
TwinDB Table Compare Documentation

Release 3.0.3

Aleksandr Kuzminsky

Sep 13, 2020

Contents

1	TwinDB Table Compare	3
1.1	Credits	3
2	Installation	5
2.1	Supported Python versions	5
2.2	Stable release	5
2.3	From sources	5
3	Usage	7
4	twindb_table_compare	9
4.1	twindb_table_compare package	9
5	History	15
5.1	3.0.2 (2020-09-13)	15
5.2	0.1.0 (2016-08-29)	15
6	Contributing	17
6.1	Types of Contributions	17
6.2	Get Started!	18
6.3	Pull Request Guidelines	19
6.4	Tips	19
7	Credits	21
7.1	Development Lead	21
7.2	Contributors	21
8	Indices and tables	23
	Python Module Index	25
	Index	27

Contents:

CHAPTER 1

TwinDB Table Compare

TwinDB Table Compare reads `percona.``checksums``` from the master and slave and shows what records are difference if there are any inconsistencies.

- Free software: Apache Software License 2.0
- Documentation: <https://twindb-table-compare.readthedocs.io>.
- Blogpost with usage examples: <https://twindb.com/pt-table-checksum-show-differences/>

1.1 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Supported Python versions

- Python 3.6, 3.7, and 3.8 are supported.

2.2 Stable release

To install TwinDB Table Compare, run this command in your terminal:

```
$ pip install twindb-table-compare
```

This is the preferred method to install TwinDB Table Compare, as it will always install the most recent stable release. If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.3 From sources

The sources for TwinDB Table Compare can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone https://github.com/twindb/twindb-table-compare.git
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/twindb/twindb-table-compare/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

TwinDB Table Compare should be used in the command line.

Note: Before running TwinDB Table Compare tool, you should run `pt-table-checksum` first. `twindb-table-compare` uses `percona.checksums`. If you don't run `pt-table-checksum`, `percona.checksums` will be empty and thus `twindb-table-compare` won't show any differences.

Note: `twindb-table-compare` should run against the slave.

This will show differences in data between a *slave* and its master.

```
twindb-table-compare slave
```

where *slave* is a hostname of a MySQL slave.

```
# twindb-table-compare --user=dba --password=qwerty 192.168.35.251
2016-09-03 22:48:01,732: INFO: twindb_table_compare.get_inconsistencies():127:
↳Executing: SELECT chunk FROM `percona`.`checksums` WHERE (this_crc<>master_
↳crc OR this_cnt<>master_cnt) AND db='mysql' AND tbl='proxies_priv'
2016-09-03 22:48:01,734: INFO: twindb_table_compare.get_inconsistencies():138: Found
↳1 inconsistent chunk
2016-09-03 22:48:01,734: INFO: twindb_table_compare.get_inconsistencies():141: #
↳mysql.proxies_priv, chunk 1
2016-09-03 22:48:01,736: INFO: twindb_table_compare.get_inconsistencies():143: #
↳chunk index: None
2016-09-03 22:48:01,736: INFO: twindb_table_compare.get_inconsistencies():215:
↳Executing: SELECT * FROM `mysql`.`proxies_priv` WHERE 1
2016-09-03 22:48:01,743: INFO: twindb_table_compare.get_inconsistencies():257:
↳Differences between slave 192.168.35.251 and its master:
--- /tmp/master.GZ8S7V 2016-09-03 22:48:01.737762174 +0000
+++ /tmp/slave.9t4HhV 2016-09-03 22:48:01.738761674 +0000
@@ -1,2 +1,2 @@
```

(continues on next page)

(continued from previous page)

[illegible]

Run `twindb-table-compare --help` for other options.

4.1 twindb_table_compare package

4.1.1 Submodules

4.1.2 twindb_table_compare.cli module

Command line routines

4.1.3 twindb_table_compare.compare module

Functions to find and print differences

`twindb_table_compare.compare.build_chunk_query(db, tbl, chunk, conn, ch_db='percona', ch_tbl='checksums')`

For a given database, table and chunk number construct a SELECT query that would return records in this chunk.

`twindb_table_compare.compare.diff(master_lines, slave_lines, color=True)`

Find differences between two set of lines.

Parameters

- **master_lines** (*list*) – First set of lines
- **slave_lines** (*list*) – Second set of lines
- **color** (*bool*) – If True return colored diff

Returns Difference between two set of lines

Return type str

`twindb_table_compare.compare.get_boundary(*args, **kwargs)`

Get lower and upper boundary values of a chunk.

Parameters

- **args** (*tuple(Connection, str, str, int)*) – Positional arguments.
- **kwargs** (*dict*) – Keyword arguments.

Returns tuple with values lower_boundary and upper_boundary of percona.checksums

Positional arguments

- **connection**: MySQL connection.
- **database**: database of the chunk.
- **tbl**: table of the chunk.
- **chunk**: chunk id.

Keyword arguments.

- **ch_db**: Database where checksums are stored. Default percona.
- **ch_tbl**: Table where checksums are stored. Default checksums.

`twindb_table_compare.compare.get_boundary_clause(oper='>', index_fields=None,
boundaries=None)`

Generate a clause for the WHERE statement based on field names and values.

Parameters

- **oper** – Can be either < for the lower boundary or > for the upper boundary.
- **index_fields** (*list*) – list of fields in the index.
- **boundaries** (*list*) – list of values for a boundary.

Returns a clause that defines a boundary (upper or lower) of a chunk.

`twindb_table_compare.compare.get_chunk_index(*args, **kwargs)`

Get index that was used to cut the chunk.

Parameters

- **args** (*tuple(Connection, str, str, int)*) – Positional arguments.
- **kwargs** (*dict*) – Keyword arguments.

Returns index name or None if no index was used

Positional arguments

- **connection**: MySQL connection.
- **database**: database of the chunk.
- **tbl**: table of the chunk.
- **chunk**: chunk id.

Keyword arguments.

- **ch_db**: Database where checksums are stored. Default percona.
- **ch_tbl**: Table where checksums are stored. Default checksums.

`twindb_table_compare.compare.get_fields(conn, db, tbl)`

Construct fields list string for a SELECT. If a field is a binary type (BLOB, VARBINARY) then HEX() it.

Parameters

- **conn** (*Connection*) – MySQL connection.
- **db** (*str*) – Database name.
- **tbl** (*str*) – Table name.

Returns A comma separated list of fields.

Return type `str`

`twindb_table_compare.compare.get_inconsistencies(db, tbl, slave, user, passwd, ch_db='percona', ch_tbl='checksums', vertical=False, color=True)`

Print differences between slave and its master.

Parameters

- **db** – Database name of the inconsistent table.
- **tbl** – Table name of the inconsistent table.
- **slave** – Hostname of the slave.
- **user** – User to connect to MySQL.
- **passwd** – Password to connect to MySQL.
- **ch_db** – Database where checksums are stored.
- **ch_tbl** – Table name where checksums are stored.
- **vertical** – If True - print result vertically (G in MySQL)
- **color** – If True - print colorful output

`twindb_table_compare.compare.get_inconsistent_tables(host, user, password, ch_db='percona', ch_tbl='checksums')`

On a given MySQL server find tables that are inconsistent with the master.

Parameters

- **host** – Hostname with potentially inconsistent tables.
- **user** – MySQL user.
- **password** – MySQL password.
- **ch_db** – Database where checksums are stored.
- **ch_tbl** – Table name where checksums are stored.

Returns List of tuples with inconsistent tables. Each tuple is database name, table name.

Return type `list`

`twindb_table_compare.compare.get_index_fields(connection, db, tbl, index)`

Get fields of the given index

Parameters

- **connection** – MySQLdb connection
- **db** – database
- **tbl** – table
- **index** – index name

Returns list of field names

`twindb_table_compare.compare.get_master(connection)`

Get master host

Parameters **connection** – MySQL connection

Returns Master hostname

`twindb_table_compare.compare.get_where(lower_boundary, upper_boundary, index_fields)`

Generate WHERE clause based on strings `lower_boundary`, `upper_boundary` from `percona.checksums` table and fields in the index.

Parameters

- **lower_boundary** (*str*) – values of lower boundary.
- **upper_boundary** (*str*) – values of upper boundary.
- **index_fields** – list of fields in the index that was used. to access a chunk.

Returns a WHERE clause to read a chunk.

Return type `str`

`twindb_table_compare.compare.is_printable(str_value)`

Checks if `str_value` is printable string.

Parameters **str_value** –

Returns True if `str_value` is printable. False otherwise

`twindb_table_compare.compare.primary_exists(conn, db, tbl)`

Check if PRIMARY index exists in table `db.tbl`

Parameters

- **conn** (*Connection*) – MySQLdb connection.
- **db** (*str*) – Database name.
- **tbl** (*str*) – Table name.

Returns True if index PRIMARY exists in table `db.tbl`

Return type `bool`

`twindb_table_compare.compare.print_horizontal(cur_master, cur_slave, query, color=True)`

Find and return differences in horizontal format i.e. one line - one record

Parameters

- **cur_master** (*Cursor*) – MySQLdb cursor on master
- **cur_slave** (*Cursor*) – MySQLdb cursor on slave

- **query** (*str*) – Query to find records in a chunk we compare
- **color** – If True - produce colorful output

Returns Differences in a chunk between master and slave

Return type *str*

`twindb_table_compare.compare.print_vertical(master, slave, user, passwd, query, color=True)`

Find and return differences in vertical format. The vertical format is when you end MySQL query with 'G'

Parameters

- **master** (*str*) – Hostname of the master.
- **slave** (*str*) – Hostname of the slave.
- **query** (*str*) – Query to find records in a chunk we compare
- **color** – If True - produce colorful output

Returns Differences in a chunk between master and slave

Return type *str*

4.1.4 Module contents

Module to read pr-table-checksum's result table (percona.checksums) and show user which records are actually different.

`twindb_table_compare.setup_logging(logger, debug=False, color=True)`

Configure logging.

Parameters

- **logger** (*Logger*) – Logger to configure.
- **debug** – If True - print debug messages
- **color** – If True - print colored messages

5.1 3.0.2 (2020-09-13)

- Python 3 support
- Command line tool renamed to twindb-table-compare

5.2 0.1.0 (2016-08-29)

- First release on PyPI.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/twindb/twindb_table_compare/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

6.1.4 Write Documentation

TwinDB Table Compare could always use more documentation, whether as part of the official TwinDB Table Compare docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/twindb/twindb_table_compare/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *twindb_table_compare* for local development.

1. Fork the *twindb_table_compare* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/twindb_table_compare.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv twindb_table_compare
$ cd twindb_table_compare/
$ make bootstrap
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass linter and the tests, including testing other Python versions with `tox`:

```
$ make lint
$ make test-all
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and for PyPy. Check https://travis-ci.org/twindb/twindb_table_compare/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ py.test tests/unit/test_twindb_table_compare.py
$ py.test tests/unit/test_twindb_table_compare.py::test_build_chunk_query
```


CHAPTER 7

Credits

7.1 Development Lead

- Aleksandr Kuzminsky <aleks@twindb.com>

7.2 Contributors

None yet. Why not be the first?

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

t

`twindb_table_compare`, [13](#)

`twindb_table_compare.cli`, [9](#)

`twindb_table_compare.compare`, [9](#)

B

`build_chunk_query()` (in *module twindb_table_compare.compare*), 9

D

`diff()` (in *module twindb_table_compare.compare*), 9

G

`get_boundary()` (in *module twindb_table_compare.compare*), 9

`get_boundary_clause()` (in *module twindb_table_compare.compare*), 10

`get_chunk_index()` (in *module twindb_table_compare.compare*), 10

`get_fields()` (in *module twindb_table_compare.compare*), 11

`get_inconsistencies()` (in *module twindb_table_compare.compare*), 11

`get_inconsistent_tables()` (in *module twindb_table_compare.compare*), 11

`get_index_fields()` (in *module twindb_table_compare.compare*), 11

`get_master()` (in *module twindb_table_compare.compare*), 12

`get_where()` (in *module twindb_table_compare.compare*), 12

I

`is_printable()` (in *module twindb_table_compare.compare*), 12

P

`primary_exists()` (in *module twindb_table_compare.compare*), 12

`print_horizontal()` (in *module twindb_table_compare.compare*), 12

`print_vertical()` (in *module twindb_table_compare.compare*), 13

S

`setup_logging()` (in *module twindb_table_compare*), 13

T

`twindb_table_compare` (module), 13

`twindb_table_compare.cli` (module), 9

`twindb_table_compare.compare` (module), 9