

TYPO3.Flow - Bug # 51188

Status:	New	Priority:	Should have
Author:	Andreas Wolf	Category:	Persistence
Created:	2013-08-19	Assigned To:	
Updated:	2013-12-07	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: Git master			
Subject:	Doctrine does not respect AOP-injected properties		
Description			
<p>When I add a property to a model via an aspect, this property is not taken into account by the automagic Doctrine schema generation. Therefore, the database field is not created and the property is also not persisted if I manually add the field via a hand-written migration.</p> <p>Doctrine does not respect the properties because the reflection components do not return the property that was woven in by the ProxyClassBuilder. This can be reproduced by examining the output of ReflectionService::getClassPropertyNames() - the properties are missing there.</p> <p>I think the root cause for this is that the reflection information is cached during compile-time before the proxy class files are built. Therefore, the injected properties are not available in the reflected (original) classes.</p> <p>One potential caveat when fixing this is that both properties from MyClass and MyClass_Original have to be taken into account - I don't know how hard this will get using the PHP reflection mechanisms.</p>			
Related issues:			
related to TYPO3.Flow - Bug # 27045: Introduced properties are not available ...			New 2011-05-26

History

#1 - 2013-08-19 17:18 - Rafael Kähm

You can use following dirty hack to show introduced properties in database:

You can put assignPropertiesToORM() method to your aspect class and modify rows 82-89.

```
1<?php 2namespace ..... 3 4useTYPO3\Flow\AnnotationsasFlow; 5useDoctrine\ORM\MappingasORM; 6 7** 8 * 9 * @Flow\Scope("singleton")10
* @Flow\Aspect11 */12classPropertyIntroductionAspect {1314/**15     * @Flow\Inject16     * @var \TYPO3\Flow\Reflection\ReflectionService17     */
18protected$reflectionService;1920/**21     * @param \TYPO3\Flow\Reflection\ReflectionService $reflectionService Description22     */23publicfunction
injectReflectionService(\TYPO3\Flow\Reflection\ReflectionService$reflectionService) {24$this->reflectionService = $reflectionService;25 }2627/**28
* @var string29     * @Flow\Introduce("some Pointcut")30     */31protected$purpose;3233/**34     * @var string35     * @Flow\Introduce("some other
Pointcut")36     */37protected$somethingElse;3839/**40     * Dirty hack for reflection service by introduced properties -> introduced properties are not in
persistence layer41     * 42     * @todo : Remove this advice if http://forge.typo3.org/issues/27045 is resolved.43     * 44     * @param
\TYPO3\Flow\Aop\JoinPointInterface $joinPoint The current join point45     * @return void46     *
@Flow\Before("method(\TYPO3\Flow\Persistence\Doctrine\EntityManagerFactory->create())")47     */48publicfunctionassignPropertiesToORM(\TYPO3
\Flow\Aop\JoinPointInterface$joinPoint) {49// those are added as property even if not tagged with entity/valueobject50$propertyTypeWhiteList = array(
51'DateTime',52'SplObjectStorage',53'Doctrine\Common\Collections\Collection',54'Doctrine\Common\Collections\ArrayCollection'55     );5657
$aspectClassName = get_class($this);58$aspectClassShema = $this->reflectionService->getClassSchema($aspectClassName);5960
$introducedPropertyNames = $this->reflectionService->getPropertyNamesByAnnotation($aspectClassName, 'TYPO3\Flow\Annotations\Introduce');61
62foreach ($introducedPropertyNamesas$propertyName) {63$isTransientProperty = $this->reflectionService->isPropertyAnnotatedWith(
$aspectClassName, $propertyName, 'TYPO3\Flow\Annotations\Transient');64if ($isTransientProperty) {continue;}6566$declaredType = trim(implode("
", $this->reflectionService->getPropertyTagValues($aspectClassName, $propertyName, 'var')), '\\');67if (preg_match('/\s/', $declaredType) === 1 ||
```

```

empty($declaredType) {68thrownew \TYPO3\Flow\Reflection\Exception\InvalidPropertyTypeException(sprintf('Introduced in "%s" property "%s" has
no @var annotation or type is not defined or is not annotated as "TYPO3\Flow\Annotations\Transient". Please define type for "%s" or annotate it as
"TYPO3\Flow\Annotations\Transient".', $aspectClassName, $propertyName, $propertyName), 1366547612);69      }7071try {72$parsedType = \
TYPO3\Flow\Utility\TypeHandling::parseType($declaredType);73      } catch (\TYPO3\Flow\Utility\Exception\InvalidTypeException$exception) {74
thrownew \InvalidArgumentException(sprintf($exception->getMessage(), 'class "' . $aspectClassName . '" for property "' . $propertyName . '"'),
1366551857);75      }76if (!in_array($parsedType['type'], $propertyTypeWhiteList)77      && (class_exists($parsedType['type']) ||
interface_exists($parsedType['type']))78      && !($this->reflectionService->isClassAnnotatedWith($parsedType['type'], 'TYPO3\Flow\Annotations\E
ntity') || $this->isClassAnnotatedWith($parsedType['type'], 'Doctrine\ORM\Mapping\Entity') || $this->isClassAnnotatedWith($parsedType['type'], 'TYPO3
\Flow\Annotations\ValueObject')) {79continue;80      }8182// get all affected classes stuff is still not present but if it is possible here then you can
add this Property to all affected classes83//      foreach ($affectedClasses as $affectedClass){84//          $classSchema =
$this->reflectionService->getClassSchema($affectedClass);85//          $classSchema->addProperty($propertyName, $declaredType,
$this->isPropertyAnnotatedWith($aspectClassName, $propertyName, 'TYPO3\Flow\Annotations\Lazy'));86//          if
($this->reflectionService->isPropertyAnnotatedWith($className, $propertyName, 'TYPO3\Flow\Annotations\Identity')) {87//
$classSchema->markAsIdentityProperty($propertyName);88//          }89//          }9091      }92 }93}94?>
this code is https://gist.github.com/RafaelKa/6270217 here

```

#2 - 2013-08-19 17:30 - Rafael Kähm

if you know which class becomes new properties then you need only following inside of assignPropertiesToORM() method:

```

1<?php 2 3// ... 4/** 5      * Dirty hack for reflection service by introduced properties -> introduced properties are not in persistence layer 6      * 7      *
@todo : Remove this advice if http://forge.typo3.org/issues/27045 is resolved. 8      * 9      * param \TYPO3\Flow\Aop\JoinPointInterface $joinPoint The
current join point10      * @return void11      * @Flow\Before("method(TYPO3\Flow\Persistence\Doctrine\EntityManagerFactory->create())")12      */13
publicfunctionassignPropertiesToORM() {14// ...15$affectedProxy = $this->reflectionService->getClassSchema($affectedClassName);16
$affectedProxy->addProperty('propertyName', 'string');17// ...18 }

```