

## TYPO3.Flow - Bug # 37574

<b>Status:</b>	Resolved	<b>Priority:</b>	Won't have this time
<b>Author:</b>	Bastian Waidelich	<b>Category:</b>	Reflection
<b>Created:</b>	2012-05-29	<b>Assigned To:</b>	Karsten Dambekalns
<b>Updated:</b>	2012-08-17	<b>Due date:</b>	
<b>PHP Version:</b>	5.3		
<b>Has patch:</b>	No		
<b>Complexity:</b>	nightmare		
<b>Affected Flow version:</b>	FLOW3 1.1.0 beta 2		
<b>Subject:</b>	ReflectionData is not regenerated automatically		
<b>Description</b>			
	If the signature of an action changes in a controller or I add a @FLOW3\Inject annotation I would expect the Controllers reflection data to be recreated in Development context. But that's not the case (even if the FLOW3 package and my custom package are not frozen).		
<b>Subtasks:</b>			
Bug # 39300: blocking ReflectionService: Call to a member function get() on a non-object			<b>Closed</b>
<b>Related issues:</b>			
related to TYPO3.Flow - Bug # 37223: Packages can be frozen in Production Con...	<b>Resolved</b>	<b>2012-05-16</b>	
related to TYPO3.Flow - Task # 37726: Packages should not be frozen by default	<b>Resolved</b>	<b>2012-06-04</b>	
related to TYPO3.Flow - Bug # 37815: Object are not persisted	<b>Closed</b>	<b>2012-06-07</b>	

### Associated revisions

#### Revision d57781ab - 2012-07-09 15:06 - Karsten Dambekalns

[TASK] Clean up some code

Fixes some comments, style issues and API usage.

Change-Id: I124c211ea218f8ffcad791bcc9eca2403f10b55f

Related: #37574

Related: #34875

Releases: 1.1, 1.2

#### Revision 8e393d38 - 2012-07-10 12:34 - Karsten Dambekalns

[TASK] Clean up some code

Fixes some comments, style issues and API usage.

Change-Id: I124c211ea218f8ffcad791bcc9eca2403f10b55f

Related: #37574

Related: #34875

Releases: 1.1, 1.2

#### Revision d3553264 - 2012-07-13 17:15 - Karsten Dambekalns

[BUGFIX] Fix order of monitoring and proxy building

During bootstrap file monitoring would be run too late, so that while change detection worked fine, the results were not used correctly: Reflection data was updated, but that happened after DI proxy building, so that e.g. added Inject annotations would not have the expected effect.

This change moves file monitoring before reflection initialization to fix this.

Change-Id: I4a16257d072e1eef99cf48bd60bf83f8818f0e1d

Fixes: #37574

Releases: 1.1, 1.2

#### **Revision 2b09113f - 2012-07-16 13:02 - Karsten Dambekalns**

[BUGFIX] Fix order of monitoring and proxy building

During bootstrap file monitoring would be run too late, so that while change detection worked fine, the results were not used correctly: Reflection data was updated, but that happened after DI proxy building, so that e.g. added Inject annotations would not have the expected effect.

This change moves file monitoring before reflection initialization to fix this.

Change-Id: I4a16257d072e1eef99cf48bd60bf83f8818f0e1d

Fixes: #37574

Releases: 1.1, 1.2

## **History**

---

### **#1 - 2012-06-01 08:56 - Robert Lemke**

- *Category set to Reflection*
- *Priority changed from Should have to Must have*
- *PHP Version set to 5.3*
- *Complexity set to medium*
- *Affected Flow version changed from Git master to FLOW3 1.1.0 beta 2*

This has some more severe consequences:

- freshly clone a FLOW3 distribution
- set file permissions and database settings
- `./flow3 doctrine:migrate`
- `./flow3 kickstart:actioncontroller --generate-actions --generate-templates --generate-related Shop Book`
- create a new Book through the form
- the message says that a new book was created but it's not persisted! No errors in log!
- `./flow3 package:refreeze`
- now it works!

So, as a workaround - until this bug is fixed - make sure that all packages are unfrozen (./flow3 package:unfreeze all)

## #2 - 2012-06-01 08:59 - Karsten Dambekalns

Packages should not be frozen by default, that way you'd need to learn about them before you can fall into any traps.

## #3 - 2012-06-02 12:36 - Bastian Waidelich

Karsten Dambekalns wrote:

| *Packages should not be frozen by default*

They are not, but the issue occurs even if the package is **not frozen** (see description above)

## #4 - 2012-06-04 11:40 - Karsten Dambekalns

- Status changed from New to Accepted
- Assigned To set to Karsten Dambekalns

## #5 - 2012-06-04 12:36 - Karsten Dambekalns

Bastian Waidelich wrote:

| *Karsten Dambekalns wrote:*

| | *Packages should not be frozen by default*

| *They are not, but the issue occurs even if the package is **not frozen** (see description above)*

Yes. The note was more a braindump, see #37726 for the separate issue.

## #6 - 2012-06-04 14:17 - Karsten Dambekalns

Robert Lemke wrote:

...  
| - *the message says that a new book was created but it's not persisted! No errors in log!*

With the current 1.1 codebase that works fine for me.

## #7 - 2012-06-04 18:18 - Bastian Waidelich

While playing around with this a bit further, I made following observation:

Adding a new property with @FLOW3\Inject annotation **does** work sometimes.

Sometimes it doesn't but then - after changing something else in the same class - it does work suddenly.

My theory is, that the ModificationTimeStrategy doesn't work correctly. Maybe it compares the modification times with the time **after** the proxy classes have been built last, not with the time **before** they have been built?

I didn't check this, just a theory ;)

**#8 - 2012-06-04 18:31 - Bastian Waidelich**

Bastian Waidelich wrote:

*Maybe it compares the modification times with the time **after** the proxy classes have been built last, not with the time **before** they have been built?*

It does not compare the modification times in fact. But the issue might be, that the modificationTimes are stored in the cache while the files are changed or something. If I wait long enough **before** and **after** I do one single change, it seems to work.

**#9 - 2012-06-04 21:26 - Karsten Dambekalns**

No, looking at the logs the file is always detected as modified, just the effect of that varies. YMMV.

**#10 - 2012-06-05 14:08 - Bastian Waidelich**

Karsten Dambekalns wrote:

*No, looking at the logs the file is always detected as modified just the effect of that varies. YMMV.*

Mh, then it's probably some kind of race condition still.. I experience a related issue lately:

If I modify a controller and hit refresh too soon, I get following error:

```
Runtime Notice: Declaration of My\Package\Controller\Project\SomeController::initializeView() should be compatible with that of My\Package\Controller\Project\SomeController_Original::initializeView()
```

Somehow the site is not locked correctly or so!? (sorry for the wild guesses, but maybe it helps anyways)

**#11 - 2012-06-26 12:23 - Karsten Dambekalns**

*- Complexity changed from medium to nightmare*

**#12 - 2012-06-26 17:39 - Karsten Dambekalns**

Ok, after adding some debugging instruments, a picture forms. The following is a log of what happens after a change to a controller. That change was the addition of FLOW3Inject to the already existing property \$objectManagerInstance.

The first run picks up the change to the file and triggers rebuilding the DI proxy:

```
12-06-26 17:33:05 17933 DEBUG FLOW3 runtime-typo3.flow3:errorhandling
12-06-26 17:33:05 17933 DEBUG FLOW3 runtime-typo3.flow3:cachemanagement
12-06-26 17:33:05 17933 DEBUG FLOW3 runtime-typo3.flow3:objectmanagement:proxyclasses
```

12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:errorhandling  
12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:cachemanagement  
12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:cachemanagement:forceflush  
12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:objectmanagement:compiletime:create  
12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:reflectionservice  
12-06-26 17:33:05 17934 DEBUG FLOW3 compiletime-typo3.flow3:objectmanagement:compiletime:finalize  
12-06-26 17:33:06 17934 DEBUG FLOW3 compiletime-typo3.flow3:systemfilemonitor  
12-06-26 17:33:06 17934 INFO FLOW3 File Monitor "FLOW3\_ClassFiles" detected 1 changed files and 0 changed directories.  
12-06-26 17:33:06 17934 INFO FLOW3 Buidling DI proxy for "TYPO3\Welcome\Controller\StandardController".

12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:classloader:cache  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:reflectionservice  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:objectmanagement:runtime  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:persistence  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:session  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:resources  
12-06-26 17:33:06 17933 DEBUG FLOW3 runtime-typo3.flow3:i18n

At this point the rebuilt proxy does **not** include injector code for the \$objectManagerInstance! This is because only during the second run the ReflectionService picks up the change and learns about the new annotation:

12-06-26 17:33:34 17971 DEBUG FLOW3 runtime-typo3.flow3:errorhandling  
12-06-26 17:33:34 17971 DEBUG FLOW3 runtime-typo3.flow3:cachemanagement  
12-06-26 17:33:34 17971 DEBUG FLOW3 runtime-typo3.flow3:objectmanagement:proxyclasses

12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:errorhandling  
12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:cachemanagement  
12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:cachemanagement:forceflush  
12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:objectmanagement:compiletime:create  
12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:reflectionservice  
12-06-26 17:33:34 17972 DEBUG FLOW3 compiletime-typo3.flow3:objectmanagement:compiletime:finalize  
12-06-26 17:33:34 17972 DEBUG FLOW3 Forget class TYPO3\Welcome\Controller\StandardController  
12-06-26 17:33:34 17972 DEBUG FLOW3 Reflected class names did not match class names to reflect  
12-06-26 17:33:34 17972 DEBUG FLOW3 Reflecting class TYPO3\Welcome\Controller\StandardController  
12-06-26 17:33:34 17972 INFO FLOW3 Reflected 1 emerged classes.  
12-06-26 17:33:35 17972 DEBUG FLOW3 compiletime-typo3.flow3:systemfilemonitor  
12-06-26 17:33:35 17972 DEBUG FLOW3 Found 1 classes whose reflection data was not cached previously.

12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:classloader:cache  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:reflectionservice  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:objectmanagement:runtime  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:persistence  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:session  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:resources  
12-06-26 17:33:35 17971 DEBUG FLOW3 runtime-typo3.flow3:i18n

At this point the proxy would be built correctly, but the system considers it up-to-date already.

If now something else is changed in that file, the proxy will be rebuilt with the new reflection data and will be correct. Depending on the nature of that second change, the proxy will still be out of date though...

**#13 - 2012-06-27 15:50 - Karsten Dambekalns**

- Target version changed from 1.1 RC1 to 1.1

**#14 - 2012-07-04 09:58 - Gerrit Code Review**

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/12575>

**#15 - 2012-07-13 17:17 - Gerrit Code Review**

Patch set 2 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/12575>

**#16 - 2012-07-13 17:35 - Karsten Dambekalns**

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:d35532647cd6c6279ff1c5e7cc9123371778c1f1.

**#17 - 2012-07-16 13:02 - Gerrit Code Review**

- Status changed from Resolved to Under Review

Patch set 1 for branch **FLOW3-1.1** has been pushed to the review server.

It is available at <http://review.typo3.org/12823>

**#18 - 2012-07-18 11:20 - Karsten Dambekalns**

- Status changed from Under Review to Resolved

- Target version changed from 1.1 to 1.1 RC2