

## Core - Bug # 46020

<b>Status:</b>	Closed	<b>Priority:</b>	Must have
<b>Author:</b>	Freddy Kühne	<b>Category:</b>	File Abstraction Layer (FAL)
<b>Created:</b>	2013-03-04	<b>Assigned To:</b>	Alexander Opitz
<b>Updated:</b>	2015-04-30	<b>Due date:</b>	
<b>TYPO3 Version:</b>	6.0		
<b>PHP Version:</b>	5.3		
<b>Complexity:</b>	hard		
<b>Is Regression:</b>	Yes		
<b>Sprint Focus:</b>			
<b>Subject:</b>	Image size is 0 when not scaled		
<b>Description</b>			
<p>When a "Text &amp; Images"-element with an unscaled image is rendered, then the width and height attributes are "0":</p> <pre>&lt;img src="fileadmin/_migrated/pics/gold250.png" width="0" height="0" border="0" alt="" /&gt;</pre> <p>If the image is manipulated in any way (width, height, quality...) then the dimensions are generated correct.</p> <p>Additional infos:</p> <ul style="list-style-type: none"><li>- installed version 6.0.2</li><li>- updated installation from 4.7</li><li>- "Install Tool / Clean up / Clear cached image sizes" executed</li><li>- FE cache cleared</li></ul>			
<b>Related issues:</b>			
related to Core - Bug # 45168: Wrong image displayed in FE if not resized		<b>Resolved</b>	<b>2013-02-05</b>
related to Core - Task # 46777: Improve behaviour when mime type detection is...		<b>Closed</b>	<b>2013-03-29</b>
related to Core - Bug # 57492: Upgrading 6.2: Do not run FileTableSplittingUp...		<b>Resolved</b>	<b>2014-04-01</b>
related to Core - Bug # 57495: Metadata db insert fails due to NULL value		<b>Resolved</b>	<b>2014-04-01</b>
related to Core - Bug # 59216: Image dimensions (width/height) are 0 when not...		<b>Closed</b>	<b>2014-05-30</b>
related to Core - Task # 57545: Missing metadata entry impossible to regenerate.		<b>Resolved</b>	<b>2014-04-02</b>
related to Core - Bug # 61181: FAL: file maxW and maxH are ignored		<b>Rejected</b>	<b>2014-08-25</b>
related to Core - Bug # 62400: Lot of entries in sys_file_processed with name...		<b>Rejected</b>	<b>2014-10-22</b>
related to Core - Bug # 44105: Image size does not get updated		<b>Closed</b>	<b>2012-12-19</b>
related to Core - Bug # 63634: FAL update Metadata in LocalStorage not workin...		<b>Closed</b>	<b>2014-12-06</b>

### Associated revisions

**Revision 9f988d68 - 2013-11-09 16:43 - Markus Klein**

[BUGFIX] FAL: Always load meta data of files

There's one place in the File class where the  
according meta data were not loaded.

Resolves: #46020

Releases: 6.2

Change-Id: I41ccb6abcf8abf47a3a5f16875203f2982872f1e

Reviewed-on: <https://review.typo3.org/25187>

Reviewed-by: Wouter Wolters

Tested-by: Martin Holtz

Reviewed-by: Steffen Ritter

Tested-by: Steffen Ritter

#### **Revision 38f484e0 - 2014-04-02 12:18 - Markus Klein**

[BUGFIX] FAL: Prevent invalid NULL value on metadata creation

The cruser\_id field is now guaranteed to be an integer.  
This patch checks if a BE\_USER is available and uses zero otherwise.

Resolves: #57495

Related: #46020

Releases: 6.2

Change-Id: I8260604eb30ca32d2a5e079fa6f5b10a6d04714e

Reviewed-on: <https://review.typo3.org/29028>

Reviewed-by: Nicole Cordes

Reviewed-by: Jigal van Hemert

Reviewed-by: Frans Saris

Reviewed-by: Stefan Neufeind

Tested-by: Stefan Neufeind

Reviewed-by: Alexander Opitz

Tested-by: Alexander Opitz

Reviewed-by: Markus Klein

Tested-by: Markus Klein

#### **Revision a5d16741 - 2014-05-04 14:15 - Alexander Opitz**

[BUGFIX] Limit FileTableSplittingUpdate to TYPO3 6.0/6.1

If the database field does not exist it may be that we have an installation before 6.0 or after 6.1 so we do not need to migrate the data to sys\_file\_metadata. This prevents empty metadata information which may lead to other issues.  
(width/height=0 for images)

Resolves: #57492

Related: #46020

Releases: 6.2

Change-Id: Id5826ea480d4139f2e232587fc2b119928b28bc8

Reviewed-on: <https://review.typo3.org/29024>

Reviewed-by: Markus Klein

Tested-by: Markus Klein

Reviewed-by: Wouter Wolters

Reviewed-by: Nicole Cordes

Reviewed-by: Anja Leichsenring

Reviewed-by: Jan Runte

Reviewed-by: Marc Bastian Heinrichs

Tested-by: Marc Bastian Heinrichs

## **History**

---

**#1 - 2013-03-05 13:54 - Steffen Mächtel**

Same problem here. Width and height is 0 on unprocessed files (use original file).

New rows in sys\_file table has mime\_type = "", width = 0 and height = 0. It looks like the indexing dont work.

**#2 - 2013-03-05 14:28 - Markus Klein**

@Andreas Wolf: Could you please have a look what's going on here?

**#3 - 2013-03-05 23:55 - Stefan Galinski**

- *Category set to File Abstraction Layer (FAL)*

**#4 - 2013-03-11 18:37 - Jens Schmietendorf**

Same problem here.

Locally installed on a Mac with OS X 10.6.8 and MAMP all works fine.

After t3d-export, upload an -import on a shared hosting account at Mittwald  
the image sizes of unprocessed files ar set to width=0 and height=0.

System info:

a fresh TYPO3 installation 6.0.4

Linux ovm2386 2.6.18-prep-pve-mittwald-el5 #5 SMP Mon Jun 22 13:23:53 CEST 2009 i686

PHP Version 5.3.15 CGI/FastCGI

This is very annoying!

**#5 - 2013-03-22 10:50 - Andreas Wolf**

- *Status changed from New to Accepted*

- *Assigned To set to Andreas Wolf*

- *Complexity set to medium*

Could you please check in your sys\_file table if the dimensions are correct there? There are width and height fields, if they contain the correct numbers, then the output is wrong, otherwise we have a problem in the indexer.

I'll try to reproduce this bug on my installation and find a fix for it - should not be too difficult.

**#6 - 2013-03-22 11:59 - Andreas Wolf**

- *Status changed from Accepted to Needs Feedback*

I tried to reproduce it with an introduction package, but that did not work - the images are displayed correctly in the frontend.

If anybody could provide me with a dump (or account on your test installation), I can have a look at your installation. Just drop me an email to andreas.wolf (ät) typo3.org.

**#7 - 2013-03-22 14:51 - Benjamin Mack**

cannot reproduce this issue either with 6.1-dev

**#8 - 2013-03-27 11:29 - Steffen Mächtel**

- File *sys\_file.zip* added

I have attached a dump of table "sys\_file".

On new rows:

mime\_type = "", width = 0 and height = 0

But "size" field is correct. Files are not indexed as image? Do I need a mime\_type apache mod or something ?!

**#9 - 2013-03-27 13:03 - Steffen Mächtel**

After some debugging I found the problem in class "TYPO3\CMS\Core\Resource\Driver\LocalDriver" in method "getMimeTypeOfFile". Both functions "finfo\_file" and "mime\_content\_type" are not available on our server. return is always false.

After hardcoded return "image/jpg" everything works for new files. Width, height and type is set correctly in sys\_file table.

```
protected function getMimeTypeOfFile($absoluteFilePath) {  
    return 'image/jpg'; // DEBUG TEST  
    if (function_exists('finfo_file')) {  
        $fileInfo = new \finfo();  
        return $fileInfo->file($absoluteFilePath, FILEINFO_MIME_TYPE);  
    } elseif (function_exists('mime_content_type')) {  
        return mime_content_type($absoluteFilePath);  
    }  
    return FALSE;  
}
```

Is there an alternative mime\_type check method as fallback?!

**#10 - 2013-03-27 13:20 - Steffen Mächtel**

- File *AdditionalConfiguration.php* added

As hotfix on our server, I put the attached function in AdditionalConfiguration.php (function is from php.net, switch on file extension, better than nothing ^^)

**#11 - 2013-03-28 17:14 - Markus Klein**

Thank you Steffen for your investigations.

I guess a mime type detection should be somewhat available on every server.

I'd suggest to close this issue.

**#12 - 2013-03-29 09:43 - Freddy Kühne**

- File `sys_file.zip` added

Sorry for my late response; I was away for a few days...  
I've attached our `sys_file` table; it's mime-type is mostly empty.

I have also tried the steps in #9 by Steffen; no changes...  
Our mime type functions seems to work correctly.

Testscript:

```
echo function_exists('finfo_file');  
echo function_exists('mime_content_type');  
$fileInfo = new finfo();  
echo $fileInfo->file('fileadmin/_migrated/pics/gold250.png', FILEINFO_MIME_TYPE);
```

--> Output is ok...

It might be, that during the update process the php-extension "php\_fileinfo.dll" was not loaded. We currently can't say, which order exactly was used. I will try it with a fresh update...

**#13 - 2013-03-29 12:30 - Freddy Kühne**

OK, I made two more updates with version 6.0.4 and the images are displayed correctly!  
Seems that we included the php extension too late, or version 6.0.4 has fixed it.  
Thank's for your time...

**#14 - 2013-03-29 18:48 - Andreas Wolf**

- Status changed from *Needs Feedback* to *Resolved*

Markus Klein wrote:

*I guess a mime type detection should be somewhat available on every server.*  
*I'd suggest to close this issue.*

I think we should have a reports check for the availability of mimetype detections, and try to get as much done in a situation where no detection is available. It is unacceptable that file indexing and output are broken (or tend to break at the very least) if the mimetype cannot be detected.

Apart from that, I think this issue is solved, at least that's what I got from the latest replies. If not, please raise your hand and we'll open it again.

**#15 - 2013-03-31 15:59 - Markus Klein**

I just checked the install tool.  
fileinfo is already a required php extension and is checked by the install tool!

**#16 - 2013-05-30 16:48 - Cornel Widmer**

I believe this bug is NOT completely resolved. On multiple fresh TYPO3 6.1 installations it occurs constantly.

Whether if we include an image over fluid (<f:image>) or over a typoscript object (lib.image = IMAGE) its always the same. I've checked the sys\_file-table and there is no mime\_type. But I think this is not the real cause. Because - when I enter the correct mime type by hand - the image does not appear.

We tested many ways to get the image working and this is what we have found out.

- 1) Delete entry in sys\_file-table for the affected file
- 2) Hit refresh on frontend - the image appears
- 3) Take a look in the sys\_file-table - entry created with: storage = 1, mime\_type = EMPTY
- 4) Hit refresh on frontend - the image disappears
- 5) Take a look in the sys\_file-table - a NEW entry created with: storage = 0, mime\_type =s EMPTY

Any ideas for this strange behaviour?

**#17 - 2013-05-30 20:25 - Chris topher**

- Status changed from Resolved to Needs Feedback

**#18 - 2013-06-01 21:41 - Mathias Bolt Lesniak**

Hi!

I can confirm this is an issue with 6.1.1 from my end.

Apparently, the unprocessed image's dimensions are never fetched by FAL, and thus a width and height of zero is stored. This happens in TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer.php, lines 5191 and 5192.

The quick fix would be to edit cImage(), replacing the line

```
$theValue = 'getBorderAttr(' border="' . intval($conf['border']) . '"') . $params . $altParam . (empty($GLOBALS['TSFE']->htmlDoctype) ? ' / ' : ' ') . '>';
```

with:

```
if($info[0] == 0 || $info[1] == 0) {
    $theValue = 'getBorderAttr(' border="' . intval($conf['border']) . '"') . $params . $altParam . (empty($GLOBALS['TSFE']->htmlDoctype) ? ' / ' : ' ') . '>';
} else {
    $theValue = 'getBorderAttr(' border="' . intval($conf['border']) . '"') . $params . $altParam . (empty($GLOBALS['TSFE']->htmlDoctype) ? ' / ' : ' ') . '>';
}
```

That just removes the width and height attributes from the img tag. No long term solution.

**#19 - 2013-06-03 09:33 - Riccardo De Contardi**

This is my personal opinion: why not removing width & height attributes at all?

**#20 - 2013-06-03 09:51 - Mathias Bolt Lesniak**

Riccardo De Contardi wrote:

| *This is my personal opinion: why not removing width & height attributes at all?*

The image width and height may be important for the design. In cases where users have turned off image rendering in their browser, the width and height attributes will define a box of the size the image would have had. This was more important before, but it's still used a bit in email newsletters, for example. :-)

#### #21 - 2013-06-04 01:35 - Sven Tappert

- *File LocalDriver.php added*

I think I found a first solution (tested in version 6.1.1, but probably works for 6.0 as well).

Reason: image width, height and (mime)type are never set to the "File" object (when the image doesn't need to be modified). So incomplete file information is written to table "sys\_file".

I'm not sure if it's the right place, but it works fine, when patching:

\TYPO3\CMS\Core\Resource\Driver\LocalDriver

(typo3/sysex/core/Classes/Resource/Driver/LocalDriver.php)

In method "extractFileInformation()" I added after line 457:

```
// get additional information for image files
$imageInfo = getimagesize($filePath);
if (is_array($imageInfo))
{
    $fileInformation['width'] = $imageInfo[0];
    $fileInformation['height'] = $imageInfo[1];
    if (!$fileInformation['mimetype'] && !empty($imageInfo['mime'])) {
        $fileInformation['mimetype'] = $imageInfo['mime'];
    }
}
```

The complete file is attached.

IMPORTANT: you have to delete the entries in table "sys\_file" to refresh the file information. Additional file information (like title & description) will get lost! You can try to identify affected entries with this SQL:

```
SELECT * FROM `sys_file` WHERE `type`=0 AND mime_type="" AND `width`=0 AND `height`=0;
```

p.s. the reason why the image shows up on the first time is, that at the first run the width and height values are casted to an empty string, while they are stored & read as integer (zero) to the database.

#### #22 - 2013-06-04 09:20 - Markus Klein

- *Status changed from Needs Feedback to Accepted*

Hi Sven,

would you mind pushing your suggested change to our review system?

You can find all the need information here: [http://wiki.typo3.org/Category:Contribution Walkthrough](http://wiki.typo3.org/Category:Contribution_Walkthrough)

Let me know, if you need any help.

**#23 - 2013-06-04 23:21 - Sven Tappert**

Markus Klein wrote:

*would you mind pushing your suggested change to our review system?*

It seems that the problem is solved in current 6.2-dev on another way (I have not found the code that makes the difference). So it's probably not useful to push my solution into the review system.

Since it's not clear when the official solution is backported to 6.1, one might use my patch in the meantime. It seems to work properly for the moment, but be careful - I have only tested it on a very small site.

**#24 - 2013-06-04 23:31 - Markus Klein**

@Sven: Can your tell us, where you found that code, so that we can at least set a relation to the according issue here?

**#25 - 2013-06-05 00:09 - Sven Tappert**

@Markus: Sorry, unfortunately I could not find the code. It seems to behave correct in current 6.2-dev and it's a different solution than mine..

**#26 - 2013-06-05 02:55 - Sven Tappert**

On my local system the error was caused by the missing PHP "fileinfo" module (disabled in php.ini).  
(Check Typo3 Installer > "System environment" or phpinfo() for "fileinfo").

However, on the live system the "fileinfo" was installed from the beginning and the error did occur as well. But for now the images are rendered correctly, without any patches. Maybe there is another cause of failure? Or am I just confused?

@Markus: It all depends on `\TYPO3\CMS\Core\Resource\Driver\LocalDriver::getMimeTypeOfFile()`. If the mime type is not detected here, the file is not considered an image and width & height are not read in `\TYPO3\CMS\Core\Resource\Service\IndexerService::indexFile()`

**#27 - 2013-11-06 17:36 - Martin Holtz**

Hi,

i have this issue in TYPO3 6.2 Beta right now.

20 = IMAGE

20.file = fileadmin/helloworld.gif



Creates

```

```

In IE8 the images will be rendered with height="1" and width="1" - it seems as they just disappear.

With 6.2 Beta there is no width and height in sys\_file anymore. But the data in sys\_file\_metadata are correct. Width and height are set.

Sidenote: TSFE:lastImageInfo|0 and TSFE:lastImageInfo|1 are empty at that point.

**#28 - 2013-11-06 17:53 - Markus Klein**

- *Is Regression set to No*

@Martin: I have a few questions:

- How did you upload the image? Via FTP or via Backend Filelist?
- fileinfo module is available at your server?

**#29 - 2013-11-06 17:58 - Markus Klein**

- *Assigned To deleted (Andreas Wolf)*

- *Target version set to next-patchlevel*

Markus Klein wrote:

| @Martin: I have a few questions: - How did you upload the image? Via FTP or via Backend Filelist?

Ok that does not matter obviously....

**#30 - 2013-11-06 18:17 - Martin Holtz**

In that case, \$processedFileObject does not have the property "width" so, \$processedFileObject->getProperty('width') returns NULL.

**#31 - 2013-11-06 19:46 - Markus Klein**

It does not need to have this property, as it is tried to retrieve it from the original file and there from the meta data.

But the meta data for the original file is never populated.

**#32 - 2013-11-06 20:00 - Markus Klein**

- *Complexity changed from medium to hard*

- *Is Regression changed from No to Yes*

This is a regression to the meta data table split.

There's one case, where the meta data were NOT loaded, hence the dimension info was missing.

**#33 - 2013-11-06 20:00 - Gerrit Code Review**

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/25187>

#### #34 - 2013-11-06 20:05 - Gerrit Code Review

Patch set 2 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/25187>

#### #35 - 2013-11-09 17:30 - Markus Klein

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:9f988d6818064737473677dc2619c9e679fd315f.

#### #36 - 2013-12-12 18:44 - Heiko Kromm

I thik it's not fixed.

I can reproduce this on 6.1.5, 6.1.6, 6.1.7

#### #37 - 2013-12-12 19:31 - Heiko Kromm

Truncating all sys\_file... tables solves the problem for me.

Perhaps something went wrong while updating

#### #38 - 2013-12-12 20:51 - Mathias Bolt Lesniak

- File *imagesize.php* added

Heiko Kromm wrote:

Truncating all sys\_file... tables solves the problem for me.  
Perhaps something went wrong while updating

I am also rid of the problem in the newest versoin of 6.1.

It's important to note that truncating sys\_file\* tables won't work if you're upgrading a site and have gone through the migration into FAL. That would mess up the references. In these cases it's better to insert all the image widths and heights manually (and clear the cache). I ran into this issue with a site just recently, so I wrote a script that does it for you. (Attached here.) It runs from within typo3conf/.

- Mathias

#### #39 - 2014-01-12 11:39 - Tobias Liegl

The problem with no images sizes (width, height) in the database happened to me also. I did an upgrade from 4.5 to latest 6.2.master today. I fixed it like Mathias by manually updating the info in the db.

#### #40 - 2014-01-26 21:19 - Marcus Schwemer

Same here ... I am using plain TS

```
temp.rightBottom {  
  
    5 = IMAGE  
    5 {  
        wrap = <div class="right_bottom_content"> | <br /><span class="right_bottom_image_text">Certified TYPO3 Integrator</span></div>  
        file = fileadmin/config/de.schwemer.www/images/Cert_button_02_black_colored.png  
        imageLinkWrap = 1  
        imageLinkWrap {  
            enable = 1  
            typolink {  
                parameter = 135  
                title = Certified TYPO3 Integrator  
            }  
        }  
    }  
}
```

HTML - Output looks like this:

```

```

After adding the correct height and width to sys\_file\_metadata the image is shown.

#### #41 - 2014-01-26 21:27 - Marcus Schwemer

Another observation:

Both files exist twice in "sys\_file". Once with zero width and height and once with the correct values.

#### #42 - 2014-01-26 22:06 - Markus Klein

- Status changed from Resolved to Accepted

#### #43 - 2014-01-26 22:18 - Markus Klein

Reopened this as it obviously needs some way to fix this in 6.0 and 6.1.

#### #44 - 2014-01-29 12:15 - Philipp Wrann

Tobias Liegl wrote:

|

2015-08-04

11/24

*The problem with no images sizes (width, height) in the database happened to me also. I did an upgrade from 4.5 to latest 6.2.master today. I fixed it like Mathias by manually updating the info in the db.*

Same here in 6.2

It happend with a conversion. When i upgraded to typo3 6.2 i didnt run the update tool but loaded the frontend, so one image was rendered, therefore a metadata entity was createt but the file was not resized before.

When creating the metadata using the update tool the problem does not seem to be present. When uploading a new file, relating it and displaying it (without resizing) everything is fine too.

So when creating the metadata on-the-fly the width/height properties are not set correct because the processedFile was not really processed (no resizing took place).

#### **#45 - 2014-02-09 20:06 - Bernd Sitzmann**

I've installed a clean test-system Typo3 6.1.7 within XAMPP and created a new database and a new site and found the same problem as described aboth.

For ex. pictures added in the content element "text with pictures" are not shown, when not resized.

In sys\_file width and height is 0.

I removed the picture added it again, put a width to the picture settings which is different from the original size, the picture shows up.

#### **#46 - 2014-02-14 16:10 - Christian Reiter**

I found this phenomenon in the folowing situation: - Site was migrated from 4.5 to 6.1 with DAM->FAL

- Affected images were not the migrated DAM ones but new ones, it is not a problem with the migration
- In a tt\_news record with 3 FAL references (using fal connector ext. for tt\_news) 2 showed up normally (scaled as defined by Typoscript), 1 got width=0, occasionally this also happened with regular textpic elements
  - expected width set by TS was 200
  - Other images got scaled t fileadmin/\_processed\_, this one not
- I tried just downloading the affected image and uploading it again with overwrite (binary identical file) => still no correct rendering. However suddenly I got the native width of the image output (width=1200) instead of 200
  - still no scaling to *processed*
- Then I examined the image for any possible corruption. It seemed OK, taken on a semi-professional DSLR and passwd through Camera Raw and Photoshop CS 5 on Macintosh. Lots of EXIF and embedded metainformation but none of it malformed as far as any graphics format interpreter said. It was 8-bit RGB. (some processing likes to fail on 16bit per channel or CMYK files...)
- Then I tried exporting the image through "Save for Web" both in Adobe PS and Irfanview
- Again upload with overwrite => now results were correct.

It seems the process of recognizing images is thus possibly very sensitive to perhaps unexpected meta-information in the JPEG?

However DSLR -> Camera Raw/Photoshop would seem like a regular and expectable workflow...

#### **#47 - 2014-02-14 16:48 - Markus Klein**

@Christian: Wow impressive testing. Thank you.

The exported for web images did not contain any EXIF?

**#48 - 2014-02-20 15:36 - Christian Reiter**

@Markus:

The exported-for-web image had as the only EXIF tag "Filename - xxxxx.jpg"

The original had a full list, with 60 parameters

ImageWidth - 3568  
ImageLength - 2368  
BitsPerSample - 8 8 8  
PhotometricInterpretation - 2  
ImageDescription - SONY DSC  
Make - SONY  
Model - DSLR-A580  
Orientation - Top left  
SamplesPerPixel - 3  
XResolution - 350.00  
YResolution - 350.00  
ResolutionUnit - Inch  
Software - Adobe Photoshop CS5.1 Macintosh

etc etc.

**#49 - 2014-02-21 13:51 - Markus Klein**

I now double checked that the Core is never reading the EXIF info directly (except indexed\_search).

So suspect that there's either a PHP bug or a gm/im bug that fails reading the files properly.

**#50 - 2014-03-14 14:06 - Christian Reiter**

I'm seeing this problem show up now also in TYPO3 6.2 migrations.

Width + height are missing in sys\_file\_metadata, mime\_type in sys\_file is OK.

Testing in the frontend shows some number of "missing" tt\_content images in the client's browser

Source code shows they are there with width, height=0

If I note the affected page IDs and query the DB with

```
select r.pid as page_id, f.uid as file_id, r.uid as ref_id, f.identifier, f.size, f.mime_type, m.width, m.height from sys_file f, sys_file_metadata m, sys_file_reference r where r.uid_local=f.uid and m.file=f.uid and m.width=0 order by r.pid limit 0,999
```

then I can see that all the pages with these problems contain the relations to files that have empty width/height in sys\_file\_metadata.

1. Updating metadata in the backend and saving does not help
2. Running the indexer from Scheduler again does not help
3. However, **renaming** the file in the backend can solve it - width shows up

- So there is not some property of the image content that keeps the width/height from being read. Otherwise renaming couldn't help
- It just seems that with a random/low probability the reading of dimensions fails.
- There is no special property I can identify, i.e. it is not only large images that are affected.

In the current project about 2% of images were affected out of ca 12.000 sys\_file records.

since they are randomly distributed it would be a real waste of time to reupload all of them and you need to be website admin to do it, since you have to query the DB directly to get the list.

**Suggestion:**

*Maybe we could have some kind of task to check and correct this problem since I see the risk that it will hit many installations.*

*The task would simply check if any images stored in sys\_file have missing width/height entries and try to reread their dimensions.*

*Maybe in install tool or as a scheduler task.*

**#51 - 2014-03-16 17:41 - Frans Saris**

Maybe we can add the storage local && filetype Image && width 0 && height 0 as extra reindex trigger to the indexer just as modification date already is.

<https://github.com/TYPO3/TYPO3.CMS/blob/master/typo3/sysext/core/Classes/Resource/Index/Indexer.php#L165>

**#52 - 2014-03-19 15:33 - Oliver Wand**

Stumbled upon the exact same problem within a fresh 6.1.7 installation. Strangely enough the problem here only occurs when using S3 as media storage. Locally stored images are rendered correctly.

**#53 - 2014-03-26 13:07 - Marc Reiter**

- File *imagesize\_6\_2.php* added

We've modified the updatescript *imagesize.php* for typo3 6.2 version.

The mime type is not updated it seem o.k. for me in my table.

Thanks a lot to Mathias for the initial work!!!!!!

**#54 - 2014-03-26 16:59 - Alexander Opitz**

Question on this, which *css\_styled\_content* is used? And is the *filemetadata* extension installed.

**#55 - 2014-03-26 21:43 - Christian Reiter**

Alexander Opitz wrote:

| *Question on this, which `css_styled_content` is used? And is the `filemetadata` extension installed.*

I do have 'filemetadata' installed.

The 'Include static (from extensions):' shows plain "CSS Styled content" - value in the DB field is "EXT:css\_styled\_content/static/". None of the "old" versions (4.5 -6.1) are included

**#56 - 2014-03-26 22:08 - Frans Saris**

If you have a remote storage you need to provide an metadata extractor to set the width and height values.  
Currently core does this only for Local storage.

**#57 - 2014-03-28 08:47 - Ole no-lastname-given**

Same here. After Updating core from 6.1.7 to 6.2.  
Changing folder name (filename also) fixes it  
Storage is local btw.

**#58 - 2014-03-28 10:38 - Alexander Opitz**

Workaround for users with the width="0" height="0" issue after upgrading from previous system.

Prerequisite:

- Configuration of ImageMagig or GraphicsMagic is working
- filemetaextract is installed

- 1) Can you take a look into the "sys\_lockedrecords" table. Are there any entries?
- 2) If yes, can you clean this table, clear FE cache and try to view the pages again. Are the images now correctly outputed?
- 3) If no, please clear the sys\_file\_metadata, again clear FE cache and view page again. Are the images now correctly outputed?

Please report back, which action helped or if the problem still exists.

HOWTO 1) and 2)

- a) Go into install tool
- b) Select last point "Clean up"
- c) In "clear tables" there is an entry for the table "sys\_lockedrecords" write down the numbers of rows.
- d) Check the checkbox and press "clear selected tables"
- e) Clear the FE cache
- f) Look into FE

HOWTO 3)

(!) CAUTION YOU SHOULD BACKUP YOUR DATABASE OR YOU MAY NEED TO DO MIGRATION AGAIN FROM CLEAN SOURCE (!)

- a) You need to have access to you SQL database (phpmyadmin/adminer etc)
- b) Use "truncate sys\_file\_metadata" to clear the table. All added data to images will go away (title, description and so on).
- c) Clear the FE cache normaly
- d) Look into FE

**#59 - 2014-03-28 11:21 - Christian Reiter**

- | 1) Can you take a look into the "sys\_lockedrecords" table. Are there any entries?

I checked the sys\_lockedrecords table in an affected installation.

It had 47 entries going back all the way to 2006... but the field "record\_table" for the locked records was never anything to do with FAL (no sys\_file\*

anything).

Are locks to tt\_content records relevant for the problem?

I can't really test the fix described above because I have already run a script to reset all the zero entries to their correct values.

**#60 - 2014-03-28 14:00 - Alexander Opitz**

| *Are locks to tt\_content records relevant for the problem?*

Yes, that might be as the records should be updated to reference to the FAL and not to the file.

| *I can't really test the fix described above because I ...*

Do you have the old database and directory structure, so you could try to migrate locally again?

**#61 - 2014-03-28 15:50 - Philipp Gampe**

We should remove all locks before the migration. The installation should be in maintenance mode anyway.

**#62 - 2014-03-31 00:26 - Newsdesigner**

Thank you for your help and workarounds. The workarounds are ineffective.

In my case, the table "sys\_lockedrecords" was empty. I think, on clearing the FE cache thinks everyone who works with Typo3. The table "sys\_file\_metadata" exists only since version 6.2. I've had the problem because I changed back from 6.2 to 6.1. Also the tip to use "Clean up" wasn't successful.

=> Number cached image sizes: 0

Caused the error is the missing "width" and "weight" values in the table "sys\_file". The simplest solution is to write the data again in the "sys\_file" table. You have to make an update only for rows where type = '2'.

**#63 - 2014-03-31 13:42 - Markus Klein**

- File *imageDimZeroChecker.patch* added

I made a patch which checks the metadata for images if width/height is zero or not specified when creating/updating the metadata. This throws an exception if such a situation arises, so we can finally have a stack trace at least.

**I kindly ask anybody who can reproduce the problem to apply the patch.**

**#64 - 2014-04-01 00:38 - Christian Reiter**

Alexander Opitz wrote:

| *Do you have the old database and directory structure, so you could try to migrate locally again?*



just tested it.

The old DB had 18 tt\_content locks.

Local migration reproduced the same images with width=0 as previously - they happen right on the startpage.

There is absolutely no overlap.

i.e. for the sys\_file\_metadata records with width=0 I checked in which tt\_content elements they were referenced.

None of these content elements were contained in the locked IDs.

Also, not one of pages that contains locked tt\_content elements has the same ID as any of the pages that contain content elements whose references lead to files that have zero-width metadata.

i.e. the locks and the width=0 problems are in completely different areas of the page tree.

**#65 - 2014-04-01 01:26 - Christian Reiter**

*I kindly ask anybody who can reproduce the problem to apply the patch.*

I have a question about the first part of the code. After applying the patch, createMetaDataRecord looks like this:

```
public function createMetaDataRecord($fileUid, array $additionalFields = array()) {
    $emptyRecord = array(
        'file' => (int)$fileUid,
        'pid' => 0,
        'crdate' => $GLOBALS['EXEC_TIME'],
        'tstamp' => $GLOBALS['EXEC_TIME'],
        'cruser_id' => TYPO3_MODE == 'BE' ? $GLOBALS['BE_USER']->user['uid'] : 0
    );
    $emptyRecord = array_merge($emptyRecord, $additionalFields);

    $fileObject = ResourceFactory::getInstance()->getFileObject($fileUid);
    if (strpos($fileObject->getMimeType(), 'image') === 0 && (empty($updateRow['width']) || empty($updateRow['height']))) {
        throw new \ErrorException('Given metadata did not contain valid width/height data for file: ' . $fileObject->getIdentifier());
    }

    $this->getDatabaseConnection()->exec_INSERTquery($this->tableName, $emptyRecord);
    $record = $emptyRecord;
    $record['uid'] = $this->getDatabaseConnection()->sql_insert_id();

    $this->emitRecordCreated($record);

    return $record;
}
```

I don't see where "\$updateRow" comes from in this function.

It's not passed in. So \$updateRow['width'] should always be empty/not set, and it should throw the exception for any image mimetype when the function is called?

Also I didn't find any example where createMetaDataRecord was called with \$additionalFields being filled so it seems this function never inserts anything else than file, pid, crdate, tstamp, cruser\_id ? Or am I missing something?

## #66 - 2014-04-01 07:38 - Markus Klein

- File `imageDimZeroChecker.patch` added

Oh yes, my patch was wrong. C&P mistake. Sorry for that.

## #67 - 2014-04-01 13:51 - Christian Reiter

OK I still have some questions here.

I'm trying to understand the code.

Throwing an exception with the description *Given metadata did not contain valid width/height data for file* from `MetaDataRepository::createMetaDataRecord` means:

- We **expect** that width/height should be given when the code flow arrives at that point
- That is, when new `sys_file_metadata` entries are created on the fly we require width/height to be set (for image mimetype) in the new record data

I don't understand yet where the w/h data are supposed to come from at this point.

- Line 123/124 in the patch <http://forge.typo3.org/attachments/26398/imageDimZeroChecker.patch> seems to **always** throw an exception when `MetaDataRepository::createMetaDataRecord` is called and the file mime type starts with 'image'?
- How do we get there? - When `MetaDataRepository::findByFileUid` is called to get the metadata and none are found (SQL query) - then `createMetaDataRecord` is called with the `sys_file` uid.
  - I find no other call to `createMetaDataRecord` anywhere...?
  - `createMetaDataRecord` can accept *additionalFields* array which could contain width/height but doesn't ever get that parameter...?
  - So it creates only an entry with file id, pid=0, crdate, tstamp, cruser\_id
  - At this point I see no way how the metadata entry can have any width or height info when first created...?

So the new `sys_file_metadata` entry has to wait for a call to `MetaDataRepository::update` to get width and height.

When a `sys_file` entry with no metadata is encountered it will only get the default fields inserted to the corresponding `sys_file_metadata` record as listed above, even though it would be possible to add w/h at this point.

Looking for the update function call I find `MetaDataRepository()->update` in `sysext/core/Classes/Resource/Index/Indexer.php` lines 124, 223 and `sysext/core/Classes/Resource/Service/IndexerService.php` line 157

`IndexerService` does

```
$rawFileLocation = $fileObject->getForLocalProcessing(FALSE);  
$metaData = array();  
list($metaData['width'], $metaData['height']) = getimagesize($rawFileLocation);  
$this->getMetaDataRepository()->update($fileObject->getUid(), $metaData);
```

`Indexer` has the exact same code in 'extractRequiredMetaData' and the metadata extraction in 'runMetaDataExtraction'

Now if I had this kind of code in `MetaDataRepository::createMetaDataRecord` starting from line 120

```
$emptyRecord = array_merge($emptyRecord, $additionalFields);
```

adding something like

```
$fileObject = \TYPO3\CMS\Core\Resource\ResourceFactory::getInstance()->getFileObject($fileUid);  
if (strpos($fileObject->getMimeType(), 'image') === 0 && ( empty($emptyRecord['width']) || empty($emptyRecord['height']) ) ) {  
    $rawFileLocation = $fileObject->getForLocalProcessing(FALSE);  
    list($emptyRecord['width'], $emptyRecord['height']) = getimagesize($rawFileLocation);  
}
```

and then continue as is:

```
$this->getDatabaseConnection()->exec_INSERTquery($this->tableName, $emptyRecord);
```

then I would be doing this:

- Merge \$additionalFields to \$emptyRecord as before (which could contain info including width/height in some implementation)
- If it's an image and has no w/h in merged \$emptyRecord - try to add that info with the same code as in IndexerService
- Save that record as the new default sys\_file\_metadata **without** throwing an exception when getimagesize returns 0/0 for w/h
  - Reason: There are image mimetypes where w/h is correctly 0/0 (or in fact should maybe NULL/NULL) such as SVG which has no raster image dimension readable by getimagewidth.
- I have SVGs in my project and the sys\_file/sys\_file\_metadata tables and don't want them to throw exceptions.

This would change the code to initialize default sys\_file\_metadata entries with w/h instead of waiting for the next indexing to happen.

Placement of the exception in createMetaDataRecord *suggests that this behaviour is expected* but I can't find how it works - admittedly I understand the TYPO3 core code only poorly.

**But I think before it makes sense to go through the stack traces it is necessary to define the expected behavior.**

**Do we expect w/h to exist when createMetaDataRecord inserts a new sys\_file\_metadata entry or not?**

1. If yes - where are we getting w/h info from right now? Is the cause of the entire problem perhaps that on-the-fly metadata generation is supposed to do this but doesn't???
2. If no - we don't need to throw any exception from createMetaDataRecord. We need to wait for indexing and see why that might fail to provide w/h

**#68 - 2014-04-01 14:02 - Markus Maier**

Mathias Bolt Lesniak wrote:

*It's important to note that truncating sys\_file\* tables won't work if you're upgrading a site and have gone through the migration into FAL. That would mess up the references. In these cases it's better to insert all the image widths and heights manually (and clear the cache). I ran into this issue with a site just recently, so I wrote a script that does it for you. (Attached here.) It runs from within typo3conf/.*

*- Mathias*

Thanks a lot!

As far as I understand the code, the script will do the SQL-commands. For some reason, it does only output the SQL-statements in my TYPO3 6.1.7. In that case, it is necessary to add " . ';' before the final semikolon in line 38 so that line 38 reads in total:

```
$sql = 'UPDATE sys_file SET mime_type = \' . $mime . '\', width = \' . $width . '\', height = \' . $height . \' WHERE uid = ' . $row['uid'] . ';;
```

After this, just copy the output and paste it into your phpMyAdmin or whatever you use...

Markus

**#69 - 2014-04-01 14:33 - Mathias Bolt Lesniak**

Markus Maier wrote:

*Mathias Bolt Lesniak wrote:*

*It's important to note that truncating sys\_file\* tables won't work if you're upgrading a site and have gone through the migration into FAL. That would mess up the references. In these cases it's better to insert all the image widths and heights manually (and clear the cache). I ran into this issue with a site just recently, so I wrote a script that does it for you. (Attached here.) It runs from within typo3conf/.*

*- Mathias*

*Thanks a lot!*

*As far as I understand the code, the script will do the SQL-commands. For some reason, it does only output the SQL-statements in my TYPO3 6.1.7. In that case, it is necessary to add " . ";" before the final semikolon in line 38 so that line 38 reads in total:  
[...]*

*After this, just copy the output and paste it into your phpMyAdmin or whatever you use...*

*Markus*

Hi Markus!

Running the file is all you need to do. The SQL output is just for debugging.  
If you get no output, it's likely that the script has found no rows to change.

- Mathias

**#70 - 2014-04-01 16:12 - Markus Klein**

Should be fixed with <https://review.typo3.org/29028>

**#71 - 2014-04-01 16:48 - Alexander Opitz**

- Assigned To set to Alexander Opitz

Issue is dragged, hopefully I can post a patch tomorrow.

**#72 - 2014-04-01 17:07 - Markus Klein**

@Alex: <https://review.typo3.org/29028>

**#73 - 2014-04-01 19:39 - Markus Klein**

## How this all happens

---

I'll describe below how this situation happens, and why the fix from #57495 solves this.

### The basics

---

We have to distinguish between upgrades from < 6.x and upgrades from 6.0/6.1.

#### Database tables

The tables involved are sys\_file and sys\_file\_metadata. These two have a 1:1 relation.

If the first case (<6.x) the database tables sys\_file and sys\_file\_metadata are not present before the upgrade and are created automatically by the first upgrade wizard with the schema required for 6.2.

In the second case (6.0/6.1) the database table sys\_file exists and the sys\_file\_metadata table will be created by the first wizard. The sys\_file table has a different schema than required for 6.2, but all information is present.

#### The wizards involved

The core ships two wizards for the two upgrade paths:

- TceformsUpdateWizard: Takes care of converting old image database fields to FAL files and relations
- FileTableSplittingUpdate: Takes care of moving metadata from sys\_file to the new table sys\_file\_metadata

## The upgrade process (for upgrades from <6.x)

---

After the new tables have been created by the first wizard at some point in time the TceformsUpdateWizard is triggered. It detects that there's data to upgrade and starts with converting each old image field to FAL files. On creation of a FAL file the metadata of this file is extracted (like width/height) and is inserted into the sys\_file\_metadata table.

When TceformsUpdateWizard is finished, FileTableSplittingUpdate has its turn. It checks the integrity of the 1:1 relation. In case there's a mismatch, it will copy the metadata from sys\_file over to sys\_file\_metadata. Keep in mind this wizard is meant for upgrades from 6.x where the sys\_file table has these metadata fields, which are not present if the table was newly created during a 4.x upgrade.

So for a regular 4.x upgrade the FileTableSplittingUpdate has nothing to do as TceformsUpdateWizard already created everything properly. FileTableSplittingUpdate will not even show up in the Install Tool.

## The bug

---

The bug is now that FAL fails to create the sys\_file\_metadata records in the TceformsUpdateWizard step. This leaves behind an empty sys\_file\_metadata table.

The subsequent FileTableSplittingUpdate then finds this situation and sees broken 1:1 integrity. It starts to "copy over" data from the metadata fields of sys\_file, which are **NOT** present. Hence it inserts **empty** sys\_file\_metadata records for all sys\_file records created by TceformsUpdateWizard.

**And voila: we have width and height of 0 for all images (and all other files).**

## Solution

---

By simply fixing the bug #57495 in FAL, TceformsUpdateWizard properly creates all records for both tables and FileTableSplittingUpdate will not even show up.

#74 - 2014-04-02 10:29 - Alexander Opitz

## Why clearing sys\_file\_metadata do not work

---

As Markus already wrote the involved tables sys\_file and sys\_file\_metadata have a 1:1 relation. If a new file is found while a backend user is in a filelist a sys\_file and empty sys\_file\_metadata record is created followed by extracting meta data which updates the sys\_file\_metadata entry.

## What happens if sys\_file\_metadata entry is missing?

---

If the metadata is requested an empty record is created in TYPO3\CMS\Core\Resource\Index\MetaDataRepository::findByFileUid():87 which won't be updated, this update only happens if filelist finds a new file.

## Running the Scheduler task: "File Abstraction Layer: Extract metadata in storage"

---

This won't help. The task tries to read the metadata, which leads to the creation of the empty sys\_file\_metadata record. Now he wants to update it, but there is no registeredExtractionService, so nothing will be updated and the empty sys\_file\_metadata record gets written a second time.

## Why it helps sometimes to delete sys\_file\_metadata for first page request (till clearing FE cache)

---

The value for width/height while processing the files (if not in sys\_file\_metadata) is "NULL". This gets saved in the database as integer field with "0" so second time it is read as "0".

On the side of HTML generation "NULL" will be converted to an empty string while the integer value "0" will be converted to the zero char 0. The browser reads the first as "use size from image" and the second as "use 0 pixel for the image".

## What is with the filemetadata Extension?

---

This extension only adds some fields to sys\_file\_metadata but it do not register any extraction service.

## Conclusion

---

After fixing #57495, normaly you shouldn't have the situation where you have no sys\_file\_metadata record for a file and so no empty metadata records should be created.

But how normal is normaly? We should rethink this situation.

And by the way, if we found a new file, we call Indexer::createIndexEntry() but why not calling ExtractionService if one exists?

**#75 - 2014-04-02 12:20 - Markus Klein**

@Alex: Would you mind moving these important questions to a new ticket? They are more visible there and we can close this one now, since the patch is merged.

**#76 - 2014-04-02 13:19 - Alexander Opitz**

- Status changed from Accepted to Closed

The issue behind this problem is resolved with #57495 so I close this issue.

The open questions are handled in new issues:

#57545 How to get missing metadata extracted into sys\_file\_metadata

#57546 Calling ExtractionService on new files

If you have issues with width/height=0 after you redoing the migration with latest TYPO3 git master or TYPO3 6.2.1 then please open a new issue with a link to this one.

If you have yet issues with width/height=0 after you already upgraded your installation, an update of the source won't fix this problem, but I think you won't have such an installation running live yet (without fixing this issue with a PHP script which updates the Database).

**#77 - 2014-04-16 11:28 - Frans Saris**

For those who run into this problem even after it is fixed now in core I have a easy fix for your local storages.

Run this update command on your DB for every storage:

```
UPDATE sys_file SET missing=1 WHERE storage=1;
```

And then run scheduler task "File Abstraction Layer: Update storage index" for every storage.

De scheduler task sees that the file isn't missing and then re-fetches the basic metadata (with+height)

**#78 - 2014-09-19 15:54 - Hupin Sébastien**

Hi,

I've exactly the same problem and I've changed the default values for width and height for NULL into the sys\_file\_metadata table and all my original image are displayed well

Hope it can help

#79 - 2014-10-15 13:25 - Gernot Ploiner

Same Problem here with TYPO3 6.2.5 after Update from 4.5  
After deleting the record in table sys\_file\_metadata, is was new generated with width/height value '0' again.  
Then i have deleted the file in filelist and uploaded the same file again. Now it works (with new uid in sys\_file).

#80 - 2014-10-15 15:44 - Frans Saris

Did you try?

Frans Saris wrote:

*For those who run into this problem even after it is fixed now in core I have a easy fix for your local storages.*

*Run this update command on your DB for every storage:*

*UPDATE sys\_file SET missing=1 WHERE storage=1;*

*And then run scheduler task "File Abstraction Layer: Update storage index" for every storage.*

*De scheduler task sees that the file isn't missing and then re-fetches the basic metadata (with+height)*

#81 - 2014-12-03 14:36 - Christof Hagedorn

missing width and height in sys\_file\_metadata still in typo3 6.2.3 when migrating raw image data:

setting sys\_file.missing to 1 and run the scheduler task "File Abstraction Layer: Update storage index" did not help.

the problem occured during a raw migration of data from a different cms system.

finally i solved the problem by setting width and height in sys\_file\_metadata with sql generated via bash:

identify -format "update sys\_file\_metadata set width='%w', height='%h' where file in (select uid from sys\_file where identifier ='/images/logos/%f');\n" /my/path/to/images/logos/\* > filesizes.sql

where /images/logos/ was the path to the migrated images.

Files			
sys_file.zip	5 kB	2013-03-27	Steffen Mächtel
AdditionalConfiguration.php	2.2 kB	2013-03-27	Steffen Mächtel
sys_file.zip	6.2 kB	2013-03-29	Freddy Kühne
LocalDriver.php	45.2 kB	2013-06-04	Sven Tappert
imagesize.php	1012 Bytes	2013-12-12	Mathias Bolt Lesniak
imagesize_6_2.php	1.1 kB	2014-03-26	Marc Reiter
imageDimZeroChecker.patch	2 kB	2014-03-31	Markus Klein

