

TYPO3.Flow - Bug #28638

Signals can't be defined in abstract classes

2011-08-01 12:44 - Bastian Waidelich

Status:	Resolved	Start date:	2011-08-01
Priority:	Should have	Due date:	
Assignee:	Robert Lemke	% Done:	100%
Category:	AOP	Estimated time:	0.00 hour
Target version:	1.0.0	Complexity:	
PHP Version:			
Has patch:			

Description

The AOP ProxyClassBuilder produces invalid code when signals are defined in an abstract class.

Steps to reproduce:

```
AbstractSignalTest.php
<?php
namespace Foo\Bar;
abstract class AbstractSignalTest {

    public function testSignal() {
        $this->emitTestSignal();
    }

    /**
     * @signal
     */
    public function emitTestSignal() {}
}

?>

SignalTest.php
<?php
namespace Foo\Bar;
class SignalTest extends AbstractSignalTest {

}

?>

In some controller:
$signalTest = new \Foo\Bar\SignalTest();
$signalTest->testSignal();
```

Result:

```
#1: Notice: Undefined index: emitTestSignal in Development\Cache\Code\FLOW3_Object_Classes\Foo_Bar_AbtractSignalTest.php line 98
```

The problem is, that the proxy code tries to access `$this->FLOW3_AOP_Proxy_targetMethodsAndGroupedAdvices` which is declared private.

Associated revisions

Revision 43f08cbd - 2011-09-19 17:05 - Bastian Waidelich

[BUGFIX] Proxy code of advised abstract classes don't produce errors

The AOP ProxyClassBuilder produced invalid code when signals were

defined in an abstract class. This was due to invalid proxy class code and a wrong initialization procedure.

This change fixes the issue by making sure that the advice information is also built for parent proxy classes when sub classes are used.

Additionally this change set contains functional tests for the AOP proxy and the signal slot mechanism.

Change-Id: I6480321c117dc0eb264fda45a952d27505156f82
Fixes: #28638

History

#1 - 2011-08-01 12:46 - Mr. Hudson

- Status changed from *Accepted* to *Under Review*

Patch set 1 of change I6480321c117dc0eb264fda45a952d27505156f82 has been pushed to the review server.
It is available at <http://review.typo3.org/3971>

#2 - 2011-08-01 17:01 - Bastian Waidelich

- Priority changed from *Should have* to *Must have*

#3 - 2011-08-04 09:13 - Sebastian Kurfuerst

- Priority changed from *Must have* to *Should have*

#4 - 2011-08-24 10:33 - Robert Lemke

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

As far as I can see, the problem is not limited to just signals, but any advised method of an abstract class. If so, we should rephrase / rename this issue.

#5 - 2011-09-09 15:17 - Robert Lemke

- Target version changed from *1.0 beta 2* to *1.0.0*

#6 - 2011-09-19 17:05 - Mr. Hudson

Patch set 2 of change I6480321c117dc0eb264fda45a952d27505156f82 has been pushed to the review server.
It is available at <http://review.typo3.org/3971>

#7 - 2011-09-20 10:36 - Bastian Waidelich

- Status changed from *Under Review* to *Resolved*

- % Done changed from *0* to *100*

Applied in changeset [43f08cbd553613dea9c927c297ab677440806f04](#).

#8 - 2011-09-20 11:54 - Christopher Hlubek

I have some kind of a regression with this after further testing (didn't notice this at first):

If a class uses a magic `__call` method to handle dynamic function calls, a `is_callable('parent::FLOW3_AOP...')` will always return `TRUE`, such that an object receives a call with `FLOW3_AOP_Proxy_buildMethodsAndAdvicesArray`. This could be misleading and caused problems with the SOAP package for me. A quick fix is to filter such method calls (e.g. starting with "FLOW3_AOP") inside a custom `__call` method, but I consider this a quick hack and not a long-term solution.

Can we detect somehow, that a parent *really* has this method during compilation (e.g. reflection)?

#9 - 2011-09-20 11:54 - Christopher Hlubek

- Status changed from *Resolved* to *Needs Feedback*

#10 - 2011-09-21 11:53 - Bastian Waidelich

- Assignee changed from *Bastian Waidelich* to *Robert Lemke*

#11 - 2011-09-29 12:35 - Karsten Dambekalns

- Status changed from *Needs Feedback* to *Resolved*

#12 - 2011-09-29 12:36 - Karsten Dambekalns

The regression has been fixed with lcf6bdf3f789162afbca61d7cf915dbb7ecd583d5 (<https://review.typo3.org/#change.5409>)