

## TYPO3.Flow - Feature # 26560

<b>Status:</b>	Resolved	<b>Priority:</b>	Should have
<b>Author:</b>	Christopher Hlubek	<b>Category:</b>	AOP
<b>Created:</b>	2011-05-02	<b>Assigned To:</b>	
<b>Updated:</b>	2012-03-14	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	Use reverse index for AOP matching		
<b>Description</b>			
<p>The AOP matching done inside the ProxyClassBuilder is very slow. It doesn't scales well, because the complexity is tied by the number of pointcuts * methods (over all proxyable classes).</p> <p>Since this number could get very large and the matching is a costly operation, we should use a reverse index to get only the interesting pointcuts for a given target class.</p> <p>Most pointcuts like <code>method(F3\FLOW3\MVC\Web\Routing\Router-&gt;resolve())</code> could be easily indexed by using the class name and method name as the keys. Also the class and method tags could be used in an index. The setting pointcut filter could be evaluated and skipped. Since we have the</p> <p>This could greatly reduce the number of pointcut matchings done for each method.</p>			

### History

#### #1 - 2011-05-02 17:24 - Christopher Hlubek

One example of a slow filter: PointcutMethodTaggedWithFilter:

It uses the very slow operation `getMethodTagsValues` which does a method reflection every time (with no caching). This filter is executed against every method.

#### #2 - 2011-08-04 09:02 - Sebastian Kurfuerst

- Category set to AOP
- Target version set to 1230

#### #3 - 2011-10-20 01:43 - Karsten Dambekalns

- Target version deleted (1230)

#### #4 - 2012-03-14 14:23 - Christian Müller

- Status changed from New to Resolved
- Has patch set to No

I will close this one, the results of the code sprint are tagged with #9568, and should include this optimization too.

#### #5 - 2012-03-14 14:46 - Christopher Hlubek

Okay, the issue would be resolved for me with the new AOP optimizations.

Christian Mueller wrote:

| *I will close this one, the results of the code sprint are tagged with #9568, and should include this optimization too.*