

TYPO3.Flow - Feature # 3581

Status:	Resolved	Priority:	Must have
Author:	Robert Lemke	Category:	Cache
Created:	2009-06-05	Assigned To:	Robert Lemke
Updated:	2010-10-20	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Subject:	Site lock / code cache flushing mechanism		
Description			
We need a concept and implementation for flushing critical caches like the code caches.			
In order to flush the code caches, we need to completely lock the application so that no requests come through while the php classes are deleted or rebuilt (which obviously would lead to some fatal error). This locking mechanism should be as simple as the bootstrap checking for the presence of a file (like FLOW3_SITE_LOCK).			
Related issues:			
related to TYPO3.Flow - Task # 3582: Concurrency stress test		Closed	2009-06-05
related to TYPO3.Flow - Feature # 3649: Caches should be flushed on package v...		Resolved	2009-06-12

Associated revisions

Revision 3990fb87 - 2009-06-23 11:58 - Robert Lemke

- [TASK] FLOW3 (Monitor): Renamed F3\FLOW3\Monitor\ChangeDetectionStrategy\ModificationTime to F3\FLOW3\Monitor\ChangeDetectionStrategy\ModificationTimeStrategy
- [TASK] FLOW3 (Core): Move the F3\FLOW3\FLOW3 class into its own sub package "Core". The class is now called "Bootstrap".
- [+FEAT] FLOW3 (Core): Implemented a basic locking mechanism which locks the whole site while code caches are cleared. It's basically working but currently not customizable. Try it out by changing some code somewhere and then run two requests (in dev context) almost simultaneously. The Lock Manager is invoked on file changes through the Signal Slot mechanism. Resolves #3581
- [TASK] FLOW3: Simplified the index.php bootstrap a little and adjusted paths for the new Core sub package.
- [FIX] FLOW3 (SignalSlot): Added a check within the Signal Slot dispatcher which now throws a meaningful exception if a given slot class does not exist. Resolves #3730
- [TASK] FLOW3: Updated FLOW3's Package.xml file.
- [TASK] FLOW3: Commented out the SetEnv declaration in the .htaccess file. FLOW3 runs in Production context by default anyway and if the environment is set in the .htaccess file by default, any environment set in a virtual host wouldn't have any effect.

Revision c8148f88 - 2009-06-23 11:58 - Robert Lemke

- [TASK] FLOW3 (Monitor): Renamed F3\FLOW3\Monitor\ChangeDetectionStrategy\ModificationTime to F3\FLOW3\Monitor\ChangeDetectionStrategy\ModificationTimeStrategy
- [TASK] FLOW3 (Core): Move the F3\FLOW3\FLOW3 class into its own sub package "Core". The class is now called "Bootstrap".
- [+FEAT] FLOW3 (Core): Implemented a basic locking mechanism which locks the whole site while code caches are cleared. It's basically working but currently not customizable. Try it out by changing some code somewhere and then run two requests (in dev context) almost simultaneously. The Lock Manager is invoked on file changes through the Signal Slot mechanism. Resolves #3581
- [TASK] FLOW3: Simplified the index.php bootstrap a little and adjusted paths for the new Core sub package.
- [FIX] FLOW3 (SignalSlot): Added a check within the Signal Slot dispatcher which now throws a meaningful exception if a given slot class does not exist. Resolves #3730
- [TASK] FLOW3: Updated FLOW3's Package.xml file.
- [TASK] FLOW3: Commented out the SetEnv declaration in the .htaccess file. FLOW3 runs in Production context by default anyway and if the

environment is set in the .htaccess file by default, any environment set in a virtual host wouldn't have any effect.

History

#1 - 2009-06-22 13:14 - Robert Lemke

- Status changed from New to Accepted
- Assigned To set to Robert Lemke
- Target version changed from 283 to 1.0 alpha 2

#2 - 2009-06-23 12:00 - Robert Lemke

- Status changed from Accepted to Resolved
- % Done changed from 0 to 100

Applied in changeset r2637.