

Core - Bug # 40669

Status:	Resolved	Priority:	Must have
Author:	Steffen Ritter	Category:	File Abstraction Layer (FAL)
Created:	2012-09-06	Assigned To:	Steffen Ritter
Updated:	2013-06-09	Due date:	
TYPO3 Version:	6.0		
PHP Version:			
Complexity:			
Is Regression:			
Sprint Focus:			
Subject:	FAL files get reindexed in storage null when used with tt_content		
Description			
Related issues:			
related to Core - Bug # 39904: Unresized image not displayed in FE after move...		Resolved	2012-08-17
related to Core - Bug # 48998: FAL files get reindexed in storage null when u...		Resolved	2012-09-06
duplicated by Core - Bug # 41783: every image used in frontend gets indexed i...		Closed	2012-10-09
blocked by Core - Bug # 43059: File processing is broken		Resolved	2012-11-17

Associated revisions

Revision 07357595 - 2012-10-10 11:21 - Steffen Ritter

[BUGFIX] Files get indexed twice

Files used within CSS Styled Content get indexed within storage null again, since they are accessed via path instead of file uid.

Change-Id: Ie529c6f9e853f429212265aa779a39086b4a652c

Fixes: #40669

Releases: 6.0

Reviewed-on: <http://review.typo3.org/15439>

Reviewed-by: Steffen Ritter

Tested-by: Steffen Ritter

Reviewed-by: Mario Rimann

Reviewed-by: Andreas Wolf

Tested-by: Andreas Wolf

Revision 66c204f0 - 2012-11-20 23:08 - Andreas Wolf

[TASK] Refactor FAL file processing

File processing is a central part of TYPO3's file usage, as e.g. all images in content elements have to be resized when they don't fit the requirements. However, the current implementation of file processing with FAL has several drawbacks and shortcomings, not to mention quite a few bugs.

This patch brings a completely reworked infrastructure behind the File Processing Service, while keeping the external API unchanged.

The processing to be done is described in tasks, which are part of a ProcessedFile's properties. The processing itself is now moved to processors, which could execute the tasks using different utilities, e.g. ImageMagick or some cloud image processing service. Currently, there is only a local image processor implementation, which relies on ImageMagick/GraphicsMagick (i.e. uses the same configuration as the old processing).

The processed file class now also supports safe handling of unchanged files, i.e. files that should have been processed, but didn't need processing.

Fixes: #43059

Fixes: #39904

Fixes: #40033

Fixes: #40669

Releases: 6.0

Change-Id: I186a46f9923dacd98be655d72be2bd89b43866c2

Reviewed-on: <http://review.typo3.org/14310>

Reviewed-by: Helmut Hummel

Tested-by: Helmut Hummel

History

#1 - 2012-10-09 15:09 - Andreas Wolf

From the description of #41783:

The FrontendContentAdapterService currently just hands over file-names. CSS Styled Content makes file objects out of them again - as only path' are given there, storage 0 is assumed. All images get reindexed in storage 0, and the uid of this file then is used within the processedfile as reference

#2 - 2012-10-09 15:14 - 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/15439>

#3 - 2012-10-09 19:06 - Gerrit Code Review

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

It is available at <http://review.typo3.org/15439>

#4 - 2012-10-09 20:03 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/15439>

#5 - 2012-10-10 11:30 - Steffen Ritter

- *Status changed from Under Review to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:07357595eaae69160171d74ed4cb5e53178a4be1.

#6 - 2013-06-09 13:10 - Klaus Hinum

This is happening (again?) with Typo3 6.1.1

perhaps a regression of the fix of this issue?

<http://forge.typo3.org/issues/47211>