# TYPO3.Flow - Bug #37473

## Subsequent Exceptions related to Doctrine Entity Manager makes it snap shut

2012-05-24 16:22 - Adrian Föder

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 2012-05-24 |
| **Priority:** | Must have | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | - Testing - | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **PHP Version:** | | **Complexity:** | |
| **Has patch:** | No | | |

**Description**

If any exception occurs while persisting, and a subsequent test also tries to persist, an ORM Exception occurs telling that the Entity manager were closed.
Reproduction code in Functional Test:

```
    /**
     * Fine, this throws, as expected, an Exception due to validation problems
     *
     * @test
     * @expectedException \TYPO3\FLOW3\Persistence\Exception\ObjectValidationFailedException
     */
    public function firstFailingTestThatCorrectlyThrowsException() {
        $invalidAccount = new \TYPO3\FLOW3\Security\Account();
        $this->persistenceManager->add($invalidAccount);
        $this->persistenceManager->persistAll();
    }

    /**
     * This (same code) throws now "Doctrine\ORM\ORMException: The EntityManager is closed."
     *
     * @test
     */
    public function secondFailingTestThatSaysEntityManagerWasClosed() {
        $invalidAccount = new \TYPO3\FLOW3\Security\Account();
        $this->persistenceManager->add($invalidAccount);
        $this->persistenceManager->persistAll();
    }
```

Maybe this is some kind related:
http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/transactions-and-concurrency.html#exception-handling

### History

**#1 - 2012-10-25 10:29 - Matthieu Napoli**

Apparently this is the intended behavior of Doctrine:

> In other words, when an exception occurs during flush/commit it is too error-prone to try to get the UoW back into a reusable state. Such an exception is unrecoverable for the EM/UoW, hence why it is closed and should not be reused.
>
> [...]
>
> You should not reuse an EM in tests like that as it causes too many potential side-effects between tests, which should be avoided. The Doctrine2 tests themselves get a fresh EM for every test method that needs one. What is reused/shared is the database connection in the functional tests. Creating a new EM is not a costly operation. A newly created EM can use the same DB connection (and Configuration and EventManager) as a previously failed one.
>
> [...]
>
> Your code can recover from the exception, if you so desire, but the EM/UoW in which this exception occurred can not recover.
> Thus, if **you** can recover and you need a new unit of work, you should create one (by creating a new EM).

from https://groups.google.com/forum/?fromgroups=#!topic/doctrine-user/cE7h5Mud_hc