

## TYPO3.Flow - Feature # 37294

<b>Status:</b>	Closed	<b>Priority:</b>	Could have
<b>Author:</b>	Carsten Bleicker	<b>Category:</b>	Property
<b>Created:</b>	2012-05-19	<b>Assigned To:</b>	Carsten Bleicker
<b>Updated:</b>	2014-02-19	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	Property converters should allow NULL results		
<b>Description</b>			
Can someone explain me why the DateTimeConverter returns a Validation Error and does not throw an exception? (Line #122) All other Converters throwing exceptions only. shouldnt the validation only be a part of @FLOW3\Validate notations?			
<b>Related issues:</b>			
related to TYPO3.Flow - Bug # 49566: NULL source values are not handled corre...		<b>New</b>	<b>2013-07-02</b>
related to TYPO3.Flow - Bug # 36996: DateTimeConverter		<b>Closed</b>	<b>2012-05-09</b>

### History

#### #1 - 2012-05-19 11:52 - Carsten Bleicker

- File *DateTimeConverter.php* added

#### #2 - 2012-05-21 08:21 - Adrian Förder

Hi Carsten,

a good question; allow me to point to this commit:

<http://git.typo3.org/FLOW3/Packages/TYPO3.FLOW3.git?a=commit;h=b47d1c61aaaf67c6d9c3379d87fe132ff7bde80>

It might explain it very well.

If you have further questions don't hesitate to ask :)

#### #3 - 2012-05-21 09:35 - Carsten Bleicker

thanks adrian,

but thats exactly the point wich describes a unexcepted behaviour for me but its not only a unexcepted datetimeconverter behaviour.

lets say f.e. i want to have a start and a stop property. both could be null to definine a runtime of an object.

if the converter returns allways a valid datetime i am not able to set some properties to NULL.

in my opinion any converter should return NULL if converting fails or the source is an empty string.

f.e. also the float or integer converter. In this case no exception cant be thrown in any Converter because NULL

is possible. But i agree that exceptions SHOULD be thrown if the property isnt nullable.

to work with exceptions in case of not allowed NULL values i suggest to extend the behavior of subproperty mapping of

the objectConverter f.e. any property converting should be able to analyze its annotations. so the converter is able to the following:

if class annotated with @ORM\Column(nullable=true) the converter is able to convert to null and it does not throw an exception.

if class isnt annotated as nullable it throws exceptions.

what do you think?

in my opinion there is realy a need to allow null resulting converters.

kind regards  
carsten

**#4 - 2012-06-28 10:33 - Karsten Dambekalns**

- Status changed from New to Accepted
- Assigned To set to Karsten Dambekalns

**#5 - 2012-07-10 16:48 - Karsten Dambekalns**

- Tracker changed from Bug to Feature
- Subject changed from *DateTimeConverter* to *Property converters should allow NULL results*
- Status changed from Accepted to New
- Assigned To deleted (Karsten Dambekalns)

**#6 - 2013-07-02 08:30 - Adrian Förder**

- Status changed from New to Needs Feedback
- Assigned To set to Carsten Bleicker

Carsten,

I don't fully understand that now, because some TypeConverters already return NULL in the cases you've mentioned (Integer- and FloatConverters for example have

```
if ($source === NULL || strlen($source) === 0) {  
    return NULL;  
}
```

So it's not a design problem of the mapping framework, but of particular converters, is that correct? Could you name the converters where you think NULL should be returned silently?

**#7 - 2014-02-19 10:11 - Bastian Waidelich**

- Status changed from Needs Feedback to Closed

For the record: Adrian is correct, if the model supports NULL-values (nullable=true and allow NULL in the setters) this should work just fine. Closing due to missing feedback. Feel free to comment or re-open!

**Files**

---

DateTimeConverter.php	6.9 kB	2012-05-19	Carsten Bleicker
-----------------------	--------	------------	------------------