

---

# Regtest Documentation

*Release 0.3.3 [4 - Beta]*

**sonntagsgesicht**

**Friday, 05 May 2023**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Tutorial</b>	<b>3</b>
<b>3</b>	<b>Releases</b>	<b>5</b>
3.1	Release 0.3.3 . . . . .	5
3.2	Release 0.3.2 . . . . .	5
3.3	Release 0.3.1 . . . . .	5
3.4	Release 0.3 . . . . .	5
3.5	Release 0.2 . . . . .	5
3.6	Release 0.1 . . . . .	6
<b>4</b>	<b>Indices and tables</b>	<b>7</b>



# CHAPTER 1

---

## Introduction

---

Regression test enhancement for the Python *unittest* framework

Writing tests is important (see [here](#)). And when it comes to an existing and running application even more. Existing results must at any chance be reproduced (*like-for-like* tests).

An easy way to add many test cases is by invoking the application and its subroutines many times. But taking notes (and hardcoding) of all results is annoying.

Here **regtest** might help.

Simply, write routines that invoke our application. The initial run will collect and store return values in files. The next time (and at any time these routines run) the return values will be checked against the stored ones.

To reset a routine simply remove the corresponding file (named accordingly) of stored reference data. The next time the reference data will be rebuild.



## CHAPTER 2

---

### Tutorial

---

Start with writing test cases for our program

```
def foo(x):  
    return x * x
```

in a file (let's call it `reg_test.py`)

```
class RegressiveTest(RegressionTestCase):  
    """our regression test case"""  
  
    def test_almost_regressive_equal(self):  
        self.assertAlmostRegressiveEqual(foo(1.01))  
        self.assertAlmostRegressiveEqual(foo(1.11))  
  
    def test_regressive_equal(self):  
        self.assertAlmostRegressiveEqual(foo.__name__)  
        self.assertAlmostRegressiveEqual(foo(2) > 0)
```

At first test run

```
$ python -m unittest reg_test.py
```

the return values are stored in files (more precise the argument values of `assertRegressiveEqual` and `assertAlmostRegressiveEqual`):

```
test/data/RegressiveTest/test_regressive_equal.json.zip  
test/data/RegressiveTest/test_almost_regressive_equal.json.zip
```

Re-running

```
$ python -m unittest reg_test.py
```

will now use those data. If any values have changed `AssertError` will be raised as usual.

If the testcase may have changed (less or resp. more calls of `assertRegressiveEqual` and `assertAlmostRegressiveEqual`) some reference data will be left over or resp. missing.

So a `LeftoverAssertValueError` or resp. `MissingAssertValueError` will be raised.

Note: All file input/output is done by the `setUp()` and `tearDown()` methods of the standard `unittest` framework. So on overwrite, don't forget to call either `super` or `enhance`

```
def setUp(self):
    self.readResults()

def tearDown(self):
    self.validateResults()
    self.writeResults()
```

Hint: To avoid compression by zip archives set the class property `compression` of the `RegressionTestCase` class to `False`

```
RegressiveTest.compression=False
```

Hence

```
test/data/RegressiveTest/test_regressive_equal.json
test/data/RegressiveTest/test_almost_regressive_equal.json
```



These changes are listed in decreasing version number order.

### 3.1 Release 0.3.3

Release date was Friday, 05 May 2023

- fixing *tuple* as *list* issue for nested tuples and list.

### 3.2 Release 0.3.2

Release date was Friday, 05 May 2023

- fixing *tuple* as *list* issue for nested tuples.

### 3.3 Release 0.3.1

Release date was Friday, 05 May 2023

- fixing *tuple* as *list* issue by `loads(dumps((1, 2)))==[1, 2]`. Now every *tuple* or *set* is asserted as a *list*.

### 3.4 Release 0.3

Release date was Sunday, 21 November 2021

- control compression by class property

### 3.5 Release 0.2

Release date was Thursday, 7 October 2021

- dropping python 2 support

- store regression data in one file per each test method (than each class)
- made it a [auxilium](#) project
- a bit more documentation

### 3.6 Release 0.1

Release date was Wednesday, 18 September 2019

## CHAPTER 4

---

### Indices and tables

---

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