database` and `connection_timeout`. The latter take precedence. Data source name syntax or `dsn` is not used; if specified, it raises a `NotSupportedError` exception.

Chapter 7. Connector/Python API Reference

Table of Contents

7.1. Errors and Exceptions	21
7.1.1. Module <code>errorcode</code>	22
7.1.2. Exception <code>errors.Error</code>	22
7.1.3. Exception <code>errors.Warning</code>	23
7.1.4. Exception <code>errors.InterfaceError</code>	23
7.1.5. Exception <code>errors.DatabaseError</code>	23
7.1.6. Exception <code>errors.InternalError</code>	23
7.1.7. Exception <code>errors.OperationalError</code>	23
7.1.8. Exception <code>errors.ProgrammingError</code>	24
7.1.9. Exception <code>errors.IntegrityError</code>	24
7.1.10. Exception <code>errors.DataError</code>	24
7.1.11. Exception <code>errors.NotSupportedError</code>	24
7.1.12. Function <code>errors.custom_error_exception(error=None, exception=None)</code>	24
7.2. Class <code>connection.MySQLConnection</code>	25
7.2.1. Constructor <code>connection.MySQLConnection(**kwargs)</code>	25
7.2.2. Method <code>MySQLConnection.close()</code>	25
7.2.3. Method <code>MySQLConnection.config(**kwargs)</code>	25
7.2.4. Method <code>MySQLConnection.connect(**kwargs)</code>	25
7.2.5. Method <code>MySQLConnection.commit()</code>	25
7.2.6. Method <code>MySQLConnection.cursor(buffered=None, raw=None, cursor_class=None)</code>	26
7.2.7. Method <code>MySQLConnection.cmd_change_user(username='', password='', database='', charset=33)</code>	26
7.2.8. Method <code>MySQLConnection.cmd_debug()</code>	26
7.2.9. Method <code>MySQLConnection.cmd_init_db(database)</code>	26
7.2.10. Method <code>MySQLConnection.cmd_ping()</code>	26
7.2.11. Method <code>MySQLConnection.cmd_process_info()</code>	27
7.2.12. Method <code>MySQLConnection.cmd_process_kill(mysql_pid)</code>	27
7.2.13. Method <code>MySQLConnection.cmd_quit()</code>	27
7.2.14. Method <code>MySQLConnection.cmd_query(statement)</code>	27
7.2.15. Method <code>MySQLConnection.cmd_query_iter(statement)</code>	27
7.2.16. Method <code>MySQLConnection.cmd_refresh(options)</code>	28
7.2.17. Method <code>MySQLConnection.cmd_shutdown()</code>	28
7.2.18. Method <code>MySQLConnection.cmd_statistics()</code>	28
7.2.19. Method <code>MySQLConnection.disconnect()</code>	28
7.2.20. Method <code>MySQLConnection.get_rows(count=None)</code>	28
7.2.21. Method <code>MySQLConnection.get_row()</code>	28
7.2.22. Method <code>MySQLConnection.get_server_info()</code>	29
7.2.23. Method <code>MySQLConnection.get_server_version()</code>	29
7.2.24. Method <code>MySQLConnection.is_connected()</code>	29
7.2.25. Method <code>MySQLConnection.isset_client_flag(flag)</code>	29
7.2.26. Method <code>MySQLConnection.ping(attempts=1, delay=0)</code>	29
7.2.27. Method <code>MySQLConnection.reconnect(attempts=1, delay=0)</code>	29
7.2.28. Method <code>MySQLConnection.rollback()</code>	30
7.2.29. Method <code>MySQLConnection.set_charset_collation(charset=None, collation=None)</code>	30
7.2.30. Method <code>MySQLConnection.set_client_flags(flags)</code>	30

7.2.31. Property <code>MySQLConnection.autocommit</code>	31
7.2.32. Property <code>MySQLConnection.charset_name</code>	31
7.2.33. Property <code>MySQLConnection.collation_name</code>	31
7.2.34. Property <code>MySQLConnection.connection_id</code>	31
7.2.35. Property <code>MySQLConnection.database</code>	31
7.2.36. Property <code>MySQLConnection.get_warnings</code>	31
7.2.37. Property <code>MySQLConnection.raise_on_warnings</code>	32
7.2.38. Property <code>MySQLConnection.server_host</code>	32
7.2.39. Property <code>MySQLConnection.server_port</code>	32
7.2.40. Property <code>MySQLConnection.sql_mode</code>	32
7.2.41. Property <code>MySQLConnection.time_zone</code>	33
7.2.42. Property <code>MySQLConnection.unix_socket</code>	33
7.2.43. Property <code>MySQLConnection.user</code>	33
7.3. Class <code>cursor.MySQLCursor</code>	33
7.3.1. Constructor <code>cursor.MySQLCursor</code>	33
7.3.2. Method <code>MySQLCursor.callproc(procname, args=())</code>	33
7.3.3. Method <code>MySQLCursor.close()</code>	34
7.3.4. Method <code>MySQLCursor.execute(operation, params=None, multi=False)</code>	34
7.3.5. Method <code>MySQLCursor.executemany(operation, seq_params)</code>	35
7.3.6. Method <code>MySQLCursor.fetchall()</code>	35
7.3.7. Method <code>MySQLCursor.fetchmany(size=1)</code>	35
7.3.8. Method <code>MySQLCursor.fetchone()</code>	36
7.3.9. Method <code>MySQLCursor.fetchwarnings()</code>	36
7.3.10. Method <code>MySQLCursor.stored_results()</code>	36
7.3.11. Property <code>MySQLCursor.column_names</code>	37
7.3.12. Property <code>MySQLCursor.statement</code>	37
7.3.13. Property <code>MySQLCursor.with_rows</code>	37
7.4. Class <code>cursor.MySQLCursorBuffered</code>	37
7.5. Class <code>constants.ClientFlag</code>	38
7.6. Class <code>constants.FieldType</code>	38
7.7. Class <code>constants.SQLMode</code>	38
7.8. Class <code>constants.CharacterSet</code>	39
7.9. Class <code>constants.RefreshOption</code>	39

This section contains the public API reference of Connector/Python. Although valid for both Python 2 and Python 3, examples should be considered working for Python 2.7, and Python 3.1 and greater.

The following overview shows the `mysql.connector` package with its modules. Currently, only the most useful modules, classes and functions for end users are documented.

```
mysql.connector
  errorcode
  errors
  connection
  constants
  conversion
  cursor
  dbapi
  locales
  eng
  client_error
  protocol
  utils
```

7.1. Errors and Exceptions

The `mysql.connector.errors` module defines exception classes for errors and warnings raised by MySQL Connector/Python. Most classes defined in this module are available when you import `mysql.connector`.

The exception classes defined in this module follow mostly the Python Database Specification v2.0 (PEP-249). For some MySQL client or server errors it is not always clear which exception to raise. It is good to discuss whether an error should be reclassified by opening a bug report.

MySQL Server errors are mapped with Python exception based on their SQLState (see [Server Error Codes and Messages](#)). The following list shows the `SQLState` classes and the exception Connector/Python will raise. It is, however, possible to redefine which exception is raised for each server error. Note that the default exception is `DatabaseError`.

- `02`: `DataError`
- `07`: `DatabaseError`
- `08`: `OperationalError`
- `0A`: `NotSupportedError`
- `21`: `DataError`
- `22`: `DataError`
- `23`: `IntegrityError`
- `24`: `ProgrammingError`
- `25`: `ProgrammingError`
- `26`: `ProgrammingError`
- `27`: `ProgrammingError`
- `28`: `ProgrammingError`
- `2A`: `ProgrammingError`
- `2B`: `DatabaseError`
- `2C`: `ProgrammingError`
- `2D`: `DatabaseError`
- `2E`: `DatabaseError`
- `33`: `DatabaseError`
- `34`: `ProgrammingError`
- `35`: `ProgrammingError`
- `37`: `ProgrammingError`
- `3C`: `ProgrammingError`
- `3D`: `ProgrammingError`

- `3F`: `ProgrammingError`
- `40`: `InternalError`
- `42`: `ProgrammingError`
- `44`: `InternalError`
- `HZ`: `OperationalError`
- `XA`: `IntegrityError`
- `0K`: `OperationalError`
- `HY`: `DatabaseError`

7.1.1. Module `errorcode`

This module contains both MySQL server and client error codes defined as module attributes with the error number as value. Using error codes instead of error numbers could make reading the source code a bit easier.

```
>>> from mysql.connector import errorcode
>>> errorcode.ER_BAD_TABLE_ERROR
1051
```

See [Server Error Codes and Messages](#) and [Client Error Codes and Messages](#).

7.1.2. Exception `errors.Error`

This exception is the base class for all other exceptions in the `errors` module. It can be used to catch all errors in a single `except` statement.

The following example shows how we could catch syntax errors:

```
import mysql.connector
try:
    cnx = mysql.connector.connect(user='scott', database='employees')
    cursor = cnx.cursor()
    cursor.execute("SELECT * FORM employees") # Syntax error in query
    cnx.close()
except mysql.connector.Error as err:
    print("Something went wrong: {}".format(err))
```

Initializing the exception supports a few optional arguments, namely `msg`, `errno`, `values` and `sqlstate`. All of them are optional and default to `None`. `errors.Error` is internally used by Connector/Python to raise MySQL client and server errors and should not be used by your application to raise exceptions.

The following examples show the result when using no or a combination of the arguments:

```
>>> from mysql.connector.errors import Error
>>> str(Error())
'Unknown error'
>>> str(Error("Oops! There was an error."))
'Oops! There was an error.'
>>> str(Error(errno=2006))
'2006: MySQL server has gone away'
>>> str(Error(errno=2002, values=('/tmp/mysql.sock', 2)))
'2002: Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)'
>>> str(Error(errno=1146, sqlstate='42S02', msg="Table 'test.spam' doesn't exist"))
```

```
"1146 (42S02): Table 'test.spam' doesn't exist"
```

The example which uses error number 1146 is used when Connector/Python receives an error packet from the MySQL Server. The information is parsed and passed to the `Error` exception as shown.

Each exception subclassing from `Error` can be initialized using the above mentioned arguments. Additionally, each instance has the attributes `errno`, `msg` and `sqlstate` which can be used in your code.

The following example shows how to handle errors when dropping a table which does not exist (when you do not want to use the `IF EXISTS` clause):

```
import mysql.connector
from mysql.connector import errorcode
cnx = mysql.connector.connect(user='scott', database='test')
try:
    cur.execute("DROP TABLE spam")
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_BAD_TABLE_ERROR:
        print("Creating table spam")
    else:
        raise
```

`errors.Error` is a subclass of the Python `StandardError`.

7.1.3. Exception `errors.Warning`

This exception is used for reporting important warnings, however, Connector/Python does not use it. It is included to be compliant with the Python Database Specification v2.0 (PEP-249).

Consider using either more strict `Server SQL Modes` or the `raise_on_warnings` connection argument to make Connector/Python raise errors when your queries produce warnings.

`errors.Warning` is a subclass of the Python `StandardError`.

7.1.4. Exception `errors.InterfaceError`

This exception is raised for errors originating from Connector/Python itself, not related to the MySQL server.

`errors.InterfaceError` is a subclass of `errors.Error`.

7.1.5. Exception `errors.DatabaseError`

This exception is the default for any MySQL error which does not fit the other exceptions.

`errors.DatabaseError` is a subclass of `errors.Error`.

7.1.6. Exception `errors.InternalError`

This exception is raised when the MySQL server encounters an internal error, for example, when a deadlock occurred.

`errors.InternalError` is a subclass of `errors.DatabaseError`.

7.1.7. Exception `errors.OperationalError`

This exception is raised for errors which are related to MySQL's operations. For example, to many connections, a hostname could not be resolved, bad handshake, server is shutting down, communication errors, and so on.

`errors.OperationalError` is a subclass of `errors.DatabaseError`.

7.1.8. Exception `errors.ProgrammingError`

This exception is raised on programming errors, for example when you have a syntax error in your SQL or a table was not found.

The following example shows how to handle syntax errors:

```
try:
    cursor.execute("CREATE DESK t1 (id int, PRIMARY KEY (id))")
except mysql.connector.ProgrammingError as err:
    if err.errno == errorcode.ER_SYNTAX_ERROR:
        print("Check your syntax!")
    else:
        print("Error: {}".format(err))
```

`errors.ProgrammingError` is a subclass of `errors.DatabaseError`.

7.1.9. Exception `errors.IntegrityError`

This exception is raised when the relational integrity of the data is affected. For example, a duplicate key was inserted or a foreign key constraint would fail.

The following example shows a duplicate key error raised as `IntegrityError`:

```
cursor.execute("CREATE TABLE t1 (id int, PRIMARY KEY (id))")
try:
    cursor.execute("INSERT INTO t1 (id) VALUES (1)")
    cursor.execute("INSERT INTO t1 (id) VALUES (1)")
except mysql.connector.IntegrityError as err:
    print("Error: {}".format(err))
```

`errors.IntegrityError` is a subclass of `errors.DatabaseError`.

7.1.10. Exception `errors.DataError`

This exception is raised when there were problems with the data. Examples are a column set to NULL when it can not, out of range values for a column, division by zero, column count does not match value count, and so on.

`errors.DataError` is a subclass of `errors.DatabaseError`.

7.1.11. Exception `errors.NotSupportedError`

This exception is raised is case some feature was used but not supported by the version of MySQL which returned the error. It is also raised when using functions or statements which are not supported by stored routines.

`errors.NotSupportedError` is a subclass of `errors.DatabaseError`.

7.1.12. Function `errors.custom_error_exception(error=None, exception=None)`

This function defines custom exceptions for MySQL server errors and returns current customizations.

If `error` is a MySQL Server error number, then you have to pass also the `exception` class. The `error` argument can also be a dictionary in which case the key is the server error number, and value the class of the exception to be raised.

To reset the customizations, simply supply an empty dictionary.

```
import mysql.connector
from mysql.connector import errorcode
# Server error 1028 should raise a DatabaseError
mysql.connector.custom_error_exception(1028, mysql.connector.DatabaseError)
# Or using a dictionary:
mysql.connector.custom_error_exception({
    1028: mysql.connector.DatabaseError,
    1029: mysql.connector.OperationalError,
})
# To reset, pass an empty dictionary:
mysql.connector.custom_error_exception({})
```

7.2. Class `connection.MySQLConnection`

The `MySQLConnection` class is used to open and manage a connection to a MySQL server. It also used to send commands and SQL queries and read result.

7.2.1. Constructor `connection.MySQLConnection(**kwargs)`

The `MySQLConnection` constructor initializes the attributes and when at least one argument is passed, it tries to connect with the MySQL server.

For a complete list of arguments, see [Chapter 6, Connector/Python Connection Arguments](#).

7.2.2. Method `MySQLConnection.close()`

See [disconnect\(\)](#).

Returns a tuple.

7.2.3. Method `MySQLConnection.config(**kwargs)`

Allows to configure a `MySQLConnection` instance after it was instantiated. See [Chapter 6, Connector/Python Connection Arguments](#) for a complete list of possible arguments.

You could use the `config()` method to change, for example, the username and call [reconnect\(\)](#).

```
cnx = MySQLConnection(user='joe', database='test')
# Connected as 'joe'
cnx.config(user='jane')
cnx.reconnect()
# Now connected as 'jane'
```

7.2.4. Method `MySQLConnection.connect(**kwargs)`

This method sets up the connection to the MySQL server. If no arguments are given, it uses the already configured or default values. See [Chapter 6, Connector/Python Connection Arguments](#) for a complete list of possible arguments.

7.2.5. Method `MySQLConnection.commit()`

This method sends the `COMMIT` command to the MySQL server, committing the current transaction. Since by default, Connector/Python does not auto commit, it is important to call this method after every transaction which updates data for tables using transactional storage engines.

See the [rollback\(\)](#) method for rolling back transactions.

Method `MySQLConnection.cursor(buffered=None, raw=None, cursor_class=None)`

```
>>> cursor.execute("INSERT INTO employees (first_name) VALUES (%s)", ('Jane'))
>>> cnx.commit()
```

7.2.6. Method `MySQLConnection.cursor(buffered=None, raw=None, cursor_class=None)`

This method returns a `MySQLCursor()` object, or a subclass of it depending the passed arguments.

When `buffered` is `True`, the cursor will fetch all rows after the operation is executed. This is useful when queries return small result sets. Setting `raw` will skip the conversion from MySQL data types to Python types when fetching rows. `Raw` is usually used when you want to have more performance and/or you want to do the conversion yourself.

The `cursor_class` argument can be used to pass a class to use for instantiating a new cursor. It has to be a subclass of `cursor.CursorBase`.

The returned object depends on the combination of the `buffered` and `raw` arguments.

- If not buffered and not raw: `cursor.MySQLCursor`
- If buffered and not raw: `cursor.MySQLCursorBuffered`
- If buffered and raw: `cursor.MySQLCursorBufferedRaw`
- If not buffered and raw: `cursor.MySQLCursorRaw`

Returns a `CursorBase` instance.

7.2.7. Method `MySQLConnection.cmd_change_user(username='', password='', database='', charset=33)`

Changes the user using `username` and `password`. It also causes the specified `database` to become the default (current) database. It is also possible to change the character set using the `charset` argument.

Returns a dictionary containing the OK packet information.

7.2.8. Method `MySQLConnection.cmd_debug()`

Instructs the server to write some debug information to the log. For this to work, the connected user must have the `SUPER` privilege.

Returns a dictionary containing the OK packet information.

7.2.9. Method `MySQLConnection.cmd_init_db(database)`

This method makes specified database the default (current) database. In subsequent queries, this database is the default for table references that do not include an explicit database specifier.

Returns a dictionary containing the OK packet information.

7.2.10. Method `MySQLConnection.cmd_ping()`

Checks whether the connection to the server is working.

This method is not to be used directly. Use `ping()` or `is_connected()` instead.

Returns a dictionary containing the OK packet information.

7.2.11. Method `MySQLConnection.cmd_process_info()`

This method raises the `NotSupportedError` exception. Instead, use the `SHOW PROCESSLIST` statement or query the tables found in the database `INFORMATION_SCHEMA`.

7.2.12. Method `MySQLConnection.cmd_process_kill(mysql_pid)`

Asks the server to kill the thread specified by `mysql_pid`. Although still available, it's better to use the SQL `KILL` command.

Returns a dictionary containing the OK packet information.

The following two lines do the same:

```
>>> cnx.cmd_process_kill(123)
>>> cnx.cmd_query('KILL 123')
```

7.2.13. Method `MySQLConnection.cmd_quit()`

This method sends the `QUIT` command to the MySQL server, closing the current connection. Since there is no response from the MySQL, the packet that was sent is returned.

7.2.14. Method `MySQLConnection.cmd_query(statement)`

This method sends the given `statement` to the MySQL server and returns a result. If you need to send multiple statements, you have to use the `cmd_query_iter()` method.

The returned dictionary contains information depending on what kind of query was executed. If the query is a `SELECT` statement, the result contains information about columns. Other statements return a dictionary containing OK or EOF packet information.

Errors received from the MySQL server are raised as exceptions. An `InterfaceError` is raised when multiple results are found.

Returns a dictionary.

7.2.15. Method `MySQLConnection.cmd_query_iter(statement)`

Similar to the `cmd_query()` method, but returns a generator object to iterate through results. Use `cmd_query_iter()` when sending multiple statements, and separate the statements with semicolons.

The following example shows how to iterate through the results after sending multiple statements:

```
statement = 'SELECT 1; INSERT INTO t1 VALUES (); SELECT 2'
for result in cnx.cmd_query_iter(statement, iterate=True):
    if 'columns' in result:
        columns = result['columns']
        rows = cnx.get_rows()
    else:
        # do something useful with INSERT result
```

Returns a generator object.

7.2.16. Method `MySQLConnection.cmd_refresh(options)`

This method flushes tables or caches, or resets replication server information. The connected user must have the `RELOAD` privilege.

The `options` argument should be a bitwise value using constants from the class `constants.RefreshOption`.

See [Section 7.9](#), “Class `constants.RefreshOption`” for a list of options.

Example:

```
>>> from mysql.connector import RefreshOption
>>> refresh = RefreshOption.LOG | RefreshOption.THREADS
>>> cnx.cmd_refresh(refresh)
```

7.2.17. Method `MySQLConnection.cmd_shutdown()`

Asks the database server to shut down. The connected user must have the `SHUTDOWN` privilege.

Returns a dictionary containing the OK packet information.

7.2.18. Method `MySQLConnection.cmd_statistics()`

Returns a dictionary containing information about the MySQL server including uptime in seconds and the number of running threads, questions, reloads, and open tables.

7.2.19. Method `MySQLConnection.disconnect()`

This method tries to send the `QUIT` command and close the socket. It does not raise any exceptions.

`MySQLConnection.close()` is a synonymous method name and more commonly used.

7.2.20. Method `MySQLConnection.get_rows(count=None)`

This method retrieves all or remaining rows of a query result set, returning a tuple containing the rows as sequence and the EOF packet information. The `count` argument can be used to get a given amount of rows. If `count` is not specified or is `None`, all rows are retrieved.

The tuple returned by `get_rows()` consists of:

- A list of tuples containing the row data as byte objects, or an empty list when no rows are available.
- EOF packet information as a dictionary containing `status_flag` and `warning_count`.

An `InterfaceError` is raised when all rows have been retrieved.

The `get_rows()` method is used by `MySQLCursor` to fetch rows.

Returns a tuple.

7.2.21. Method `MySQLConnection.get_row()`

This method retrieves the next row of a query result set, returning a tuple.

The tuple returned by `get_row()` consists of:

- The row as a tuple containing byte objects, or `None` when no more rows are available.
- EOF packet information as a dictionary containing `status_flag` and `warning_count`, or `None` when the row returned is not the last row.

The `get_row()` method is used by `MySQLCursor` to fetch rows.

7.2.22. Method `MySQLConnection.get_server_info()`

This method returns the MySQL server information verbatim, for example `'5.5.24-log'`, or `None` when not connected.

Returns a string or `None`.

7.2.23. Method `MySQLConnection.get_server_version()`

This method returns the MySQL server version as a tuple, or `None` when not connected.

Returns a tuple or `None`.

7.2.24. Method `MySQLConnection.is_connected()`

Reports whether the connection to MySQL Server is available.

This method checks whether the connection to MySQL is available using the `ping()` method, but unlike `ping()`, `is_connected()` returns `True` when the connection is available, `False` otherwise.

Returns `True` or `False`.

7.2.25. Method `MySQLConnection.isset_client_flag(flag)`

This method returns `True` if the client flag was set, `False` otherwise.

Returns `True` or `False`.

7.2.26. Method `MySQLConnection.ping(attempts=1, delay=0)`

Check whether the connection to the MySQL server is still available.

When `reconnect` is set to `True`, one or more attempts are made to try to reconnect to the MySQL server using the `reconnect()` method. Use the `delay` argument (seconds) if you want to wait between each retry.

When the connection is not available, an `InterfaceError` is raised. Use the `is_connected()` method to check the connection without raising an error.

Raises `InterfaceError` on errors.

7.2.27. Method `MySQLConnection.reconnect(attempts=1, delay=0)`

Attempt to reconnect with the MySQL server.

The argument `attempts` specifies the number of times a reconnect is tried. The `delay` argument is the number of seconds to wait between each retry.

You might set the number of attempts higher and use a longer delay when you expect the MySQL server to be down for maintenance, or when you expect the network to be temporarily unavailable.

7.2.28. Method `MySQLConnection.rollback()`

This method sends the `ROLLBACK` command to the MySQL server, undoing all data changes from the current `transaction`. Since by default, Connector/Python does not auto commit, it is possible to cancel transactions when using transactional storage engines such as `InnoDB`.

See the `commit()` method for `committing` transactions.

```
>>> cursor.execute("INSERT INTO employees (first_name) VALUES (%s)", ('Jane'))
>>> cnx.rollback()
```

7.2.29. Method `MySQLConnection.set_charset_collation(charset=None, collation=None)`

This method sets the character set and collation to be used for the current connection. The `charset` argument can be either the name of a character set, or the numerical equivalent as defined in `constants.CharacterSet`.

When `collation` is `None`, the default will be looked up and used.

The `charset` argument then be either:

In the following example, we set the character set to `latin1` and the collation will be set to the default `latin1_swedish_ci`:

```
>>> cnx = mysql.connector.connect(user='scott')
>>> cnx.set_charset('latin1')
```

Specify a specific collation as follows:

```
>>> cnx = mysql.connector.connect(user='scott')
>>> cnx.set_charset('latin1', 'latin1_general_ci')
```

7.2.30. Method `MySQLConnection.set_client_flags(flags)`

This method sets the client flags which are used when connecting with the MySQL server and returns the new value. The `flags` argument can be either an integer or a sequence of valid client flag values (see [Section 7.5, “Class constants.ClientFlag”](#)).

If `flags` is a sequence, each item in the sequence will set the flag when the value is positive or unset it when negative. For example, to unset `LONG_FLAG` and set the `FOUND_ROWS` flags:

```
>>> from mysql.connector.constants import ClientFlag
>>> cnx.set_client_flags([ClientFlag.FOUND_ROWS, -ClientFlag.LONG_FLAG])
>>> cnx.reconnect()
```

Note that client flags are only set or used when connecting with the MySQL server. It is therefore necessary to reconnect after making changes.

Returns an integer.

7.2.31. Property `MySQLConnection.autocommit`

This property is used to toggle the auto commit feature of MySQL and retrieve the current state. When the value evaluates to True, auto commit will be turned, otherwise it is turned off.

Note that auto commit is disabled by default when connecting through Connector/Python. This can be toggled using the [connection parameter](#) `autocommit`.

When the auto commit is turned off, you have to [commit](#) transactions when using transactional storage engines such as InnoDB or NDBCluster.

```
>>> cnx.autocommit
False
>>> cnx.autocommit = True
>>> cnx.autocommit
True
```

Returns True or False.

7.2.32. Property `MySQLConnection.charset_name`

This property returns which character set is used for the connection whether it is connected or not.

Returns a string.

7.2.33. Property `MySQLConnection.collation_name`

This property returns which collation is used for the connection whether it is connected or not.

Returns a string.

7.2.34. Property `MySQLConnection.connection_id`

This property returns the connection ID (thread ID or session ID) for the current connection or None when not connected.

Returns a integer or None.

7.2.35. Property `MySQLConnection.database`

This property is used to set current (active) database executing the [USE](#) command. The property can also be used to retrieve the current database name.

```
>>> cnx.database = 'test'
>>> cnx.database = 'mysql'
>>> cnx.database
u'mysql'
```

Returns a string.

7.2.36. Property `MySQLConnection.get_warnings`

This property is used to toggle whether warnings should be fetched automatically or not. It accepts True or False (default).

Fetching warnings automatically could be useful when debugging queries. Cursors will make warnings available through the method `MySQLCursor.fetchwarnings()`.

```
>>> cnx.get_warnings = True
>>> cursor.execute('SELECT "a"+1')
>>> cursor.fetchall()
[(1.0,)]
>>> cursor.fetchwarnings()
[(u'Warning', 1292, u'Truncated incorrect DOUBLE value: 'a'")]
```

Returns True or False.

7.2.37. Property `MySQLConnection.raise_on_warnings`

This property is used to toggle whether warnings should raise exceptions or not. It accepts True or False (default).

Toggling `raise_on_warnings` will also toggle `get_warnings` since warnings need to be fetched so they can be raised as exceptions.

Note that you might always want to check setting SQL Mode if you would like to have the MySQL server directly report warnings as errors. It is also good to use transactional engines so transactions can be rolled back when catching the exception.

Result sets needs to be fetched completely before any exception can be raised. The following example shows the execution of a query which produces a warning

```
>>> cnx.raise_on_warnings = True
>>> cursor.execute('SELECT "a"+1')
>>> cursor.fetchall()
..
mysql.connector.errors.DataError: 1292: Truncated incorrect DOUBLE value: 'a'
```

Returns True or False.

7.2.38. Property `MySQLConnection.server_host`

This read-only property returns the hostname or IP address used for connecting with the MySQL server.

Returns a string.

7.2.39. Property `MySQLConnection.server_port`

This read-only property returns the TCP/IP port used for connecting with the MySQL server.

Returns a integer.

7.2.40. Property `MySQLConnection.sql_mode`

This property is used to retrieve and set the SQL Modes for the current. The value should be list of different modes separated by comma (","), or a sequence of modes, preferably using the constants.SQLMode class.

To unset all modes, pass an empty string or an empty sequence.

```
>>> cnx.sql_mode = 'TRADITIONAL,NO_ENGINE_SUBSTITUTION'
>>> cnx.sql_mode.split(',')
[u'STRICT_TRANS_TABLES', u'STRICT_ALL_TABLES', u'NO_ZERO_IN_DATE',
u'NO_ZERO_DATE', u'ERROR_FOR_DIVISION_BY_ZERO', u'TRADITIONAL',
u'NO_AUTO_CREATE_USER', u'NO_ENGINE_SUBSTITUTION']
>>> from mysql.connector.constants import SQLMode
>>> cnx.sql_mode = [ SQLMode.NO_ZERO_DATE, SQLMode.REAL_AS_FLOAT]
>>> cnx.sql_mode
u'REAL_AS_FLOAT,NO_ZERO_DATE'
```

Returns a string.

7.2.41. Property `MySQLConnection.time_zone`

This property is used to set the time zone session variable for the current connection and retrieve it.

```
>>> cnx.time_zone = '+00:00'
>>> cur.execute('SELECT NOW()') ; cur.fetchone()
(datetime.datetime(2012, 6, 15, 11, 24, 36),)
>>> cnx.time_zone = '-09:00'
>>> cur.execute('SELECT NOW()') ; cur.fetchone()
(datetime.datetime(2012, 6, 15, 2, 24, 44),)
>>> cnx.time_zone
u'-09:00'
```

Returns a string.

7.2.42. Property `MySQLConnection.unix_socket`

This read-only property returns the UNIX socket user for connecting with the MySQL server.

Returns a string.

7.2.43. Property `MySQLConnection.user`

This read-only property returns the username used for connecting with the MySQL server.

Returns a string.

7.3. Class `cursor.MySQLCursor`

The `MySQLCursor` class is used to instantiate object which can execute operation such as SQL queries. They interact with the MySQL server using a `MySQLConnection` object.

7.3.1. Constructor `cursor.MySQLCursor`

The constructor initializes the instance with the optional `connection`, which should be an instance of `MySQLConnection`.

In most cases, the `MySQLConnection` method `cursor()` is used to instantiate a `MySQLCursor` object.

7.3.2. Method `MySQLCursor.callproc(procname, args=())`

This method calls a stored procedure with the given name. The `args` sequence of parameters must contain one entry for each argument that the routine expects. The result is returned as modified copy of

the input sequence. Input parameters are left untouched, output and input/output parameters replaced with possibly new values.

Result set provided by the stored procedure are automatically fetched and stored as `MySQLBufferedCursor` instances. See `stored_results()` for more information.

The following example shows how to execute a stored procedure which takes two parameters, multiplies the values and returns the product:

```
# Definition of the multiply stored procedure:
# CREATE PROCEDURE multiply(IN pFac1 INT, IN pFac2 INT, OUT pProd INT)
# BEGIN
#   SET pProd := pFac1 * pFac2;
# END
>>> args = (5, 5, 0) # 0 is to hold value of the OUT parameter pProd
>>> cursor.callproc('multiply', args)
('5', '5', 25L)
```

7.3.3. Method `MySQLCursor.close()`

This method will close the MySQL cursor, resetting all results and removing the connection.

Use `close()` every time you are done using the cursor.

7.3.4. Method `MySQLCursor.execute(operation, params=None, multi=False)`

This method prepare the given database `operation` (query or command). The parameters found in the tuple or dictionary `params` will be bound to the variables in the operation. Variables are specified using `%s` markers or named markers `%(name)s`.

For example, insert information about a new employee and selecting again the data of this person:

```
insert = (
"INSERT INTO employees (emp_no, first_name, last_name, hire_date) "
"VALUES (%s, %s, %s, %s)"
data = (2, 'Jane', 'Doe', datetime.date(2012, 3, 23))
cursor.execute(insert, data)
select = "SELECT * FROM employees WHERE emp_no = %(emp_no)s"
cursor.execute(select, { 'emp_no': 2 })
```

Note that the data is converted from Python object to something MySQL understand. In the above example, the `datetime.date()` instance is converted to `'2012-03-23'` in the above example.

When `multi` is set to `True`, `execute()` will be able to execute multiple statements. It will return an iterator which makes it possible to go through all results for each statement. Note that using parameters is not working well in this case, and it's usually a good idea to execute each statement on its own.

In the following example we select and insert data in one operation and display the result:

```
operation = 'SELECT 1; INSERT INTO t1 VALUES (); SELECT 2'
for result in cursor.execute(operation):
    if result.with_rows:
        print("Statement '{}' has following rows:".format(
            result.statement))
        print(result.fetchall())
    else:
        print("Affected row(s) by query '{}' was {}".format(
            result.statement, result.rowcount))
```

If the connection was configured to fetch warnings, warnings generated by the operation will be available through the method `MySQLCursor.fetchwarnings()`.

Returns an iterator when `multi` is `True`.

7.3.5. Method `MySQLCursor.executemany(operation, seq_params)`

This method prepares a database operation (query or command) and then execute it against all parameter sequences or mappings found in the sequence `seq_of_params`.

The `executemany()` is simply iterating through the sequence of parameters calling the `execute()` method. Inserting data, however, is optimized by batching them using the multiple rows syntax.

In the following example we are inserting 3 records:

```
data = [
    ('Jane', date(2005, 2, 12)),
    ('Joe', date(2006, 5, 23)),
    ('John', date(2010, 10, 3)),
]
stmt = "INSERT INTO employees (first_name, hire_date) VALUES (%s, %s)"
cursor.executemany(stmt, data)
```

In the above example, the INSERT statement sent to MySQL would be as follows: `INSERT INTO employees (first_name, hire_date) VALUES ('Jane', '2005-02-12'), ('Joe', '2006-05-23'), ('John', '2010-10-03')`.

Note that it is not possible to execute multiple statements using the `executemany()` method. Doing so will raise an `InternalError` exception.

7.3.6. Method `MySQLCursor.fetchall()`

The method fetches all or remaining rows of a query result set, returning a list of tuples. An empty list is returned when no rows are (anymore) available.

The following examples shows how to retrieve the first 2 rows of a result set, and then retrieve the remaining rows:

```
>>> cursor.execute("SELECT * FROM employees ORDER BY emp_no")
>>> head_rows = cursor.fetchmany(size=2)
>>> remaining_rows = cursor.fetchall()
```

Note that you have to fetch all rows before being able to execute new queries using the same connection.

Returns a list of tuples or empty list when no rows available.

7.3.7. Method `MySQLCursor.fetchmany(size=1)`

This method fetches the next set of rows of a query results, returning a list of tuples. An empty list is returned when no more rows are available.

The number of rows returned can be specified using the `size` argument, which defaults to one. Fewer rows might be returned, when there are not more rows available than specified by the argument.

Note that you have to fetch all rows before being able to execute new queries using the same connection.

Returns a list of tuples or empty list when no rows available.

7.3.8. Method `MySQLCursor.fetchone()`

This method retrieves the next row of a query result set, returning a single sequence, or `None` when no more data is available. The returned tuple consists of data returned by the MySQL server converted to Python objects.

The `fetchone()` method is used by `fetchmany()` and `fetchall()`. It is also used when using the `MySQLCursor` instance as an iterator.

The following examples show how to iterate through the result of a query using `fetchone()`:

```
# Using a while-loop
cursor.execute("SELECT * FROM employees")
row = cursor.fetchone()
while row is not None:
    print(row)
    row = cursor.fetchone()
# Using the cursor as iterator
cursor.execute("SELECT * FROM employees")
for row in cursor:
    print(row)
```

Note that you have to fetch all rows before being able to execute new queries using the same connection.

Returns a tuple or `None`.

7.3.9. Method `MySQLCursor.fetchwarnings()`

This method returns a list of tuples containing warnings generated by previously executed statement. Use the connection's `get_warnings` property to toggle whether warnings has to be fetched.

The following example shows a `SELECT` statement which generated a warning:

```
>>> cnx.get_warnings = True
>>> cursor.execute('SELECT "a"+1')
>>> cursor.fetchall()
[(1.0,)]
>>> cursor.fetchwarnings()
[(u'Warning', 1292, u'Truncated incorrect DOUBLE value: 'a'')]
```

It is also possible to raise errors when warnings are found. See the `MySQLConnection` property `raise_on_warnings`.

Returns a list of tuples.

7.3.10. Method `MySQLCursor.stored_results()`

This method returns an list iterator object which can be used to go through result sets provided by stored procedures after calling them using the `callproc()` method.

In the following example we execute a stored procedure which will provide two result sets. We use `stored_results()` to retrieve them:

```
>>> cursor.callproc('sp1')
()
>>> for result in cursor.stored_results():
...     print result.fetchall()
... 
```

```
[(1,)]
[(2,)]
```

Note that the result sets stay available until you executed another operation or call another stored procedure.

Returns a listiterator.

7.3.11. Property `MySQLCursor.column_names`

This read-only property returns the column names of a result set as sequence of (unicode) strings.

The following example shows how you can create a dictionary out of a tuple containing data with keys using `column_names`:

```
cursor.execute("SELECT last_name, first_name, hire_date "
              "FROM employees WHERE emp_no = %s", (123,))
row = dict(zip(cursor.column_names, cursor.fetchone()))
print("{last_name}, {first_name}: {hire_date}".format(row))
```

Returns a tuple.

7.3.12. Property `MySQLCursor.statement`

This read-only property returns the last executed statement. In case multiple statements were executed, it will show the actual statement.

The `statement` property might be useful for debugging and showing what was send to the MySQL server.

Returns a string.

7.3.13. Property `MySQLCursor.with_rows`

This read-only property will return True when the result of the executed operation provides rows.

The `with_rows` property is useful when executing multiple statements and you need to fetch rows. In the following example we only report the affected rows by the `UPDATE` statement:

```
import mysql.connector
cnx = mysql.connector.connect(user='scott', database='test')
cursor = cnx.cursor()
operation = 'SELECT 1; UPDATE t1 SET c1 = 2; SELECT 2'
for result in cursor.execute(operation, multi=True):
    if result.with_rows:
        result.fetchall()
    else:
        print("Updated row(s): {}".format(result.rowcount))
```

7.4. Class `cursor.MySQLCursorBuffered`

This class is inheriting from `cursor.MySQLCursor` and if needed automatically retrieves rows after an operation has been executed.

`MySQLCursorBuffered` can be useful in situations where two queries, with small result sets, need to be combined or computed with each other.

You can either use the `buffered` argument when using the connection's `cursor()` method, or you can use the `buffered connection option` to make all created cursors by default buffering.

```
import mysql.connector
cnx = mysql.connector.connect()
# Only this particular cursor will be buffering results
cursor.cursor(buffered=True)
# All cursors by default buffering
cnx = mysql.connector.connect(buffered=True)
```

See [Section 5.1, “Tutorial: Raise employee's salary using a buffering cursor”](#) for a practical use case.

7.5. Class `constants.ClientFlag`

This class provides constants defining MySQL client flags which can be used upon connection to configure the session. The `ClientFlag` class is available when importing `mysql.connector`.

```
>>> import mysql.connector
>>> mysql.connector.constants.ClientFlag.FOUND_ROWS
2
```

See [Section 7.2.30, “Method `MySQLConnection.set_client_flags\(flags\)`”](#) and the [connection argument `client_flag`](#).

Note that the `ClientFlag` class can not be instantiated.

7.6. Class `constants.FieldType`

This class provides all supported MySQL field or data types. They can be useful when dealing with raw data or defining your own converters. The field type is stored with every cursor in the description for each column.

The following example shows how you can print the name of the data types for each of the columns in the result set.

```
from __future__ import print_function
import mysql.connector
from mysql.connector import FieldType
cnx = mysql.connector.connect(user='scott', database='test')
cursor = cnx.cursor()
cursor.execute(
    "SELECT DATE(NOW()) AS `c1`, TIME(NOW()) AS `c2`, "
    "NOW() AS `c3`, 'a string' AS `c4`, 42 AS `c5`")
rows = cursor.fetchall()
for desc in cursor.description:
    colname = desc[0]
    coltype = desc[1]
    print("Column {} has type {}".format(
        colname, FieldType.get_info(coltype)))
cursor.close()
cnx.close()
```

Note that the `FieldType` class can not be instantiated.

7.7. Class `constants.SQLMode`

This class provides all known MySQL [Server SQL Modes](#). It is mostly used when setting the SQL modes at connection time using the connection's property `sql_mode`. See [Section 7.2.40, “Property `MySQLConnection.sql_mode`”](#).

Note that the `SQLMode` class can not be instantiated.

7.8. Class `constants.CharacterSet`

This class provides all known MySQL characters sets and their default collations. See [Section 7.2.29](#), “[Method `MySQLConnection.set_charset_collation\(charset=None, collation=None\)`](#)” for examples.

Note that the `CharacterSet` class can not be instantiated.

7.9. Class `constants.RefreshOption`

- `RefreshOption.GRANT`

Refresh the grant tables, like `FLUSH PRIVILEGES`.

- `RefreshOption.LOG`

Flush the logs, like `FLUSH LOGS`.

- `RefreshOption.TABLES`

Flush the table cache, like `FLUSH TABLES`.

- `RefreshOption.HOSTS`

Flush the host cache, like `FLUSH HOSTS`.

- `RefreshOption.STATUS`

Reset status variables, like `FLUSH STATUS`.

- `RefreshOption.THREADS`

Flush the thread cache.

- `RefreshOption.SLAVE`

On a slave replication server, reset the master server information and restart the slave, like `RESET SLAVE`.

- `RefreshOption.MASTER`

On a master replication server, remove the binary log files listed in the binary log index and truncate the index file, like `RESET MASTER`.

Appendix A. Licenses for Third-Party Components

Table of Contents

A.1. <code>dtoa.c</code> License	42
A.2. Editline Library (<code>libedit</code>) License	42
A.3. <code>FindGTest.cmake</code> License	44
A.4. Fred Fish's Dbug Library License	45
A.5. <code>getarg</code> License	46
A.6. GNU General Public License Version 2.0, June 1991	47
A.7. GNU Lesser General Public License Version 2.1, February 1999	52
A.8. GNU Libtool License	60
A.9. GNU Readline License	61
A.10. Google Controlling Master Thread I/O Rate Patch License	61
A.11. Google Perftools (TCMalloc utility) License	62
A.12. Google SMP Patch License	62
A.13. <code>lib_sql.cc</code> License	63
A.14. Libaio License	63
A.15. <code>libevent</code> License	63
A.16. Linux-PAM License	65
A.17. md5 (Message-Digest Algorithm 5) License	66
A.18. <code>nt_servc</code> (Windows NT Service class library) License	66
A.19. OpenPAM License	67
A.20. Percona Multiple I/O Threads Patch License	67
A.21. Red HAT RPM Spec File License	68
A.22. RegEX-Spencer Library License	68
A.23. RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License	69
A.24. Richard A. O'Keefe String Library License	69
A.25. SHA-1 in C License	70
A.26. <code>zlib</code> License	70

MySQL 5.5

- [Section A.1, “`dtoa.c` License”](#)
- [Section A.2, “Editline Library \(`libedit`\) License”](#)
- [Section A.3, “`FindGTest.cmake` License”](#)
- [Section A.4, “Fred Fish's Dbug Library License”](#)
- [Section A.5, “`getarg` License”](#)
- [Section A.6, “GNU General Public License Version 2.0, June 1991”](#)
- [Section A.7, “GNU Lesser General Public License Version 2.1, February 1999”](#)
- [Section A.8, “GNU Libtool License”](#)
- [Section A.9, “GNU Readline License”](#)
- [Section A.10, “Google Controlling Master Thread I/O Rate Patch License”](#)
- [Section A.11, “Google Perftools \(TCMalloc utility\) License”](#)

- [Section A.12, "Google SMP Patch License"](#)
- [Section A.13, "lib_sql.cc License"](#)
- [Section A.14, "Libaio License"](#)
- [Section A.15, "libevent License"](#)
- [Section A.16, "Linux-PAM License"](#)
- [Section A.17, "md5 \(Message-Digest Algorithm 5\) License"](#)
- [Section A.18, "nt_servc \(Windows NT Service class library\) License"](#)
- [Section A.19, "OpenPAM License"](#)
- [Section A.20, "Percona Multiple I/O Threads Patch License"](#)
- [Section A.21, "Red HAT RPM Spec File License"](#)
- [Section A.22, "RegEX-Spencer Library License"](#)
- [Section A.23, "RFC 3174 - US Secure Hash Algorithm 1 \(SHA1\) License"](#)
- [Section A.24, "Richard A. O'Keefe String Library License"](#)
- [Section A.25, "SHA-1 in C License"](#)
- [Section A.26, "zlib License"](#)

A.1. `dtoa.c` License

The following software may be included in this product:

`dtoa.c`

```
The author of this software is David M. Gay.

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for
any purpose without fee is hereby granted, provided that this entire
notice is included in all copies of any software which is or includes
a copy or modification of this software and in all copies of the
supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR
IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT
MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE
MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR
PURPOSE.
```

A.2. Editline Library (`libedit`) License

The following software may be included in this product:

Editline Library (`libedit`)

Some files are:

```
Copyright (c) 1992, 1993
```

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some files are:

Copyright (c) 2001 The NetBSD Foundation, Inc.
All rights reserved.

This code is derived from software contributed to The NetBSD Foundation by Anthony Mallet.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

```
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

Some files are:

```
Copyright (c) 1997 The NetBSD Foundation, Inc.
All rights reserved.
```

```
This code is derived from software contributed to The NetBSD Foundation
by Jaromir Dolecek.
```

```
Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

```
THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC.
AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

Some files are:

```
Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>
```

```
Permission to use, copy, modify, and distribute this
software for any purpose with or without fee is hereby
granted, provided that the above copyright notice and
this permission notice appear in all copies.
```

```
THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER
DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE
FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.
```

A.3. FindGTest.cmake License

The following software may be included in this product:

```
FindGTest.cmake helper script (part of CMake)

Copyright 2009 Kitware, Inc.
Copyright 2009 Philip Lowman
Copyright 2009 Daniel Blezek

Distributed under the OSI-approved BSD License (the "License");
see accompanying file Copyright.txt for details.

This software is distributed WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the License for more information.
=====
(To distributed this file outside of CMake, substitute the full
License text for the above reference.)

Thanks to Daniel Blezek for the GTEST_ADD_TESTS code

Text of Copyright.txt mentioned above:

CMake - Cross Platform Makefile Generator
Copyright 2000-2009 Kitware, Inc., Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium,
nor the names of their contributors may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

A.4. Fred Fish's Dbug Library License

The following software may be included in this product:

Fred Fish's Dbug Library

N O T I C E

Copyright Abandoned, 1987, Fred Fish

```
This previously copyrighted work has been placed into the public
domain by the author and may be freely used for any purpose,
private or commercial.
```

```
Because of the number of inquiries I was receiving about the use
of this product in commercially developed works I have decided to
simply make it public domain to further its unrestricted use. I
specifically would be most happy to see this material become a
part of the standard Unix distributions by AT&T and the Berkeley
Computer Science Research Group, and a standard part of the GNU
system from the Free Software Foundation.
```

```
I would appreciate it, as a courtesy, if this notice is left in
all copies and derivative works. Thank you.
```

```
The author makes no warranty of any kind with respect to this
product and explicitly disclaims any implied warranties of mer-
chantability or fitness for any particular purpose.
```

The `debug_analyze.c` file is subject to the following notice:

```
Copyright June 1987, Binayak Banerjee
All rights reserved.
```

```
This program may be freely distributed under the same terms and
conditions as Fred Fish's Dbug package.
```

A.5. **getarg** License

The following software may be included in this product:

getarg Function (`getarg.h`, `getarg.c` files)

```
Copyright (c) 1997 - 2000 Kungliga Tekniska Högskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.
```

```
Redistribution and use in source and binary forms, with
or without modification, are permitted provided that the
following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.6. GNU General Public License Version 2.0, June 1991

The following applies to all products licensed under the GNU General Public License, Version 2.0: You may not use the identified files except in compliance with the GNU General Public License, Version 2.0 (the "License.") You may obtain a copy of the License at <http://www.gnu.org/licenses/gpl-2.0.txt>. A copy of the license is also reproduced below. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim
copies of this license document, but changing it is not
allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that

you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it
does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version
2 of the License, or (at your option) any later version.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type 'show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type 'show c'
for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program 'Gnomovision' (which makes passes at compilers) written
by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

A.7. GNU Lesser General Public License Version 2.1, February 1999

The following applies to all products licensed under the GNU Lesser General Public License, Version 2.1: You may not use the identified files except in compliance with the GNU Lesser General Public License, Version 2.1 (the "License"). You may obtain a copy of the License at <http://www.gnu.org/licenses/lgpl-2.1.html>. A copy of the license is also reproduced below. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using

a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of

a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit

modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library

facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License

may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

```
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
```

```
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
```

```
You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James
Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

A.8. GNU Libtool License

The following software may be included in this product:

GNU Libtool (The GNU Portable Library Tool)

```
If you are receiving a copy of the Oracle software in
source code, you are also receiving a copy of two files
(ltmain.sh and ltdl.h) generated by the GNU Libtool in
source code. If you received the Oracle software under
a license other than a commercial (non-GPL) license,
then the terms of the Oracle license do NOT apply to
these files from GNU Libtool; they are licensed under
the following licenses, separately from the Oracle
programs you receive.
```

```
Oracle elects to use GNU General Public License version
2 (GPL) for any software where a choice of GPL or GNU
Lesser/Library General Public License (LGPL) license
versions are made available with the language indicating
that GPL/LGPL or any later version may be used, or where
a choice of which version of the GPL/LGPL is applied is
unspecified.
```

From GNU Libtool:

```
ltmain.sh - Provide generalized library-building support
services.
```

```
NOTE: Changing this file will not affect anything until
you rerun configure.
```

```
Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005, 2006, 2007 Free Software Foundation, Inc.
Originally by Gordon Matzigkeit, 1996
```

```
This program is free software; you can redistribute it
and/or modify it under the terms of the GNU General
Public License as published by the Free Software Foundation;
either version 2 of the License, or (at your option) any
later version.
```

```
This program is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more
details. You should have received a copy of the GNU General
Public License along with this program; if not, write to
the Free Software Foundation, Inc., 51 Franklin Street,
Fifth Floor, Boston, MA 02110-1301, USA.
```

```
As a special exception to the GNU General Public License,
if you distribute this file as part of a program that
contains a configuration script generated by Autoconf,
you may include it under the same distribution terms that
you use for the rest of that program.
```

This component is licensed under [Section A.6, “GNU General Public License Version 2.0, June 1991”](#)

A.9. GNU Readline License

The following software may be included in this product:

GNU Readline Library

```
GNU Readline Library
With respect to MySQL Server/Cluster software licensed
under GNU General Public License, you are receiving a
copy of the GNU Readline Library in source code. The
terms of any Oracle license that might accompany the
Oracle programs do NOT apply to the GNU Readline Library;
it is licensed under the following license, separately
from the Oracle programs you receive. Oracle elects to
use GNU General Public License version 2 (GPL) for any
software where a choice of GPL license versions are
made available with the language indicating that GPLv2
or any later version may be used, or where a choice of
which version of the GPL is applied is unspecified.
```

This component is licensed under [Section A.6, “GNU General Public License Version 2.0, June 1991”](#)

A.10. Google Controlling Master Thread I/O Rate Patch License

The following software may be included in this product:

Google Controlling master thread I/O rate patch

```
Copyright (c) 2009, Google Inc.
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
```

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software

```
without specific prior written permission.
```

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

A.11. Google Perftools (TCMalloc utility) License

The following software may be included in this product:

```
Google Perftools (TCMalloc utility)
```

```
Copyright (c) 1998-2006, Google Inc.
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the following
conditions are met:
```

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

A.12. Google SMP Patch License

The following software may be included in this product:

Google SMP Patch

```
Google SMP patch
```

```
Copyright (c) 2008, Google Inc.
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
```

are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.13. lib_sql.cc License

The following software may be included in this product:

lib_sql.cc

Copyright (c) 2000
SWsoft company

This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk.

Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

This code was modified by the MySQL team.

A.14. Libaio License

The following software may be included in this product:

libaio

This component is licensed under [Section A.7, "GNU Lesser General Public License Version 2.1, February 1999"](#).

A.15. libevent License

The following software may be included in this product:

libevent

Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

==

Parts developed by Adam Langley

==

==

log.c

Based on err.c, which was adapted from OpenBSD libc *err*warncode.

Copyright (c) 2005 Nick Mathewson

Copyright (c) 2000 Dug Song

Copyright (c) 1993 The Regents of the University of California.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

==

min_heap.h

Copyright (c) 2006 Maxim Yegorushkin

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

==

win32.c

Copyright 2000-2002 Niels Provos
Copyright 2003 Michael A. Davis
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

A.16. Linux-PAM License

The following software may be included in this product:

Linux-PAM (pam-devel, Pluggable authentication modules for Linux)

Copyright Theodore Ts'o, 1996. All rights reserved.

```
(For the avoidance of doubt, Oracle uses and distributes this component under the terms below and elects not to do so under the GPL even though the GPL is referenced as an option below.)
```

```
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

```
ALTERNATIVELY, this product may be distributed under the terms of the GNU Public License, in which case the provisions of the GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential bad interaction between the GPL and the restrictions contained in a BSD-style copyright.)
```

```
THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

A.17. md5 (Message-Digest Algorithm 5) License

The following software may be included in this product:

```
md5 (Message-Digest Algorithm 5)
```

```
This code implements the MD5 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.
```

```
Equivalent code is available from RSA Data Security, Inc.
This code has been tested against that, and is equivalent,
except that you don't need to include two pages of legalese
with every copy.
```

```
The code has been modified by Mikael Ronstroem to handle
calculating a hash value of a key that is always a multiple
of 4 bytes long. Word 0 of the calculated 4-word hash value
is returned as the hash value.
```

A.18. nt_servc (Windows NT Service class library) License

The following software may be included in this product:

```
nt_servc (Windows NT Service class library)
```

```
Windows NT Service class library
Copyright Abandoned 1998 Irena Pancirov - Irnet Snc
This file is public domain and comes with NO WARRANTY of any kind
```

A.19. OpenPAM License

The following software may be included in this product:

OpenPAM

```
Copyright (c) 2002-2003 Networks Associates Technology, Inc.
Copyright (c) 2004-2007 Dag-Erling Smørgrav
All rights reserved.
```

```
This software was developed for the FreeBSD Project by
ThinkSec AS and Network Associates Laboratories, the
Security Research Division of Network Associates, Inc.
under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"),
as part of the DARPA CHATS research program.
```

```
Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

A.20. Percona Multiple I/O Threads Patch License

The following software may be included in this product:

Percona Multiple I/O threads patch

```
Copyright (c) 2008, 2009 Percona Inc
All rights reserved.
```

```
Redistribution and use of this software in source and binary forms,
with or without modification, are permitted provided that the
following conditions are met:
```

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

```
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of Percona Inc. nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission of Percona Inc.
```

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

A.21. Red HAT RPM Spec File License

The following software may be included in this product:

Red Hat RPM Spec File

```
You are receiving a copy of the Red Hat spec file. The terms of the Oracle
license do NOT apply to the Red Hat spec file; it is licensed under the
following license, separately from the Oracle programs you receive.
```

```
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
```

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
[for rest of text, see following link]
```

This component is licensed under [Section A.6, "GNU General Public License Version 2.0, June 1991"](#)

A.22. RegEX-Spencer Library License

The following software may be included in this product: Henry Spencer's Regular-Expression Library (RegEX-Spencer)

```
Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone
and Telegraph Company or of the Regents of the University of California.
```

```
Permission is granted to anyone to use this software for any purpose on
any computer system, and to alter it and redistribute it, subject
to the following restrictions:
```

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

A.23. RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License

The following software may be included in this product:

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

A.24. Richard A. O'Keefe String Library License

The following software may be included in this product:

Richard A. O'Keefe String Library

The Richard O'Keefe String Library is subject to the following notice:

These files are in the public domain. This includes getopt.c, which is the work of Henry Spencer, University of Toronto Zoology, who says of it "None of this software is derived from Bell software. I had no access to the source for Bell's versions at the time I wrote it. This software is hereby explicitly placed in the public domain. It may be used for any purpose on any machine by anyone." I would greatly prefer it if *my* material received no military use.

The t_ctype.h file is subject to the following notice:

Copyright (C) 1998, 1999 by Pruet Boonma, all rights reserved.

Copyright (C) 1998 by Theppitak Karoonboonyanan, all rights reserved.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above

```
copyright notice appear in all copies.
```

```
Smaphan Raruenrom and Pruet Boonma makes no representations about  
the suitability of this software for any purpose. It is provided  
"as is" without express or implied warranty.
```

A.25. SHA-1 in C License

The following software may be included in this product:

SHA-1 in C

```
SHA-1 in C  
By Steve Reid <steve@edmweb.com>  
100% Public Domain
```

A.26. [zlib](#) License

The following software may be included in this product:

zlib

Oracle gratefully acknowledges the contributions of Jean-loup Gailly and Mark Adler in creating the zlib general purpose compression library which is used in this product.

```
zlib.h -- interface of the 'zlib' general purpose compression library  
Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler
```

```
zlib.h -- interface of the 'zlib' general purpose compression library  
version 1.2.3, July 18th, 2005  
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
```

```
zlib.h -- interface of the 'zlib' general purpose compression library  
version 1.2.5, April 19th, 2010  
Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler
```

```
This software is provided 'as-is', without any express or implied warranty.  
In no event will the authors be held liable for any damages arising from the  
use of this software. Permission is granted to anyone to use this software  
for any purpose, including commercial applications, and to alter it and  
redistribute it freely, subject to the following restrictions:
```

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

```
Jean-loup Gailly jloup@gzip.org  
Mark Adler madler@alumni.caltech.edu
```