

## TYPO3 Core - Bug #68849

### Unstable persistence handling of DateTime (don't get what you set)

2015-08-07 18:06 - Sebastian Mazza

<b>Status:</b>	Closed	<b>Start date:</b>	2015-08-07
<b>Priority:</b>	Must have	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Extbase	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Complexity:</b>	
<b>TYPO3 Version:</b>	7	<b>Is Regression:</b>	No
<b>PHP Version:</b>	5.6	<b>Sprint Focus:</b>	
<b>Tags:</b>			

#### Description

DATE or DATETIME database values are treated as UTC during the mapping of an object and the php default time zone (date\_default\_timezone\_get()) is applied to it. E.g.: If your php default time zone is "Europe/Berlin" (timezone\_type: 3), it is summer and you have stored 2015-08-07 13:00:00 in the Database you will get a DateTime object with the date 2015-08-07 15:00:00.

This is only done if your dates are stored in a column of type DATE or DATETIME. If you store the dates in a column of type INTEGER as unix timestamp the date values are not converted. Therefore, changing the column type of the database changes the values you get from a repository. This, however, cannot be correct.

Because the TCE in the Backend does not convert any dates/time zones, you will see different times in FE and BE.

But all of that is not the biggest problem. What is really annoying is the fact that the DateTime objects are not converted back to UTC during saving to DB (persistAll()). As a result, every time you read and write a DateTime object your Time is increased/ decreased by the time offset of your default php time zone (in my case 2 hours).

#### Example

DB value for start: 2015-08-07 13:00:00

DB value after execution the following piece of code: 2015-08-07 15:01:00

```
public function dateTestAction() {
    $obj = $this->operatingTimeRepository->findByUid(1);
    $obj->getStart()->add(new \DateInterval('P0Y0DT0H1M')); // add one minute
    $this->operatingTimeRepository->update($obj);
}
```

#### Involved code:

typo3/sysext/extbase/Classes/Persistence/Generic/Mapper/DataMapper.php  
save to db: getPlainValue()  
read from db: mapDateTime()

#### possible solution:

Remove the conversion during read from Database and let the extension programmer decide how to deal with time zones. See attached patch.

#### related:

[\[\[https://forge.typo3.org/issues/68651#note-1\]\]](https://forge.typo3.org/issues/68651#note-1)

#### Related issues:

Related to TYPO3 Core - Bug #64953: dbType date and datetime fields are saved...	Closed	2015-02-10
Related to TYPO3 Core - Bug #72878: wrong datetime handling, they are not UTC...	Closed	2016-01-21
Related to TYPO3 Core - Bug #68651: Datetime() properties have wrong timezone	Accepted	2015-07-30
Is duplicate of TYPO3 Core - Bug #74376: Extbase persistence not working with...	Closed	2016-03-06

---

## History

---

### #1 - 2016-04-24 18:09 - Nicole Cordes

- Status changed from New to Needs Feedback

Hi Sebastian,

Thank you for your bug report.

Your problem should be solved with <https://review.typo3.org/#/c/47136/>

Can you please test this patch, which was also backported to 7.6 <https://review.typo3.org/#/c/47846/> and give feedback if this solves your problem as well.

### #2 - 2016-06-27 19:33 - Sebastian Mazza

Hi Nicole,

thank you for this patch, it solves my Date/Time update problem with the Exbase DataMapper.

But, while testing, I found an other, in my opinion inconsistent behaviour with the backend.

If you store Date/Time values in the DB as UNIX Timestamp (int). The values, displayed in backend forms are increased by the time offset of your local Timezone. But if you store the Date/Time values in table columns of type datetime, the backend form displays the exact UTC time that are stored in the DB.

In other words: If you store your Date/Time values as UNIX Timestamps the frontend output is exact the same as the backend form output. But if you use datetime columns instead, your frontend output is increased by the time offset of your local time zone. (e.g. submit 27.06.2016 10:00 in backend, leads to an output of 27.06.2016 12:00 in the frontend)

### #3 - 2016-09-27 10:56 - Riccardo De Contardi

- Status changed from Needs Feedback to New

### #4 - 2017-04-15 01:15 - Mona Muzaffar

- Related to Epic #80852: Datetime handling in backend added

### #5 - 2017-06-06 18:19 - Markus Klein

- Parent task set to #77562

### #6 - 2017-06-06 19:12 - Markus Klein

- Related to Bug #68651: Datetime() properties have wrong timezone added

### #7 - 2017-06-06 19:14 - Markus Klein

- Is duplicate of Bug #74376: Exbase persistence not working with datetime fields added

### #8 - 2017-06-06 19:14 - Markus Klein

Closing as duplicate of [#74376](#).

The other mentioned issue is dealt with in [#77562](#)

### #9 - 2017-06-06 19:14 - Markus Klein

- Status changed from New to Closed

### #10 - 2017-06-06 19:25 - Markus Klein

- Parent task deleted (#77562)

### #11 - 2017-06-07 11:02 - Markus Klein

- Related to deleted (Epic #80852: Datetime handling in backend)

## Files

---

patch.diff

1.31 KB

2015-08-07

Sebastian Mazza