Core - Bug # 50392

Status:	Closed	**Priority:**	Should have
Author:	Oliver Salzburg	**Category:**	File Abstraction Layer (FAL)
Created:	2013-07-24	**Assigned To:**	
Updated:	2015-06-18	**Due date:**	
TYPO3 Version:	6.0		
PHP Version:	5.4		
Complexity:	medium		
Is Regression:	No		
Sprint Focus:			
Subject:	Specifying size in ImageViewHelper does nothing		

Description

I'm having a bit of trouble with cached image sizes (http://forge.typo3.org/issues/44105) and thought I could simply get around it by specifying a desired size in the ViewHelper. But the image is still rendered with the old, cached size.

Fluid:

```
<f:image src="EXT:user_website/Resources/Public/Images/logo.png" width="300" minWidth="300" height="70" minHeight="70" />
```

Result:

```
<img src="/typo3conf/ext/user_website/Resources/Public/Images/logo.png" width="114" height="114">
```

I would have expected the processed file object to inherit the desired size provided from the ViewHelper.

Related issues:			
related to Core - Bug # 44105: Image size does not get updated		**Closed**	**2012-12-19**
related to Core - Bug # 45922: image replacement, width and height are kept e...		**Accepted**	**2013-02-28**
related to Core - Bug # 46446: sys_file doesn't get updated		**Needs Feedback**	**2013-03-20**
related to Core - Bug # 52658: Overriding image does not change the dimensions		**Rejected**	**2013-10-10**
related to FAL Indexing - Bug # 60215: sys_file_processedfile with empty iden...		**Under Review**	**2014-07-09**
related to Core - Task # 64196: Make it possible to re index files from stora...		**Needs Feedback**	**2015-01-08**
related to Core - Bug # 62400: Lot of entries in sys_file_processed with name...		**Rejected**	**2014-10-22**
related to Core - Bug # 61181: FAL: file maxW and maxH are ignored		**Rejected**	**2014-08-25**

History

#1 - 2013-07-29 19:05 - Philipp Gampe

- Status changed from New to Needs Feedback

Did you clear all caches. Changing the Fluid template does neither invalidate page cache nor Fluid template cache.

#2 - 2013-07-29 20:00 - Oliver Salzburg

Philipp Gampe wrote:

> *Did you clear all caches. Changing the Fluid template does neither invalidate page cache nor Fluid template cache.*

Clearing caches has no effect on this issue (but, yes, I've tried it several times anyway). The width and height values will always be those that are stored (and never updated) in the sys_file table. :(

#3 - 2013-07-30 03:05 - Philipp Gampe

- Category set to File Abstraction Layer (FAL)

- Status changed from Needs Feedback to Accepted

- TYPO3 Version changed from 6.1 to 6.0

- Complexity set to medium

Those values should be updated by either the indexing task or by visiting the folder in the file module.

We might want to implement a force index action too.

#4 - 2013-07-30 11:11 - Oliver Salzburg

Philipp Gampe wrote:

> *Those values should be updated by either the indexing task or by visiting the folder in the file module.*
>
> *We might want to implement a force index action too.*

The file in question is a file that is stored in the Resources directory of an extension (speciality in the Bootstrap package https://github.com/Ecodev/bootstrap_package). The indexing task only affects files that are stored in a registered storage location, which this Resources folder isn't. The file also isn't accessible through the File module.

Even removing and reinstalling the extension will not invoke an update of the cached data. I must assume that this *could* also be a potential issue with extension updates.

Either way, I assume this underlying problem best fits within the other reported issue (http://forge.typo3.org/issues/44105), where I reported this initially. I was just hoping that I could get around it by specifying the desired image size explicitly in the ViewHelper, but my requested image size is overruled :(

#5 - 2013-07-30 12:35 - Philipp Gampe

OK, lets track the wrong DB values in #44105 and use this issue for the Fluid part. I agree that explicit values in the viewhelper should overrule automatic detection from the DB.

Are you fine with moving this ticket to Fluid? I will do so this afternoon.

#6 - 2013-07-30 12:39 - Oliver Salzburg

Philipp Gampe wrote:

> *Are you fine with moving this ticket to Fluid? I will do so this afternoon.*

If that was directed at me, sure, move the ticket to wherever you see it fit. Thanks for the effort.

#7 - 2013-07-30 13:22 - Philipp Gampe

- Project changed from Core to Fluid

- Category deleted (File Abstraction Layer (FAL))

#8 - 2014-04-02 20:31 - Markus Klein

- Project changed from Fluid to Core

#9 - 2014-04-02 20:32 - Markus Klein

- Status changed from Accepted to Needs Feedback

- Is Regression set to No

Is this still an issue on 6.2?

#10 - 2014-04-02 20:52 - Oliver Salzburg

Markus Klein wrote:

> *Is this still an issue on 6.2?*

After this much time, we can only *hope* that it isn't :P I'll be sure to report again when the issue comes up.

#11 - 2014-04-02 20:53 - Thomas Luzat

The problem persists for me. I verified this in two cases using 6.2 LTS:

<f:image src="fileadmin/..."/>
<f:image src="EXT:my_ext/..."/>

In both cases width and height of the previous images are used if I clear the caches in the backend and, just to make sure, of typo3temp.

I did replace the images outside of TYPO3 (that is using the shell). I am not so sure whether this is really unrelated to #44105 and that one is really solved. Unfortunately I am not familiar with the file management code yet.

Cheers,

Thomas Luzat

#12 - 2014-05-10 19:31 - Arne-Kolja Bachstein

Does anyone know where this is actually cached? I am fighting with this one, too. But I can't seem to find the source of it. Cleared everything obvious like caching tables, typo3temp cache folders and so on.

Whenever changing the width and height attribute, the old values are still used when viewing the output.

#13 - 2014-06-11 20:50 - Christian Ludwig

Same here with TYPO3 v6.2.3. Images sometimes appeared in the desired size and sometimes not. Until this is fixed, I am useing the old ImageViewHelper from v4.7 and change the views of my extensions.

Here you can find an extended ViewHelper of v4.7 that can simply be renamed and added to your own extension.

#14 - 2014-06-16 08:58 - Alexander Opitz
- Category set to Frontend
- Status changed from Needs Feedback to New

#15 - 2014-07-09 20:57 - filigivuji filigivuji

Not sure if this is related (if it's not, I can create a new bug if there isn't already one).
I have upgraded TYPO3 4.5 to 6.2.
In one case there's an "images" content element, all its pictures were migrated to fileadmin/_migrated/pics/ by the upgrade script.
The "height" setting in the content element's "Appearance" tab is being ignored, and the <img>'s source attributes just link to fileadmin/_migrated/pics/,
i.e. the pictures have not been rescaled by TYPO3, and the image's height/width attributes are the picture's original size.
When I add a picture from a folder inside fileadmin/ that has its own file mount, the rescaling works properly.

#16 - 2014-07-14 20:54 - Christian Ludwig

I found out that we had lots of entries in table 'sys_file_processedfile' that had an empty 'name', 'width' and 'height'. Deleting those by

```
DELETE FROM `sys_file_processedfile` WHERE `name` IS NULL;
```

solved my problems.

My questions are:

1. Why does the install tool "Clean up" -> "Clear tables" not offer a function to clean image cache sizes and what does "cache_imagesizes" do in FAL?

2. And again the install tool: "Clean up" -> "Clean typo3temp/ folder" -> "_processed_" suggests cleaning exactly what we want but it does not touch files in fileadmin or in the sys_file_processedfile table.

3. Shouldn't the install tool "Important actions" -> "Clear all caches" clean those caches too?

It would be great to have a proper Cache cleaning tool for FAL that wipes out all files in the *processed* folders of fileadmin plus cleaning the database sys_file_processedfile (and maybe cache_imagesizes and cache_typo3temp_log if they are involved in the caching). Or a scheduler task that checks the real sizes and existence of "processed" files.

#17 - 2014-09-01 10:53 - Christian Ludwig

I also noticed this bug when uploading two files with the same name, only differing in case (i. e. "Test.jpg" and "test.JPG"). You receive two results from *sys_file* when searching in field *identifier*.
Maybe this helps identifying the bug.

#18 - 2014-09-02 09:29 - Alexander Opitz

@Christian Ludwig

This sounds like you are on a case-insensitive file system (Windows, MacOS X Default partition) but your storage is configured as case-sensitive.

#19 - 2014-09-02 09:43 - Christian Ludwig

@Alexander
No it is a Linux system and there are two files exiting on the file storage. But this time, when I analysed the problem with image sizes, I noticed that it was a file that existed twice, only differing in case.
I was not aware of the "Uses case sensitive identifiers" setting, so FAL is always case sensitive on db requests and this could not cause the described problem?

#20 - 2014-11-04 19:42 - Romain Seignez

On typo3 6.2.4

I had the same problem from a resource plugin file.
I finally found a record of this file in sys_file table and i fixed size manually in sys_file_metadata.

#21 - 2015-01-06 13:40 - Mathias Schreiber

- Target version set to 7.1 (Cleanup)
- Sprint Focus set to On Location Sprint

#22 - 2015-01-11 12:03 - Frans Saris
- Category changed from Frontend to File Abstraction Layer (FAL)

#23 - 2015-03-24 23:01 - Markus Klein
- Status changed from New to Needs Feedback
- Target version deleted (7.1 (Cleanup))
- Sprint Focus deleted (On Location Sprint)

Is this still an issue on 6.2.11+?

#24 - 2015-03-31 10:26 - Thomas Sperling

Markus Klein wrote:

> *Is this still an issue on 6.2.11+?*

I have updated to 6.2.11 and will give feedback next week.

EDIT: I can confirm: the bug still exist. And the

    DELETE FROM `sys_file_processedfile` WHERE `name` IS NULL;

does not work for me anymore (but it is not reason of 6.2.11).

#25 - 2015-04-02 15:58 - Grégory Duchesnes

I had the exact same problem after an upgrade from 4.5 to 6.1 and then 6.2

the problem was that im_path, im_path_lzw and im_version_5 were not set in the LocalConfiguration.php file since they were included from an external file in my 4.5 install.

Therefore images were rendered in their original size.

Sounds stupid but worth checking anyway ;-)

#26 - 2015-04-09 08:33 - Thomas Sperling

Grégory Duchesnes wrote:

> the problem was that im_path, im_path_lzw and im_version_5 were not set in the LocalConfiguration.php file

that was not my problem but it lead me to the right place. There was an issue with ImageMagick: in the error.log I had a lot of lcms: Error #12288; Pointer error; probably corrupted file
So I switched from im6 to gm and voilà, it worked. This solved my problem that every picture wasn't scaled, but I assume that my old problem with some few unscaled pictures still exist ...

#27 - 2015-04-09 11:59 - Oliver Salzburg

The original issue I reported is not related to IM/GM and it is specific to files located in typo3conf/ext. These files would under no circumstance have their metadata updated in the database. All other files, in fileadmin or any other file mount are not relevant to this issue.

Files in filemounts can have their metadata refreshed, because you can easily clear the metadata cache or update it with a scheduler task. But you need a filemount for that. The same operation does not work for typo3conf/ext. So whatever metadata was recorded when accessing the file for the first time, will be persisted until you manually update the database.

Given the age of this issue, we obviously found ways to work around this behavior soon after FAL was introduced. So it's not like we still have this issue anywhere or could judge whether it still exists. We also don't have a particular interest in it any longer, because we don't rely on FAL here.

#28 - 2015-04-10 22:09 - Frans Saris
- Status changed from Needs Feedback to Closed

Updating the "cached" info for storage 0 will tackled with #64196.

#29 - 2015-06-18 12:48 - Michael Ganner

I ran into the same problem. Executing the following scheduler task solved it: TYPO3\CMS\Scheduler\Task\FileStorageIndexingTask