

TYPO3.Flow - Feature # 8258

Status:	Resolved	Priority:	Should have
Author:	Sebastian Kurfuerst	Category:	Object
Created:	2010-06-15	Assigned To:	Sebastian Kurfuerst
Updated:	2010-10-20	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Subject:	call initializeObject() also after Reconstitution		
Description			
<p>for some cases, it makes sense to call the initializer method (initializeObject()) also after reconstituting an object, and re-injecting its dependencies -- so one can set up calculated properties where a dependency is needed for calculation.</p> <p>We agreed to pass a parameter (constant) to initializeObject, so one can determine if it is the "NEW" or the "RECONSTITUTE"-case.</p> <p>TODO: also Documentation must be updated.</p>			

Associated revisions

Revision 8d3c9924 - 2010-06-16 12:25 - Sebastian Kurfuerst

[!!!][+FEATURE] FLOW3 (Object): The lifecycle initialization method (typically initializeObject()) is now also called after recreating an object (f.e. when it is recreated from persistent storage). Additionally, initializeObject() now gets a parameter to determine if creation or recreation took place. Resolves #8258.

History

#1 - 2010-06-15 18:36 - Sebastian Kurfuerst

- File flow3-8258.patch added

This is the first version of the Object Container which calls initializeObject also after reconstitution.

Comments are of course highly welcome!

#2 - 2010-06-15 20:55 - Karsten Dambekalns

- Category set to Object

The methods buildLifecycleInitializationCommand() and buildLifecycleReconstitutionCommand() differ only in the used constant. Better use only one and pass the constant as parameter, no?

#3 - 2010-06-16 09:05 - Robert Lemke

- Target version set to 1.0 alpha 10

Generally fine but please consider the following changes:

- (Karsten's comment) The methods `buildLifecycleInitializationCommand()` and `buildLifecycleReconstitutionCommand()` differ only in the used constant. Better use only one and pass the constant as parameter, no?

- It's a bit ugly, but maybe more meaningful: How about naming the constants "INITIALIZATIONCAUSE_CREATED" and "INITIALIZATIONCAUSE_RECREATED" respectively?

- As a further optimization step you can use the value of the constant directly when building the static object container code because it's just a cache file anyway and not supposed to be read by humans:

```
$command = "\n\t\t\t$o->$lifecycleInitializationMethodName(" .  
\F3\FLOW3\Object\Container\ObjectContainerInterface::OBJECT_RECONSTITUTED . ");";
```

#4 - 2010-06-16 09:10 - Sebastian Kurfuerst

Hey Robert,

- (Karsten's comment) The methods `buildLifecycleInitializationCommand()` and `buildLifecycleReconstitutionCommand()` differ only in the used constant. Better use only one and pass the constant as parameter, no?

Yep.

- It's a bit ugly, but maybe more meaningful: How about naming the constants "INITIALIZATIONCAUSE_CREATED" and "INITIALIZATIONCAUSE_RECREATED" respectively?

Then I would suggest `INITIALIZATIONCAUSE_CREATED` and `INITIALIZATIONCAUSE_RECONSTITUTED`; would this be better as "recreate"? Is the place for the constants correct?

- As a further optimization step you can use the value of the constant directly when building the static object container code because it's just a cache file anyway and not supposed to be read by humans:

Will do.

Greets,
Sebastian

#5 - 2010-06-16 13:00 - Sebastian Kurfuerst

- Status changed from *Accepted* to *Resolved*

- % Done changed from 0 to 100

Applied in changeset r4517.

Files

flow3-8258.patch

9.1 kB

2010-06-15

Sebastian Kurfuerst