Core - Bug # 44105

Status:	Closed	**Priority:**	Must have
Author:	Camelia M	**Category:**	File Abstraction Layer (FAL)
Created:	2012-12-19	**Assigned To:**	
Updated:	2015-06-11	**Due date:**	
TYPO3 Version:	6.0		
PHP Version:	5.3		
Complexity:			
Is Regression:	No		
Sprint Focus:			
Subject:	Image size does not get updated		

Description

My test scenario is like this: I upload an image and use it in a 'images only' content element. Then I go in fileadmin where file was originally placed and I want to change the image. So I upload a different one with same name and mark the 'override files'. The new image is uploaded and replaces the old one, but in frontend (as in database table sys_file) are still used old sizes (from the original image).

Related issues:

related to Core - Task # 44550: Preview image and database entry not deleted ...	**Closed**	**2013-01-15**
related to Core - Bug # 44645: Preview images don't get a new filename after ...	**Resolved**	**2013-01-18**
related to Core - Bug # 45922: image replacement, width and height are kept e...	**Accepted**	**2013-02-28**
related to Core - Bug # 46020: Image size is 0 when not scaled	**Closed**	**2013-03-04**
related to Core - Bug # 61181: FAL: file maxW and maxH are ignored	**Rejected**	**2014-08-25**
related to Core - Bug # 50508: Re-uploading file in backend fails	**Resolved**	**2013-07-29**
related to Core - Bug # 52658: Overriding image does not change the dimensions	**Rejected**	**2013-10-10**
related to Core - Bug # 54533: imgResource.noScale doesn't scale image tag	**Rejected**	
related to Core - Bug # 62400: Lot of entries in sys_file_processed with name...	**Rejected**	**2014-10-22**
related to Core - Bug # 59216: Image dimensions (width/height) are 0 when not...	**Closed**	**2014-05-30**
related to FAL Indexing - Bug # 60215: sys_file_processedfile with empty iden...	**Under Review**	**2014-07-09**
related to Core - Bug # 62247: FAL: Uppercase File Extensions of Image cannot...	**Rejected**	**2014-10-15**
related to Core - Bug # 50392: Specifying size in ImageViewHelper does nothing	**Closed**	**2013-07-24**
related to Core - Bug # 57565: Creating new sys_file_metadata record for exis...	**Closed**	**2014-04-02**
related to Core - Bug # 58019: FAL Indexer for broken files: Column 'width' c...	**Rejected**	**2014-04-17**
duplicates Core - Bug # 46446: sys_file doesn't get updated	**Needs Feedback**	**2013-03-20**

History

#1 - 2012-12-19 16:20 - Camelia M

Also the problem is quite severe. Because even if I delete the image and upload again, it seems it keeps some kind of connection so that it still uses old image sizes. Tried to delete it even from the recycler but that didn't help either.

#2 - 2013-01-30 10:08 - Camelia M

I still have this issue even after updating to 6.0.1

#3 - 2013-01-30 10:38 - Camelia M

- File fal-error.png added

you can see in the attachment the image size is 231x185 but the img tag uses old sizes (238x187) thus stretching my picture.

#4 - 2013-02-01 16:57 - Alexander Opitz

Maybe this problem have to do with #44585

Can you try the patch from #44585 and report if that fixes your issue?

#5 - 2013-02-04 08:57 - Camelia M

Alexander Opitz wrote:

> *Maybe this problem have to do with #44585*
>
> *Can you try the patch from #44585 and report if that fixes your issue?*

I tried it but it doesn't fix my problem. Still width and height in sys_file do not get updated. I tried upload with 'override' mark set, also tried deleting and uploading again.. Still old sizes are shown. In 'view item' in backend though all information is correct (not sure if this comes from db or is automatically generated).

#6 - 2013-02-05 12:43 - Alexander Opitz

Ok, do you have 1 ore multiple database entries for that image?

#7 - 2013-02-05 13:10 - Camelia M

Alexander Opitz wrote:

> *Ok, do you have 1 ore multiple database entries for that image?*

In table sys_file i think they are all be unique (although I'm not completely sure that the same image can't be in different folders in fileadmin). But in table sys_file_processedfile I have several db rows that point to same 'original' file (even tough several are different versions of the original, some actually have nothing in common with it and therefore should not point to it). For example for one 'original I found 8 rows. Part of the 8 are different 'cropscales' of the original, while 2 of them are different images.

#8 - 2013-02-09 17:06 - Andreas Wolf

Camelia M wrote:

> *Alexander Opitz wrote:*
>
> > *Ok, do you have 1 ore multiple database entries for that image?*
>
> *In table sys_file i think they are all be unique (although I'm not completely sure that the same image can't be in different folders in fileadmin). But in table sys_file_processedfile I have several db rows that point to same 'original' file (even tough several are different versions of the original, some actually have nothing in common with it and therefore should not point to it). For example for one 'original I found 8 rows. Part of the 8 are different*

> *'cropscales' of the original, while 2 of them are different images.*

I think this is related to bug #45308, not #44585.

#9 - 2013-02-09 17:08 - Andreas Wolf

Andreas Wolf wrote:

> *Camelia M wrote:*
>
> > *Alexander Opitz wrote:*
> >
> > > *Ok, do you have 1 ore multiple database entries for that image?*
> >
> > *In table sys_file i think they are all be unique (although I'm not completely sure that the same image can't be in different folders in fileadmin). But in table sys_file_processedfile I have several db rows that point to same 'original' file (even tough several are different versions of the original, some actually have nothing in common with it and therefore should not point to it). For example for one 'original I found 8 rows. Part of the 8 are different 'cropscales' of the original, while 2 of them are different images.*
>
> *I think this is related to bug #45308, not #44585.*

Ah, and I think the problem Camelia described originally is indeed related to #44585. Could you check if your problems are solved with most recent master? Please empty sys_file_processedfile before testing; this will remove all cached image versions.

#10 - 2013-02-09 17:08 - Andreas Wolf
- Status changed from New to Needs Feedback

#11 - 2013-02-11 08:21 - Camelia M

I did apply patch from #44585 and it fixes the problem reported by bug #44585 (0 filesize shows in filelist) but it does not fix the image dimension problem (not filesize but width and height).
I also applied patch from #45308 but this doesn't fix my problem either.
I did empty sys_file_processedfile after each patch but it seems to me that the sys_file is the problem, not sys_file_processedfile because sys_file keeps the original width and height of the image (no matter if the newly updated image has different sizes).
So the originally reported issue still stands (ignore the post about several rows in sys_file_processedfile pointing to same original in sys_file as this was treated in another issue #44616 and I think is not related to this one).
But it is related (if not duplicated) by #44550 and that was marked as accepted, so I guess when that gets fixed, this will be fixed also ;)

#12 - 2013-03-28 14:08 - Camelia M

Although I now use typo3 6.0.4, I still have this problem.
The status for this bug report shows as "Needs feedback". Is there any other info I should provide?

#13 - 2013-06-13 13:42 - Matthias Secker

I got the same problem as well in TYPO3 6.1.1
I uploaded logo.png (FTP) and used IMAGE cObject in TS to include that image. Later I saw that my image was a bit too small in height and width so I

replaced the file with a new one, the same name (FTP overwrite).
HTML output show that the new image is used but IMAGE cObject created the old height/width parameters. I found the entry "logo.png" in sys_file twice.

#14 - 2013-07-09 19:56 - Ralf Hettinger

Issue still exists in 6.0.6:
In table sys_file, fields width and height are not updated when the file itself is overwritten by upload.

#15 - 2013-07-22 19:42 - Oliver Salzburg

I just tripped over this in 6.1.1. I was trying to replace a logo in a website and the new image always had the wrong size.

The file is just kept in an extension folder and was only updated through FTP. I cleared the image size cache and any other cache I could think of, but to no avail.

Is this to be expected? What is a possible workaround?

#16 - 2013-07-23 09:45 - Marcel Burkhalter

@Oliver: For FAL to be able to detect changes done via FTP you should add the job "File Abstraction Layer: Indexing job" to your TYPO3 scheduler and let it run periodically. You can also trigger it manually in the scheduler, clear all caches and check if your problem is fixed.

If not, the following workaround did it for us:

1.)
Check the generated filename of the image with the wrong size in the FE (e.g. "csm_kontakt2_b26c111e24.jpg")

2.)
Search for this file with phpMyAdmin in the processed file table:

```
SELECT *
FROM `sys_file_processedfile`
WHERE `name` LIKE '%csm_kontakt2_b26c111e24.jpg%'
```

3.)
Look out for the "originalfilesha1" of the found file (e.g. "0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723")

4.)
Search for the original file with phpMyAdmin:

```
SELECT *
FROM `sys_file`
WHERE `sha1` = '0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723'
```

5.)
Manually delete the wrong entry/entries. In our case the "storage" value was 0 (which is invalid, storage IDs start with 1). And the wrong entries had the old height/width values.

Hope this helps.

#17 - 2013-07-23 10:40 - Camelia M

Marcel Burkhalter wrote:

> @Oliver: For FAL to be able to detect changes done via FTP you should add the job "File Abstraction Layer: Indexing job" to your TYPO3 scheduler and let it run periodically. You can also trigger it manually in the scheduler, clear all caches and check if your problem is fixed.
>
> If not, the following workaround did it for us:
>
> 1.)
> Check the generated filename of the image with the wrong size in the FE (e.g. "csm_kontakt2_b26c111e24.jpg")
>
> 2.)
> Search for this file with phpMyAdmin in the processed file table:
> SELECT *
> FROM `sys_file_processedfile`
> WHERE `name` LIKE '%csm_kontakt2_b26c111e24.jpg%'
>
> 3.)
> Look out for the "originalfilesha1" of the found file (e.g. "0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723")
>
> 4.)
> Search for the original file with phpMyAdmin:
> SELECT *
> FROM `sys_file`
> WHERE `sha1` = '0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723'
>
> 5.)
> Manually delete the wrong entry/entries. In our case the "storage" value was 0 (which is invalid, storage IDs start with 1). And the wrong entries had the old height/width values.
>
> Hope this helps.

In my case the problem is there also if I replace it in the backend by using filelist->upload with the checkmark "overrride existing filelist". My guess is that problem has nothing to do with the method of uploading the file.

#18 - 2013-07-24 14:59 - Oliver Salzburg

Marcel Burkhalter wrote:

> @Oliver: For FAL to be able to detect changes done via FTP you should add the job "File Abstraction Layer: Indexing job" to your TYPO3 scheduler and let it run periodically. You can also trigger it manually in the scheduler, clear all caches and check if your problem is fixed.
>
> If not, the following workaround did it for us:
>
> 1.)
> Check the generated filename of the image with the wrong size in the FE (e.g. "csm_kontakt2_b26c111e24.jpg")
>
> 2.)
> Search for this file with phpMyAdmin in the processed file table:
> SELECT *

> *FROM `sys_file_processedfile`*
> *WHERE `name` LIKE '%csm_kontakt2_b26c111e24.jpg%'*
>
> *3.)*
> *Look out for the "originalfilesha1" of the found file (e.g. "0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723")*
>
> *4.)*
> *Search for the original file with phpMyAdmin:*
> *SELECT **
> *FROM `sys_file`*
> *WHERE `sha1` = '0526f87243d4b9dd5e2e2e8eec535f1b6fe8b723'*
>
> *5.)*
> *Manually delete the wrong entry/entries. In our case the "storage" value was 0 (which is invalid, storage IDs start with 1). And the wrong entries had the old height/width values.*
>
> *Hope this helps.*

Thanks for the advice. Sadly, the scheduler task doesn't help, because it only indexes registered storage locations (like fileadmin) and our extension that defines the website is not (and should probably not) be configured as a storage location.

I'm aware of being able to manipulate the data in the database directly, but would obviously like to avoid this.

Upon further inspection, not only is the cached size incorrect, the stored SHA1 checksum, the file size (and probably the cdate and mdate) are also incorrect. I currently don't see how I could get the data properly updated. Even deactivating and reactivating doesn't cause the entry to be updated.

#19 - 2013-07-24 15:25 - Marcel Burkhalter

> *I'm aware of being able to manipulate the data in the database directly, but would obviously like to avoid this.*
>
> *Upon further inspection, not only is the cached size incorrect, the stored SHA1 checksum, the file size (and probably the cdate and mdate) are also incorrect. I currently don't see how I could get the data properly updated. Even deactivating and reactivating doesn't cause the entry to be updated.*

AFAIK there is currently no other workaround than manipulating in the database until this issue has been fixed :(

#20 - 2013-07-24 16:01 - Oliver Salzburg

Marcel Burkhalter wrote:

> *AFAIK there is currently no other workaround than manipulating in the database until this issue has been fixed :(*

Oh well...

Just FYI, setting up a storage location for the website extension (in the hope to be able to utilize that scheduler task) doesn't work either. The files will be identified with their relative name (to the storage location). So instead of getting /typo3conf/ext/user_website/Resources/logo.png I get /Resources/logo.png. So it doesn't even affect my file in question.

#21 - 2013-07-30 12:33 - Philipp Gampe

This is an issue with extension updates. Any changed ressources are **not** detected. IMHO this is a major issue of FAL.

#22 - 2013-09-16 10:54 - Bjoern Jacob

We are experiencing the same problem. Some of our customers use heavily SFTP for uploading pictures. I also think this is a major problem. The status of this ticket is "Needs Feedback". So what kind of feedback do you need to fix this problem?

#23 - 2013-09-18 11:38 - Philipp Gampe

I can't change it, please ask one of the members to move this ticket to core.

Anyway you need to run the indexer job after sftp upload. Or you need to trigger this as a post-upload script if your sFTP server supports this.

#24 - 2013-09-18 11:48 - Steffen Ritter

- Project changed from File Abstraction Layer to Core
- Target version deleted (6.1)

#25 - 2013-09-18 16:38 - Bjoern Jacob

I haven't heard of this indexer but a short while after I updated this post I found it. So I've set up the FileIndexingTask. But as soon as I started it I got the next error message saying "You are not allowed to access the given folder". I don't know where the problem is discussed. Does it belong to #51747? Anyway, we found a working solution. We've copied the FileIndexingTask and added one more line within the foreach loop.

```
    $storageObject->setEvaluatePermissions(FALSE);
```

Do you know the correct corresponding ticket? Than I can send a patch.

#26 - 2013-09-19 09:37 - Alexander Opitz

- Category set to File Abstraction Layer (FAL)
- Is Regression set to No
- TYPO3 Version set to 6.0

If you have TYPO3 CMS 6.1 lower then 6.1.5 (or TYPO3 CMS 6.0 lower then 6.0.10) then update and try again.

Else, please create a new issue report about the problem with the indexer task.

#27 - 2013-09-19 11:18 - Oliver Salzburg

Just FYI, I updated to 6.1.5 and tried the indexer task again. It has no effect on cached data related to files stored in extension directories.

#28 - 2013-09-19 14:09 - Philipp Gampe

The indexer just updates files inside a storage ... this tip was related to sFTP upload for editors which likely appear within at least on storage+mountpoint.

#29 - 2013-09-19 14:09 - Philipp Gampe

- Priority changed from Should have to Must have
- Target version set to 7.0
- PHP Version set to 5.3

#30 - 2013-10-15 10:47 - Thorsten Kahler

- Target version changed from 7.0 to 6.2.0

Should be fixed in 6.2 IMO.

#31 - 2013-11-21 17:21 - Alexander Opitz

Please test 6.2beta2 if this issue is gone for your installation. Please try it with newly changed images.

#32 - 2014-02-27 09:11 - Thomas Etter

I testet this with TYPO3 6.1.7 and it seems to be solved here, the image is always resized when i upload another image and overwrite the existing one both in the backend via filelist or directly. However i have a related issue with fluid, where the image size is not updated, when i change the size in the template (without changing the image).

#33 - 2014-04-02 20:29 - Markus Klein

- Status changed from Needs Feedback to Closed
- Target version deleted (6.2.0)

Closing this issue as it seems to be fixed in 6.2

#34 - 2014-09-18 09:16 - Markus D

I still have the described error in version 6.2.4

the image size is not updated

i've cleared the cache in the install tool, typo3temp and all caches in be.

#35 - 2014-11-20 12:13 - Andreas Allacher

I just noticed this issue again too with 6.2.6
I updated an image inside my extension but the sys_file record still used the old image size and therefore the image was rendered wrongly (not even ratio was correct) and in this case the image was not even resized but just rendered directly.

I was using the ImageViewHelper for the output.
I can avoid this by using uri.resource and the tag directly but that way I am losing resize functionality etc.

Maybe checking the if the original file timestamp matches the current timestamp and if not check the SHA1 value before rendering?

#36 - 2015-05-02 10:40 - Bastian Stargazer

This bug still exists in the version 6.2.11! My (ugly) workaround is to set a number suffix to all my images which I upload to fileadmin/ manually. For example: "content-fill-img_house01.jpg" will be "content-fill-img_house02.jpg" if I changed the size...

To delete the caches or the files in fileadmin/_processed_/ doesn´t help!

#37 - 2015-06-10 19:41 - Carsten Tornow

That problem is still there! Using TYPO3 v. 6.2.12

#38 - 2015-06-11 09:35 - Benjamin Mack

Could you recheck with 6.2.13 please?

Files

fal-error.png	7.3 kB	2013-01-30	Camelia M