
ZMySQLDA Documentation

Release 5.0

Zope Foundation and Contributors

Feb 02, 2023

CONTENTS

1	Compatibility	3
2	Narrative documentation	5
2.1	Connection Strings	5
2.2	Usage from the Zope ZMI	6
2.3	Development	7
2.4	Contributors	9
3	API documentation	11
3.1	Products.ZMySQLDA.DA.Connection	11
4	Indices and tables	15
	Python Module Index	17
	Index	19

This is the MySQL database adapter product for Zope.

COMPATIBILITY

- Zope 5 under Python 3.7 and higher.
- MySQL versions 3.22 and later including MariaDB. You need version 4.1 or higher for unicode support.

NARRATIVE DOCUMENTATION

2.1 Connection Strings

The connection string used for Z MySQL Database Connection objects are of the form:

```
[*lock_name][+|-]database[@host[:port]] [user [password [unix_socket]]]
```

or typically just:

```
database user password
```

to use a MySQL server on the local host via the standard UNIX socket.

The components are as follows:

- ***lock_name** at the beginning of the connection string leads to pseudo-transactional behavior. When the Zope transaction begins, a lock named after **lock_name** is acquired on the database server. When the Zope transaction commits, the database lock will be released. If the Zope transaction is aborted and restarted, which can happen due to **ConflictErrors**, you'll get an error in the logs, and inconsistent data. In this respect, it's equivalent to transactions turned off.

Transactions are highly recommended. Using a named lock in conjunctions with transactions is probably pointless.

- **+ or -**: Integrate database transactions with the Zope transaction machinery. A **-** in front of the database tells ZMySQLDA to not use Zope's Transaction Manager, even if the server supports transactions. A **+** in front of the database tells ZMySQLDA that it must use transactions; an exception will be raised if they are not supported by the server. If neither **-** or **+** are present, then transactions will be enabled if the server supports them. If you are using non-transaction safe tables (TSTs) on a server that supports TSTs, use **-**. If you require transactions, use **+**. If you aren't sure, don't use either.
- **database**: The name of the database to connect to.
- **host/port**: Host and port where the database server listens. Only use this if the database server is on a remote system. To use a non-standard port on the local system, use 127.0.0.1 for the host instead of the hostname localhost.
- **user/password**: Log into the database with the provided user and password.
- **unix_socket**: If the UNIX socket is in a non-standard location, you can specify the full path to it after the password.

2.2 Usage from the Zope ZMI

The database connection object can be manipulated in the Zope ZMI on a series of screens, accessible through named tabs in the main window.

2.2.1 Status

Shows the database connection status and allows the user to open or close the connection.

2.2.2 Properties

Edit the database connection attributes and apply any changes:

- *Title*: An optional title that shows up in the ZMI.
- *Database Connection String*: A string encapsulating how to connect to the database. See [Connection Strings](#) for details.
- *Connect immediately*: Should the database connection be established immediately or when the first database query is run.
- *Unicode Support*: If set to `True`, values from columns of type `CHAR`, `VARCHAR` and `TEXT` are returned as unicode strings by the database backend.
- *Character set*: Query results will be encoded in the character set specified here:
 - *Not set* always defaults to UTF-8.
- *Automatically create database*: If the *Database Connection String* refers to a database that does not yet exist *and* this setting is activated, the ZMySQLDA connector will attempt to create the database.

2.2.3 Test

The Test tab can be used as long as the database connection is connected. You can enter SQL statements into the text field and view the results sent back from the database.

2.2.4 Security

Change the Zope role to permission mappings here.

2.2.5 Undo

If your particular ZODB flavor supports it, you can undo Zope transactions affecting the database connector object here. These transactions don't reflect relational database transactions in the underlying MySQL or MariaDB databases, only ZODB transactions.

2.2.6 Ownership

Information about the Zope user who owns the database connector object. Ownership in the Zope sense confers additional rights.

2.2.7 Interfaces

View and change the Zope Interface assignments for the database connector object.

2.2.8 Browse

You can browse the database tables and columns from the relational database specified in the connection string.

2.3 Development

2.3.1 Getting the source code

The source code is maintained on GitHub. To check out the trunk:

```
$ git clone https://github.com/zopectoundation/Products.ZMySQLDA.git
```

You can also browse the code online at <https://github.com/zopectoundation/Products.ZMySQLDA>

2.3.2 Bug tracker

For bug reports, suggestions or questions please use the GitHub issue tracker at <https://github.com/zopectoundation/Products.ZMySQLDA/issues>.

2.3.3 Running the tests using `zc.buildout`

Products.ZMySQLDA ships with its own `buildout.cfg` file for setting up a development buildout:

```
$ cd Products.ZMySQLDA
$ python3 -m venv .
$ bin/pip install -U pip wheel
$ bin/pip install zc.buildout tox twine
$ bin/buildout
...
```

Once you have a buildout, the tests can be run as follows:

```
$ bin/test
Running tests at level 1
Running zope.testrunner.layer.UnitTests tests:
  Set up zope.testrunner.layer.UnitTests in 0.000 seconds.
  Running:
.....
  Ran 62 tests with 0 failures and 0 errors in 0.043 seconds.
```

(continues on next page)

(continued from previous page)

Tearing down left over layers:

Tear down zope.testrunner.layer.UnitTests in 0.000 seconds.

To run tests for all supported Python versions, code coverage and a PEP-8 coding style checker, you can use `tox` after completing the buildout step above:

```
$ bin/tox
GLOB sdist-make: ...
...
----- summary -----
py37: commands succeeded
py38: commands succeeded
py39: commands succeeded
py310: commands succeeded
py311: commands succeeded
lint: commands succeeded
coverage: commands succeeded
congratulations :)
```

2.3.4 Running the functional tests

Some tests are hard or even impossible to perform without a real running database backend. During a normal test run they will be skipped, and you will see output like this:

Total: 62 tests, 0 failures, 0 errors and 5 skipped in 0.090 seconds.

To run those functional tests you need to have a MySQL/MariaDB server running and listening on the standard unix socket, normally located at `/tmp/mysql.sock`. This database server must have a database named `zmysqldatest` that can be accessed by a user `zmysqldatest` with password `zmysqldatest`. To set this up, log into the running database server with an admin user and execute the following statements:

```
mysql> CREATE DATABASE IF NOT EXISTS zmysqldatest;
mysql> CREATE USER 'zmysqldatest'@'localhost' IDENTIFIED BY 'zmysqldatest';
mysql> GRANT ALL PRIVILEGES ON zmysqldatest.* TO 'zmysqldatest'@'localhost';
```

If everything worked you'll see test output like this:

Total: 62 tests, 0 failures, 0 errors and 0 skipped in 0.105 seconds.

If the functional tests are still skipped, uncomment the `print` call in the `_mySQLNotAvailable` function in the module `Products.ZMySQLDA.tests.base`. It will print any errors emitted by the database server.

2.3.5 Building the documentation using `zc.buildout`

The `Products.ZMySQLDA` buildout installs the Sphinx scripts required to build the documentation, including testing its code snippets:

```
$ cd docs
$ make html
...
build succeeded.
```

The HTML pages are in `_build/html`.

2.3.6 Making a release

These instructions assume that you have a development sandbox set up using `zc.buildout` as the scripts used here are generated by the buildout.

```
$ bin/buildout -N
$ bin/buildout setup setup.py sdist bdist_wheel
$ bin/twine upload -s dist/Products.ZMySQLDA-X.X.X*
```

The `bin/buildout` step will make sure the correct package information is used.

2.4 Contributors

The following list of people who have contributed code or documentation makes no claims about completeness.

- Andy Dustman
- John Eikenberry
- Vincent Pelletier
- Graeme Mathieson
- Federico Schwindt
- Brett Carter
- Mark Van den Borre
- Robert Buchholz
- Peter Seifert
- Martin Häcker
- Dr. Frank Hoffmann
- Jürgen Gmach
- Jens Vagelpohl
- Michael Howitz

API DOCUMENTATION

3.1 Products.ZMySQLDA.DA.Connection

The ZODB-based MySQL Database Connection object

```
class Products.ZMySQLDA.DA.Connection(id, title, connection_string, check, use_unicode=None,  
                                       charset=None, auto_create_db=None, timeout=None)
```

Bases: Connection

Zope database adapter for MySQL/MariaDB

```
__init__(id, title, connection_string, check, use_unicode=None, charset=None, auto_create_db=None,  
         timeout=None)
```

Instance setup. Optionally opens the connection.

String

id – The id of the ZMySQLDA Connection

String

title – The title of the ZMySQLDA Connection

String

connection_string – The connection string describes how to connect to the relational database.
See the documentation for details.

Bool

check – Check if the database connection can be opened after instantiation.

Bool

use_unicode – If set to True, values from columns of type CHAR, VARCHAR and TEXT are
returned as unicode strings by the database backend. Combined with the hardcoded utf8
character set of this package the setting allows you to control the character set of database
return values better. Default: False.

String

charset – The character set for the connection. MySQL/MariaDB will encode query results
to this character set.

Only utf8 will work. Default: utf8

Bool

auto_create_db – If the database given in connection_string does not exist, create it au-
tomatically. Default: False.

Int

timeout – The connect timeout for the connection in seconds. Default: None

connect(*conn_string*)

Base API. Opens connection to mysql. Raises if problems.

String

conn_string – The database connection string

factory()

Base API. Returns factory method for DB connections.

manage_edit(*title, connection_string, check=None, use_unicode=None, charset=None, auto_create_db=None, timeout=None, REQUEST=None*)

Edit the connection attributes through the Zope ZMI.

String

title – The title of the ZMySQLDA Connection

String

connection_string – The connection string describes how to connect to the relational database. See the documentation for details.

Bool

check – Check if the database connection can be opened after instantiation. Default: False.

Bool

use_unicode – Use unicode internally. Default: False.

String

charset – The character set for the connection. MySQL/MariaDB will encode query results to this character set.

Only utf8 will work. Default: utf8

Bool

auto_create_db – If the database given in *connection_string* does not exist, create it automatically. Default: False.

Int

timeout – The connect timeout for the connection in seconds. Default: None

Request

REQUEST – A Zope REQUEST object

sql_quote__(*sql_str, escapes={}*)

Base API. Used to massage SQL strings for use in queries.

String

sql_str – The raw SQL string to transform.

Dict

escapes – Additional escape transformations. Default: empty dict.

tpValues()

Support the DTML `tree` tag

Used in the Zope ZMI Browse tab

`Products.ZMySQLDA.DA.manage_addMySQLConnection`(*self, id, title, connection_string, check=None, use_unicode=None, auto_create_db=None, charset=None, timeout=None, REQUEST=None*)

Factory function to add a connection object from the Zope ZMI.

String

id – The id of the ZMySQLDA Connection

String

title – The title of the ZMySQLDA Connection

String

connection_string – The connection string describes how to connect to the relational database. See the documentation for details.

Bool

check – Check if the database connection can be opened after instantiation. Default: False.

Bool

use_unicode – If set to True, values from columns of type CHAR, VARCHAR and TEXT are returned as unicode strings by the database backend. Combined with the hardcoded `utf8` character set of this package the setting allows you to control the character set of database return values better. Default: False.

String

charset – The character set for the connection. MySQL/MariaDB will encode query results to this character set.

Only utf8 will work. Default: utf8

Bool

auto_create_db – If the database given in `connection_string` does not exist, create it automatically. Default: False.

Int

timeout – The connect timeout for the connection in seconds. Default: None

Object

REQUEST – The currently active Zope request object. Default: None.

INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)
- [glossary](#)

PYTHON MODULE INDEX

p

`Products.ZMySQLDA.DA`, [11](#)

Symbols

`__init__()` (*Products.ZMySQLDA.DA.Connection*
method), [11](#)

C

`connect()` (*Products.ZMySQLDA.DA.Connection*
method), [11](#)

`Connection` (*class in Products.ZMySQLDA.DA*), [11](#)

F

`factory()` (*Products.ZMySQLDA.DA.Connection*
method), [12](#)

M

`manage_addZMySQLConnection()` (*in module Prod-*
ucts.ZMySQLDA.DA), [12](#)

`manage_edit()` (*Products.ZMySQLDA.DA.Connection*
method), [12](#)

`module`

`Products.ZMySQLDA.DA`, [11](#)

P

`Products.ZMySQLDA.DA`
`module`, [11](#)

S

`sql_quote__()` (*Products.ZMySQLDA.DA.Connection*
method), [12](#)

T

`tpValues()` (*Products.ZMySQLDA.DA.Connection*
method), [12](#)