

TYPO3.Flow - Bug # 27088

Status:	On Hold	Priority:	Should have
Author:	Andreas Förthner	Category:	Object
Created:	2011-05-27	Assigned To:	
Updated:	2013-04-08	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: FLOW3 1.0.0			
Subject:	initializeObject() is called too early when reconstructing entities		
Description			
As it is called in __wakeup(), Doctrine did not yet inject the persisted properties, this is not what we want. The object should be fully initialized (DI and Persistence) when initializeObject() ist called. The same is true for objects fetched from persistence.			

History

#1 - 2011-06-17 09:34 - Karsten Dambekalns

- Assigned To set to Karsten Dambekalns

#2 - 2011-06-28 14:02 - Karsten Dambekalns

- Status changed from New to Needs Feedback

Well, looking at one of the proxies it seems the data should be loaded before the initializeObject() call:

```
/**
 * Fetches the identifier from the set content object. If that
 * is not using automatically introduced UUIDs by FLOW3 it tries
 * to call persistAll() and fetch the identifier again. If it still
 * fails, an exception is thrown.
 *
 * @return void
 */
public function initializeObject()
{
    $this->__load();
    return parent::initializeObject();
}
```

Can you provide sample code that shows the error?

#3 - 2011-06-28 14:04 - Karsten Dambekalns

The above is the Doctrine proxy, here is the FLOW3 proxy:

```
/**
 * Autogenerated Proxy Method
 */
```

```

public function __wakeup() {
    $this->FLOW3_AOP_Proxy_buildMethodsAndAdvicesArray();

    if (property_exists($this, 'FLOW3_Persistence_RelatedEntities') && is_array($this->FLOW3_Persistence_RelatedEntities)) {
        $persistenceManager = \F3\FLOW3\Core\Bootstrap::$staticObjectManager->get('F3\FLOW3\Persistence\PersistenceManagerInterface');
        foreach ($this->FLOW3_Persistence_RelatedEntities as $entityInformation) {
            $this->$entityInformation['propertyName'] = $persistenceManager->getObjectByIdentifier($entityInformation['identifier'],
            $entityInformation['entityType']);
        }
        unset($this->FLOW3_Persistence_RelatedEntities);
    }

    $this->FLOW3_Proxy_injectProperties();
    $result = NULL;
    if (is_callable('parent::__wakeup')) parent::__wakeup();

    $this->initializeObject(2);
    return $result;
}

```

#4 - 2011-08-04 08:18 - Sebastian Kurfuerst

- Priority changed from Must have to Should have

@Andi: can you provide feedback here? Else, I'll defer it to beta2...

#5 - 2011-08-04 10:54 - Lienhart Woitok

At least one case where this happens is when reconstituting objects from db. Not with lazy loading, but completely create them at once. Doctrine creates the class by unserializing an empty object of the entity class and then fills it with the properties. But unserialize does call __wakeup() and therefore initializeObject(). I think this changed at some point in Doctrine, when Andi reported this I believe Doctrine unserialized every new instance, but in current version I find that the unserialized instance is cached in Doctrine (which might lead to even more unpredictable behavior in this context). Hope this clears it up a bit.

Currently, I work around this by defining initializeObject() as a PostLoad lifecycle callback and check that everything I need is there before doing anything.

#6 - 2011-08-24 10:24 - Robert Lemke

- Target version changed from 1.0 beta 1 to 1.0 beta 2

#7 - 2011-09-09 15:17 - Robert Lemke

- Target version changed from 1.0 beta 2 to 1.0.0

#8 - 2011-09-30 10:19 - Karsten Dambekalns

- Assigned To deleted (Karsten Dambekalns)

#9 - 2011-10-20 01:46 - Karsten Dambekalns

- Target version changed from 1.0.0 to 1.0.1

#10 - 2011-10-21 13:20 - Karsten Dambekalns

- Affected Flow version set to FLOW3 1.0.0

#11 - 2011-11-22 12:48 - Karsten Dambekalns

- Target version changed from 1.0.1 to 1.0.2

#12 - 2011-12-22 08:51 - Karsten Dambekalns

- Target version changed from 1.0.2 to 1.0.3

#13 - 2012-03-07 18:52 - Karsten Dambekalns

- Target version changed from 1.0.3 to 1.0.4

#14 - 2012-04-18 22:49 - Karsten Dambekalns

- Target version changed from 1.0.4 to 1.0.5

#15 - 2012-06-05 21:20 - Karsten Dambekalns

- Target version changed from 1.0.5 to 1.1 RC1

#16 - 2012-06-25 22:15 - Karsten Dambekalns

- Status changed from Needs Feedback to On Hold

- Target version deleted (1.1 RC1)

- Has patch set to No

#17 - 2013-04-08 17:53 - Adrian Föder

Also stumbled upon that behavior currently.

Note #5 boils it down; in `\Doctrine\ORM\Mapping\ClassMetadataInfo::newInstance` you can find

```
1 /**
2  * Creates a new instance of the mapped class, without invoking the constructor.
3  *
4  * @return object
5  */
6 public function newInstance()
7 {
8     if ($this->_prototype === null) {
9         $this->_prototype = unserialize(sprintf('O:%d:"%s":0:{}'.strlen($this->name), $this->name));
10    }
11
12    return clone $this->_prototype;
13 }
```

Line 9 in that example calls, e.g., `unserialize("O:52:"Acme\Acme\Domain\Model\Foo\Bar":0:{}".strlen($this->name))`, which invokes `__wakeup` and immediately after that, `initializeObject()`, but with obviously no data to work on (see what has been unserialized...)