

## TYPO3.Flow - Feature # 31788

<b>Status:</b>	Rejected	<b>Priority:</b>	Should have
<b>Author:</b>	rottenrice no-lastname-given	<b>Category:</b>	Reflection
<b>Created:</b>	2011-11-13	<b>Assigned To:</b>	Karsten Dambekalns
<b>Updated:</b>	2012-03-07	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	[PERFORMANCE] cache implementation for AnnotationReader in Flow3AnnotationDriver		
<b>Description</b>			
<p>doctrine has classes to cache the parsed annotations. with cached annotations you can boost the parsetime by ~200ms (=15%, tested with TYPO3.Blog).</p> <p>some caching strategies:</p> <pre>\$this-&gt;reader = new \Doctrine\Common\Annotations\CachedReader(     new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationReader()),     new \Doctrine\Common\Cache\ApcCache() );  \$this-&gt;reader = new \Doctrine\Common\Annotations\CachedReader(     new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationReader()),     new \Doctrine\Common\Cache\ArrayCache() );  \$this-&gt;reader = new \Doctrine\Common\Annotations\FileCachedReader(     new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationReader()),     CACHING_DIR );</pre> <p>it would be better to use the Objects.yaml to set the caching strategy instead of using __construct()</p> <pre>/**  * @var \Doctrine\Common\Annotations\AnnotationReader  */ protected \$reader;</pre> <p>i think it is better to use the interface \Doctrine\Common\Annotations\Reader</p> <pre>-- class: TYPO3\FLOW3\Persistence\Doctrine\Mapping\Driver\Flow3AnnotationDriver</pre>			

### History

#1 - 2011-11-25 11:32 - Christopher Hlubek

+1 for the interface. We should implement a custom CachedReader using the FLOW3 caching framework. Should be fairly easy.

**#2 - 2011-12-14 15:40 - Karsten Dambekalns**

- *Category set to Reflection*
- *Status changed from New to Needs Feedback*
- *Assigned To set to Karsten Dambekalns*

Christopher Hlubek wrote:

| +1 for the interface. We should implement a custom *CachedReader* using the *FLOW3* caching framework. Should be fairly easy.

The strategy currently is to cache the results the *ReflectionService* builds with the reader. No doubled caching, if it can be avoided.

**#3 - 2012-03-07 16:07 - Karsten Dambekalns**

- *Status changed from Needs Feedback to Rejected*

Recent improvements have shown the caching we have in place does the job.