TYPO3 Core - Bug #79613

Epic # 77562 (Accepted): Misbehaviors with datetime values and timezones

Saving wrong Date into DB if Field is of type DATE

2017-02-03 11:33 - Sascha Rademacher

Status:	New	**Start date:**	2017-02-03
Priority:	Must have	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Extbase	**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	7	**Complexity:**	
PHP Version:	5.6	**Is Regression:**	No
Tags:		**Sprint Focus:**	

Description

According to this Bugfix:
https://forge.typo3.org/issues/72878

I can Confirm the Problem from Viktor Livakivskyis comment:

> There is a special case with DB fields of type 'date'.
>
> Imagine a case, when a DateTime object is created in "Europe/Berlin" like this
>
> $dateObject = new \DateTime('2016-07-13');
>
> This will correspond to 2016-07-13 00:00:00.000000 in timezone "Europe/Berlin".
> So, in case ExtBase will convert it to UTC during persistence, the date will correspond to 2016-07-12 22:00:00.000000 in UTC timezone. And saved to DB will result in "2016-07-12" stored.
>
> Then on retrieval from DB the DateTime object will be created and Timezone shift added, which will correspond to 2016-07-12 02:00:00.000000 in "Europe/Berlin".
> One day is lost then.
>
> So, for case with native DB 'date' field the timezone shift should be totally ignored.

Please respect suggestion from "This Mächler":

> Suggestion:
> Make it possible to choose how TYPO3 should treat dateTime values:
>
> 1. TYPO3 assumes that dateTime values from the DB are in local time and need no conversion from/to the FE or BE. This should be the default setting to avoid breaking a running system.
>
> 2. TYPO3 assumes that dateTime values are stored in UTC, conversion takes place from/to the FE or BE.
>
> This will not solve the problem with the already implemented workarounds. Maybe there is no solution for that.

Related issues:

Related to TYPO3 Core - Bug #79925: Native date/datetime (TCA dbType) fields ...	**Closed**	**2017-02-20**
Related to TYPO3 Core - Feature #80559: datetime and time handling should alw...	**Closed**	**2017-03-29**
Related to TYPO3 Core - Bug #80679: Input field eval datetime in FlexForm con...	**Closed**	**2017-04-04**
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**

History

#1 - 2017-02-03 16:50 - Susanne Moog

Hi,

to get the surrounding facts straight:

- You have a DB field that is not capable of storing time zone information (date)
- You create a date *with* timezone information
- You store that field and the timezone information is obviously lost
- When retrieving the data due to the lost timezone information the date is now wrong (which worked before by accident in case you created a date in the default timezone or anything "close enough")

The change to convert all dates to UTC is meant to ensure consistent date handling throughout the core - especially because date fields don't have timezone information - when displaying those dates later, we need "something" to assume as base. This could be:

- UTC always
- the php timezone
- the mysql / server timezone
- the timezone you set manually when creating the date
- the timezone on the client side (because he used a javascript datepicker)

only one of these options is in my eyes "safe" - and that is UTC always. To fix your issue I'd just use new \DateTime('2016-07-13', DateTimeZone::UTC); - or am I missing something?

For migration of "old" stored dates I'd directly use an mysql script doing something along
https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_convert-tz

#2 - 2017-02-14 15:23 - Sascha Rademacher

I have a date field in the database. No matter what I do, when I read it, I always get the date, which is in the database, no matter what time zone it was before, like always UTC.
A pure date is always independent of time zones for me (human logic, not PHP logic). So I still do not understand if I want to save a datetime object to a date field, why there has to be a conversion, the only thing that can happen during the conversion is that the date is broken. There is no advantage in any case.
With the datetime type it is not necessary to create it in UTC because TYPO3 converts it to default timezone when reading from database. This is confusing and inconsistent. I understand this in principle but I mean it is wrong.

And what about existing projects? I knew about the fact that TYPO3 stores dates time-zone independent or in other words, that TYPO3 does not convert dates before writing data to databse and therefore i wrote my code without converting anything / creating Dates in UTC.
Now I would have to adjust code in each project and unittests do no longer work anymore and I can not ensure that my changes work.

#3 - 2017-04-15 00:07 - Mona Muzaffar

- Related to Bug #80679: Input field eval datetime in FlexForm converts to wrong timestamp added

#4 - 2017-04-15 00:40 - Mona Muzaffar

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

#5 - 2017-06-06 18:15 - Markus Klein

- Parent task set to #81489

#6 - 2017-06-06 18:19 - Markus Klein

- Parent task changed from #81489 to #77562

#7 - 2017-06-06 19:21 - Markus Klein

- Related to Bug #72878: wrong datetime handling, they are not UTC in db added

#8 - 2017-06-06 19:24 - Markus Klein

- Category set to Extbase

#9 - 2017-06-06 19:24 - Markus Klein

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

#10 - 2017-06-07 13:03 - Riccardo De Contardi

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

#11 - 2017-06-08 00:38 - Markus Klein

- Status changed from New to Needs Feedback

Maybe somebody can rephrase this ticket to a shorter summary.

From what I understood by reading: Yes, if native date(time) datatype is used for date fields, extbase must take care when reading the value to convert it to a proper DateTime value that is representing the stored time in the default timezone.

#12 - 2017-06-12 07:38 - Sascha Rademacher

In summary: it should not be converted if the database field is a "date" (not datetime) field. The backconversion to the original date is not possible. The use of the conversion should be switchable in the installtool to ensure the compatibility to already existing projects which handle conversions by themselfes.

#13 - 2017-11-01 09:27 - Alexander Opitz

- Status changed from Needs Feedback to New