

TYPO3.Flow - Bug # 34555

Status:	Resolved	Priority:	Should have
Author:	Bastian Waidelich	Category:	Resource
Created:	2012-03-06	Assigned To:	Bastian Waidelich
Updated:	2012-03-23	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: FLOW3 1.0.0			
Subject:	StreamWrapperAdapter::stream_lock does not return StreamWrapper result		
Description			
<p>While writing unit tests for the StreamWrapperAdapter, I came upon this issue: StreamWrapperAdapter::stream_lock() does not return the results of the actual stream wrapper. But according to the docs this "returns TRUE on success or FALSE on failure".</p> <p>Apart from that the StreamWrapperAdapter is lacking unit tests currently</p> <p>Most of the functions in StreamWrapperAdapter call createStreamWrapper() in order to retrieve the actual StreamWrapper. The method looks like this:</p> <pre>1 protected function createStreamWrapper(\$path) { 2 if (\$this->streamWrapper === NULL) { 3 \$explodedPath = explode(':', \$path, 2); 4 \$scheme = array_shift(\$explodedPath); 5 \$this->streamWrapper = self::\$objectManager->create(self::\$registeredStreamWrappers[\$scheme]); 6 } 7 }</pre> <p>If I get it right, this means that the streamWrapper is only instantiated the first time this method is called. createStreamWrapper('scheme1://foo'); createStreamWrapper('scheme2://bar'); would return the same instance.</p> <p>NOTE: As Karsten mentioned in the comments, this is probably not true as PHP will create a fresh instance of the adapter for every use. But this should be secured with tests (also to prevent issues like #34547)</p>			
Related issues:			
related to TYPO3.Flow - Bug # 34547: wrong variable name inside rename() of S...		Resolved	2012-03-06

Associated revisions

Revision 2fd21b11 - 2012-03-21 19:42 - Bastian Waidelich

[BUGFIX] StreamWrapperAdapter::stream_lock does not return result

StreamWrapperAdapter::stream_lock() does not return the results of the actual stream wrapper. But according to the docs this should return TRUE on success and FALSE on failure.

This change fixes this and adds unit tests for all public methods of the adapter.

Change-Id: Iac8e08156221e76ec22e1818578404cb8c2b7697

Fixes: #34555

Releases: 1.1

History

#1 - 2012-03-06 15:25 - Karsten Dambekalns

Bastian Waidelich wrote:

*If I get it right, this means that the streamWrapper is only instantiated the **first** time this method is called. createStreamWrapper('scheme1://foo'); createStreamWrapper('scheme2://bar'); would return the same instance.*

Yes, but for every use of a stream wrapper, PHP creates a fresh instance. Thus this is not an issue AFAICS.

#2 - 2012-03-06 16:28 - Bastian Waidelich

- Tracker changed from Bug to Task
- Subject changed from StreamWrapperAdapter only works for one scheme to Add tests for StreamWrapperAdapter

Karsten Dambekalns wrote:

*If I get it right, this means that the streamWrapper is only instantiated the **first** time [...]*

*Yes, but for every use of a stream wrapper, PHP creates a fresh instance.
Thus this is not an issue AFAICS.*

Ok, this makes sense of course - but we should secure this with some tests IMO. I adjusted the subject accordingly.

#3 - 2012-03-21 19:24 - Bastian Waidelich

- Status changed from New to Accepted
- Assigned To set to Bastian Waidelich
- Target version set to 1.1

#4 - 2012-03-21 19:28 - Bastian Waidelich

- Tracker changed from Task to Bug
- Subject changed from Add tests for StreamWrapperAdapter to StreamWrapperAdapter::stream_lock does not return StreamWrapper result

#5 - 2012-03-21 19:42 - 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/9788>

#6 - 2012-03-21 19:43 - Gerrit Code Review

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

It is available at <http://review.typo3.org/9788>

#7 - 2012-03-23 14:39 - Bastian Waidelich

- *Status changed from Under Review to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:2fd21b11f5b72a668d3d408328a3f7bbd72dc646.