TYPO3.Flow - Bug # 43659

Status:	Resolved	**Priority:**	Must have
Author:	Stephan Schuler	**Category:**	Object
Created:	2012-12-06	**Assigned To:**	Robert Lemke
Updated:	2014-07-11	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version:	Git master		
Subject:	Proxy class building calls __construct and initializeObject before DI objects are present		

Description

Hey there.

Given the following situation:

```
class ObjectA;
class ObjectB extends ObjectA;
class ObjectC extends ObjectB;
```

The code of pre 1.1 did this:

```
new ObjectC ();

ObjectC_Proxy::__construct(); // triggered by my code
ObjectC_Original::__construct(); // triggered autoamtically by ObjectA_Proxy::__construct(), directly addressed as parent::__construct()

ObjectB_Proxy::__construct(); // triggered autoamtically by ObjectC_Proxy::__construct(), directly addressed as parent::__construct()
ObjectB_Original::__construct(); // triggered autoamtically by ObjectC_Proxy::__construct(), directly addressed as parent::__construct()

ObjectA_Proxy::__construct(); // triggered autoamtically by ObjectB_Proxy::__construct(), directly addressed as parent::__construct()
ObjectA_Original::__construct(); // triggered autoamtically by ObjectB_Proxy::__construct(), directly addressed as parent::__construct()

ObjectC_Proxy::DI(); // Should be ObjectA_Proxy, but it's "$this->", so the most outer method is used. But doesn't matter because it works.
ObjectC_Original::initializeObject(); // Should be ObjectA_Original, but it's "$this->", so the most outer method is used. But doesn't matter, it works.

ObjectC_Proxy::DI(); // Should be ObjectB_Proxy, but it's "$this->", so the most outer method is used. But doesn't matter because it has nothing to do.
ObjectC_Original::initializeObject(); // Should be ObjectB_Original, but it's "$this->" as well.

ObjectC_Proxy::DI(); // This should really be ObjectC_Proxy. But doesn't mater, it has been called three times before
```

ObjectC_Original::initializeObject(); // This should really be ObjectC_Original. But doesn't mater, it has been called three times before

The current code acts like this:

new ObjectC ();

ObjectC_Proxy::__construct(); // triggered by my code
ObjectC_Original::__construct(); // triggered autoamtically by ObjectA_Proxy::__construct(), directly addressed as parent::__construct()

ObjectB_Proxy::__construct(); // triggered autoamtically by ObjectC_Proxy::__construct(), directly addressed as parent::__construct()
ObjectB_Original::__construct(); // triggered autoamtically by ObjectC_Proxy::__construct(), directly addressed as parent::__construct()

ObjectA_Proxy::__construct(); // triggered autoamtically by ObjectB_Proxy::__construct(), directly addressed as parent::__construct()
ObjectA_Original::__construct(); // triggered autoamtically by ObjectB_Proxy::__construct(), directly addressed as parent::__construct()

ObjectC_Original::initializeObject(); // Should be ObjectA_Original, but it's "$this->", so the most outer method is used. This is way to early, no DI done until now.

ObjectC_Original::initializeObject(); // Should be ObjectB_Original, but it's "$this->" as well. To early, too.

ObjectC_Proxy::DI(); // Now we get the correct DI, but a little to late.
ObjectC_Original::initializeObject(); // This is the first call of ObjectC_Original::initializeObject() that really acts as expected since the other two calls had no DI properties available.

In my oppinion, the DI sould be available in __construct. So I would move the DI stuff right before __construct and introduce a boolean helper flag to avoid multiple calls of DI. In general, I would call Flow_Proxy_injectProperties seeral times but make the injected method aware of multiple calls. This should clean up the genreated __construct a little.

Please see my DependencyHierarchy*.tar.gz demo files attached. Those show tue current and past situation and demonstrate the error.

Related issues:

duplicated by TYPO3.Flow - Bug # 47975: initializeObject in a Entity is calle...	**Resolved**	**2013-05-06**

Associated revisions

Revision 6d2ea6b3 - 2013-10-11 10:28 - Robert Lemke

[BUGFIX] Injected properties are not available in initializeObject()

This fixes an issue where properties injected to the parent class A

were not available when the initializeObject() method of sub class B
was called.

The root cause of this bug was that, in order to avoid double injection,
injectProperties() was only called in the constructor of sub class B
and not in the constructor of parent class A. The initializeObject()
methods however, were still executed in both constructors.

Change-Id: I7fbe4fc61029af60924356f49aef4964dbb346f7
Resolves: #43659
Releases: 2.0, master

Revision 8a28ec3b - 2014-01-31 16:08 - Robert Lemke

[BUGFIX] Injected properties are not available in initializeObject()

This fixes an issue where properties injected to the parent class A
were not available when the initializeObject() method of sub class B
was called.

The root cause of this bug was that, in order to avoid double injection,
injectProperties() was only called in the constructor of sub class B
and not in the constructor of parent class A. The initializeObject()
methods however, were still executed in both constructors.

Change-Id: I7fbe4fc61029af60924356f49aef4964dbb346f7
Resolves: #43659
Releases: 2.0, master

History

#1 - 2012-12-13 12:35 - Karsten Dambekalns

- Affected Flow version changed from Git 1.2 (master) to Git master

#2 - 2013-10-11 10:11 - Robert Lemke

- Status changed from New to Accepted
- Assigned To set to Robert Lemke
- Target version set to 2.x

#3 - 2013-10-11 10:30 - 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 https://review.typo3.org/24550

#4 - 2014-01-31 16:08 - Gerrit Code Review

Patch set 1 for branch **2.0** of project **Packages/TYPO3.Flow** has been pushed to the review server.
It is available at https://review.typo3.org/27208

#5 - 2014-07-11 20:19 - Robert Lemke

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:6d2ea6b3fc5db90f3c7e6ccd0dea5304ec32ef5e.

Files

DependencyHierarchy-1.1.tar.gz	1.7 kB	2012-12-06	Stephan Schuler
DependencyHierarchy-1-2.tar.gz	1.8 kB	2012-12-06	Stephan Schuler