Core - Bug # 65842

Status:	Resolved	**Priority:**	Must have
Author:	Gute Botschafter	**Category:**	Image Generation
Created:	2015-03-19	**Assigned To:**	
Updated:	2015-05-15	**Due date:**	
TYPO3 Version:	6.2		
PHP Version:	5.5		
Complexity:			
Is Regression:	Yes		
Sprint Focus:	Stabilization Sprint		
Subject:	Cropping Images in GIFBUILDER is broken since TYPO3 CMS 6.2.10		

Description

Instead of getting a scaled and then cropped version of the image I seem to get a crop with the given dimensions starting at the top left corner of the original image.
The following snippet worked in TYPO3 CMS 6.2.9 and breaks in 6.2.10 and 6.2.11:

```
10 = IMAGE
10 {
  file = GIFBUILDER
  file {
    XY = 700, 344
    format = jpg
    10 = IMAGE
    10 {
      file.import = uploads/tx_templavoila/
      file.import.current = 1
      file.import.listNum = 0
      file.width = 700
      file.height = 344c
    }
  }
}
```

Downgrading to TYPO3 6.2.9 gives the expected result, the image is being scaled to fit around the given box and then cropped to the required size.

Related issues:

related to Core - Bug # 63519: sys_file_processedfile rows contain zero dimen...	**Resolved**	**2014-12-02**
related to Core - Task # 64643: Remove feature enable_typo3temp_db_tracking	**Resolved**	**2015-01-30**
related to Core - Bug # 65378: Scaling/cropping images in GIFBUILDER does not...	**Resolved**	**2015-02-27**

Associated revisions

Revision c5b7a0d9 - 2015-05-15 14:28 - Morton Jonuschat

[BUGFIX] Temporary filename collision in imageMagickConvert()

Add more entropy to the temporary filename used by imageMagickConvert()
to ensure different processing configurations resulting in the same
output dimensions get different temporary names.

In combination with LocalDriver moving the temporary file to a different location and file_exists_typo3temp_file() having a 30s window where it doesn't detect the moved file (when the user enabled $TYPO3_CONF_VARS['GFX']['enable_typo3temp_db_tracking']) this leads to entries in the sys_processed_file table with a width and height of 0, thus referencing the original file and resulting in unscaled images showing in the frontend.

Resolves: #65842
Resolves: #63519
Resolves: #60215
Related: #65378
Releases: 6.2
Change-Id: I42748d0899bf1e24f9f92f2e55802d64309c5704
Reviewed-on: http://review.typo3.org/39373
Reviewed-by: Stephan Großberndt <stephan@grossberndt.de>
Reviewed-by: Frans Saris <franssaris@gmail.com>
Reviewed-by: Andreas Fernandez <typo3@scripting-base.de>
Tested-by: Andreas Fernandez <typo3@scripting-base.de>
Reviewed-by: Markus Klein <markus.klein@typo3.org>
Tested-by: Markus Klein <markus.klein@typo3.org>

History

#1 - 2015-03-19 15:03 - Torben Hansen

I tried to reproduce the problem but with no success.

On a TYPO3 6.2.9 site, I created a TemplaVoila FCE with a single image and applied the TypoScript code you attached. The resulting image was cropped as expected

Then I switched the core to TYPO3 6.2.10, where the GIFBUILDER problem from #65378 appeared.

Finally I switched the core to TYPO3 6.2.11 and the resulting image is the same as for the TYPO3 6.2.9 site.

Does the problem still appear if you remove temporary files in typo3temp/GB, typo3temp/_processed_ and fileadmin/_processed_ and then let TYPO3 6.2.11 recreate the image?

#2 - 2015-03-19 15:16 - Morton Jonuschat

The problem is persistent. I tried your steps using the Install-Tool before reporting the bug and it doesn't go away when I perform a manual cleanup using a `rm -rf typo3temp/ fileadmin/_processed_` and clear all cache_tables manually using a TRUNCATE SQL statement.
Switching from 6.2.9 directly to 6.2.11 doesn't make any (noticable) difference as well, the images are unscaled/cropped then as well, so I don't think it's leftover files rendered by 6.2.10.

#3 - 2015-04-03 13:33 - Frans Saris

- Status changed from New to Needs Feedback

Is the problem still there even after truncating sys_file_processedfile?

gr. Frans

#4 - 2015-04-25 22:55 - Martin Kokes

I can confirm same problems with heavy cropping/resizing in my fluid templates (<f:image src="..." width="128c" height="96c" ... multiple sizes/pictures on same page) too, 6.2.12. Page (whole gallery) isn't cached at all, this leads to random not-resizing of random images with every (or not every) page reload.

Downgrading to 6.2.9 helps.

#5 - 2015-04-29 07:51 - Morton Jonuschat

I can confirm it's still happening on 6.2.12 as well. Seen it using TypoScript and Extbase as well.

#6 - 2015-05-05 13:57 - Stephan Großberndt

- Subject changed from Cropping Images in GIFBUILD is broken in TYPO3 CMS 6.2.10/6.2.11 to Cropping Images in GIFBUILDER is broken since TYPO3 CMS 6.2.10

#7 - 2015-05-05 14:14 - Morton Jonuschat

A few more details for an incident where the images were not scaling on 6.2.12 (but on 6.2.9):

Extension that's using the following TCA for an image field in a record:

```
'image' => array(
  'exclude' => 0,
  'label' => 'LLL:EXT:foo/Resources/Private/Language/locallang_db.xml:tx_foo_domain_model_bar.image',
  'config' => array(
    'type' => 'group',
    'internal_type' => 'file_reference',
    'uploadfolder' => 'fileadmin/user_upload/foo',
    'allowed' => $GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext'],
    'disallowed' => '',
    'show_thumbs' => 1,
    'readOnly' => 1,
    'size' => 1,
  ) ,
) ,
```

Two calls to the Fluid Image ViewHelper in the Template:

1. {f:image(src: bar.image, alt: bar.title, width: '355m', height: '400m')}
2. {f:image(src: bar.image, alt: bar.title, width: '750m', height: '564m')}

Expected result: Two scaled images with the given dimensions, stored in fileadmin/_processed/
Actual result: Two unscaled images directly referenced in the original folder, completely unscaled.

#8 - 2015-05-05 14:24 - Christian Kuhn

- Sprint Focus set to Stabilization Sprint

#9 - 2015-05-05 14:34 - Markus Klein

- Status changed from Needs Feedback to Accepted

- Target version set to next-patchlevel

#10 - 2015-05-05 14:35 - Christian Kuhn

- Target version changed from next-patchlevel to 6.2.13

#11 - 2015-05-06 20:02 - Morton Jonuschat

Next installation exhibiting this problem:

TYPO3 6.2.12
PHP 5.5.10
GraphicsMagick 1.3.12

Happens in a CE (Type Image). Original file has been migrated to fileadmin/_migrated/pics and is 855x402px large. CE is set to scale the Image to Width 200, Height 0.
Image is shown in the Frontend with original path and size. There is a (presumably broken) entry showing up in sys_file_processedfile referencing the right file in table sys_file.
The serialized config matches the parameters but the fields identifier & name are empty, width & height are recorded as 0

Fix in this case is removing the record from sys_file_processedfile. But it seems to happen a lot, truncating the table it only takes a couple of seconds to have more than 50 more records like this again.

Update:

Switching to ImageMagick makes no difference

#12 - 2015-05-06 20:13 - Frans Saris

For images that didn't need any processing a record is created with: identifier & name = empty, width & height = 0

So that is only wrong if you original images is bigger than the info in the configuration field.

gr. Frans

#13 - 2015-05-06 20:29 - Morton Jonuschat

Ok, I'd need to check that for each entry (or build an SQL query), but in the reported case that's absolutely the case. Original 855x402 (sys_file_metadata agrees), Target: 200xSomething, ProcessedFile: identifier & name = empty, width & height = 0.
Config stored in the entry in sys_file_processedfile:

```
a:10:{s:5:"width";s:3:"200";s:6:"height";N;s:13:"fileExtension";N;s:8:"maxWidth";i:720;s:9:"maxHeight";i:0;s:8:"minWidth";i:0;s:9:"minHeight";i:0;s:7:"noScale";N;s:20:"additionalParameters";N;s:5:"frame";i:0;}
```

#14 - 2015-05-06 20:41 - Frans Saris

No need to check every entry :) I believe that there is an issue. But just wanted to point out that is isn't always wrong when the values are empty.

Would be great if you find a way to reproduce the issue. I encountered is my self too, but everytime I tried to find the source I couldn't reproduce the issue.

gr. Frans

#15 - 2015-05-06 23:13 - Morton Jonuschat

Ok, I think I know what happens, but I'm not yet sure how to reproduce. Going through the call chain:

FileProcessingService passes processing to LocalImageProcessor->processTask()
LocalImageProcessor->processTask() uses LocalCropScaleMaskHelper->process()
LocalCropScaleMaskHelper->process() uses GrapicalFunctions->imageMagickConvert()
GrapicalFunctions->imageMagickConvert() uses GrapicalFunctions::imageMagickExec()

Given the situation that - due to whatever reason (Race condition, Resource Contention...). - no processed file gets written by GrapicalFunctions::imageMagickExec()
The missing output file results in a NULL being returned by GrapicalFunctions->imageMagickConvert() and passed up the call chain by LocalCropScaleMaskHelper->process()
LocalImageProcessor->processTask() sees the NULL and takes it as a hint that the original file should be used (setUsesOriginalFile())
The FileProcessingService takes the (unscaled) ProcessedFile and (mistakenly) add's it to the database where it's used from now on.

#16 - 2015-05-07 20:11 - Morton Jonuschat

For the last 12 hours I logged every call to gm executed by imageMagickExec(), including the exitcode and the output to stdout on a busy installation (after clearing sys_file_processedfile). From ~14.000 calls not a single execution of graphicsmagick exited with an error code (i validated that gm returns an error code that gets detected on an error), yet I had at least two occurrences of this „phenomenon“ within the first 1000 entries in sys_file_processedfile.

The wrong DB entries also happens on 6.2.9 by the way, but the frontend rendering isn't impacted.

#17 - 2015-05-08 21:19 - Morton Jonuschat

I finally managed to track the bug down. To trigger it multiple things are needed.

Conditions to trigger the bug

1. $TYPO3_CONF_VARS['GFX']['enable_typo3temp_db_tracking'] must be enabled
2. The same source image needs to be rendered with different processing configurations that result in the same output dimensions, for example:
   Configuration 1: width = 1400m / height = 1400m / maxWidth = 0 / maxHeight = 0 / minWidth = 0 / minHeight = 0
   Configuration 2: width = 0 / height = 0 / maxWidth = 1400 / maxHeight = 0 / minWidth = 0 / minHeight = 0
3. Scaling with the second configurations needs to happen within 30seconds of first scaling

Analysis

Given the preconditions from above the following happens within imageMagickConvert()

Configuration 1

1. getImageScale() is called and determines the same output dimensions for both configurations
2. The command gets assembled for the scaling

3. Based on the same command and shortMD5() the same filename for the target file gets created
4. file_exists_typo3temp_file() is called and allows the call to imageMagickExec() (no tempfile, no entry in cache_typo3temp_log)
5. The check for the output file succeeds and a valid result array gets returned to LocalImageProcessor->processTask()
6. The output file gets moved to fileadmin/_processed/ and a record is added to sys_file_processedfile

Configuration 2

1. getImageScale() is called and determines the same output dimensions for both configurations
2. The command gets assembled for the scaling
3. Based on the same command as in configuration 1 the same filename for the target file gets created
4. file_exists_typo3temp_file() is called and prohibits the call to imageMagickExec() as there is an entry in cache_typo3temp_log that is less than 30 seconds old
5. The check for the output file **fails** and NULL gets returned to LocalImageProcessor->processTask()
6. This is misinterpreted by LocalImageProcessor as a hint to use the original image
7. due to the different processing configuration and associated checksum a new record is added to sys_file_processedfile (using the original unscaled file)

Regression

This problem has been in existance for much longer than the originally assumed TYPO3 CMS 6.2.10, it started when the files got moved from typo3temp/ to fileadmin/_processed_/
The change in 6.2.10+ to use the processed file information in GIFBUILDER made this visible.
Before the bug resulted in a „use original file“ entry in sys_file_processedfile but that information didn't get used on subsequent calls to Gifbuilder. Since file_exists_typo3temp_file() sets the page to not cached when there is already a rendering in progress this endedup „fixing“ (or working around) the display bug when the page got rendered again 30+ seconds later.

Now the page is still set to not cached for 30 seconds, but the processed file information gets used on the next rendering and we end up with an unscaled image in the frontend. And as long as the processing configuration doesn't change the „wrong“ image gets output even after a cache clear.

Workaround

Disable $TYPO3_CONF_VARS['GFX']['enable_typo3temp_db_tracking'], the image get's rendered twice (once for each configuration) and everything is fine

Solution

Use more/all information from the processingconfiguration to generate the temporary output filename, avoiding the name collision in file_exists_typo3temp_file()

#18 - 2015-05-08 21:36 - Frans Saris

Tnx for finding the cause!

Let's see if we can change the temp name part to get this fixed

#19 - 2015-05-08 22:13 - 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 http://review.typo3.org/39372

#20 - 2015-05-08 22:23 - Gerrit Code Review

Patch set 1 for branch **TYPO3_6-2** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at http://review.typo3.org/39373

#21 - 2015-05-08 22:35 - Gerrit Code Review

Patch set 2 for branch **TYPO3_6-2** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at http://review.typo3.org/39373

#22 - 2015-05-08 23:06 - Gerrit Code Review

Patch set 3 for branch **TYPO3_6-2** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at http://review.typo3.org/39373

#23 - 2015-05-15 14:30 - Morton Jonuschat

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:c5b7a0d92d66ee670d53f976511e44540682195e.