Core - Bug # 61861

Status:	New	**Priority:**	Should have
Author:	Grigori Prokhorov	**Category:**	File Abstraction Layer (FAL)
Created:	2014-09-24	**Assigned To:**	
Updated:	2015-07-06	**Due date:**	
TYPO3 Version:	6.2		
PHP Version:	5.3		
Complexity:	medium		
Is Regression:	No		
Sprint Focus:	On Location Sprint		
Subject:	Frontend throws exception on missing image		

Description

With TYPO3 6.2.3 I currently experience the quite annoying issue that whenever there is an image that is referenced by a viewhelper or any other method that uses the TYPO3 API, the absence of said image leads to Exceptions being thrown in the frontend.

I feel that this behaviour is as annyoing as it is dangerous, since it poses the risk of killing off a complete website with a simple editor's mistake.

Although I understand that there is a need for the handling of a situation like this, I think that the Exception should only be thrown if a file is missing which is not located either in "fileadmin" or "uploads" - both of which are more or less user created content.

The exception in question is 1314516809, along with 1319455097

A possible solution might be as simple as adding && !strpos($path, 'fileadmin') && !strpos($path, 'uploads') to the respective conditions to allow for user generated images to be missing.

Possibly, some kind of error message should be put out, but an Exception seems to much.

Related issues:

related to TYPO3.Fluid - Bug # 9210: imageViewHelper should not throw exception	**Rejected**	**2011-10-15**	**2011-10-15**
related to TYPO3.Fluid - Feature # 9211: Improve ViewHelper exception handling	**Rejected**	**2010-08-09**	
related to Core - Feature # 47919: Catch exceptions in USER TS objects	**Resolved**	**2013-05-04**	
related to Core - Bug # 61125: FAL exception if file does not exist	**Rejected**	**2014-08-21**	
related to Core - Bug # 63955: Fatal Error if Ressource could not be found	**Closed**	**2014-12-17**	

History

#1 - 2014-10-08 19:14 - Frank Gerards

Tbh, simply throwing an exception on such a deep-layer class like LocalDriver and then expecting it will be catched
in dozens of use-cases etc. and WITHOUT exaggerating this in documentation and stuff is simply crap.

Our authors sometimes falsely delete a header/galery image thus breaking multiple pages and the backend without being able
to recover from that error or deleting the corrupt content elements.

So PLEASE:

Throw a FileDoesNotExistsException in the LocalDriver and dont throw an exception at all in the getPermission method,
this is way overdone .

Can one identify all the cases and spots in the core/TS code, where these exceptions could be thrown ?

#2 - 2014-11-20 13:07 - Grigori Prokhorov

Please look into this bug. It does break TYPO3 6.2 installations in the wild - this is undocumented behaviour, AFAIK, also I agree with Frank that this is way too harsh of an error handling in this context.
This is also not simply a question of "The core throws the exception, it's the DEV's responsibility to catch it properly." - This issue occurs without any additional extensions installed.

Reproduction steps:

- Create a new Image Content Element
- Add an image from fileadmin
- Delete the image from the filesystem (yes, there are reasons and cases in which the restriction that referenced images can't be deleted does not suffice)
- Open the page with the Image CE in the frontend

Expected Result:
- The page loads, no image is shown

Actual Result:
- The page throws the above mentioned error

Please give this a higher priority.

#3 - 2014-11-23 20:28 - Moritz Ahl

I can confirm this issue. FAL throws an exception when being called from within a custom viewHelper which is rendering multiple images for responsive layouts (since fluid doesn't support this feature out of the box). Sure, I could catch possible Exceptions in my viewhelper. But in my opinion, FAL should deal with missing (image) files in a more unobtrusive way out of the box so that developers don't have to deal with such basic issues in each and every place which is using images.

This is the last part of my stack trace:

```
Uncaught TYPO3 Exception
#1314516810: File /uploads/tx_news/TeamTour_Bewerbung.jpeg/ does not exist. (More information)

TYPO3\CMS\Core\Resource\Exception\FolderDoesNotExistException thrown in file
C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\core\Classes\Resource\Driver\LocalDriver.php in line 272.

212 TYPO3\CMS\Core\Resource\Driver\LocalDriver::getFolderInfoByIdentifier("uploads/tx_news/TeamTour_Bewerbung.jpeg")

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\core\Classes\Resource\ResourceStorage.php:
02064:   */
02065:  public function getFolder($identifier, $returnInaccessibleFolderObject = FALSE) {
02066:   $data = $this->driver->getFolderInfoByIdentifier($identifier);
02067:   $folder = ResourceFactory::getInstance()->createFolderObject($this, $data['identifier'], $data['name']);
02068:

211 TYPO3\CMS\Core\Resource\ResourceStorage::getFolder("uploads/tx_news/TeamTour_Bewerbung.jpeg")

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\core\Classes\Resource\ResourceFactory.php:
00498:    }
00499:   }
00500:   return $this->getStorageObject($storageUid, array(), $folderIdentifier)->getFolder($folderIdentifier);
```

```
00501: }
00502:

210 TYPO3\CMS\Core\Resource\ResourceFactory::getFolderObjectFromCombinedIdentifier("uploads/tx_news/TeamTour_Bewerbung.jpeg")

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\core\Classes\Resource\ResourceFactory.php:
00468:   } else {
00469:    // only the local path
00470:    return $this->getFolderObjectFromCombinedIdentifier($input);
00471:   }
00472:  }

209 TYPO3\CMS\Core\Resource\ResourceFactory::retrieveFileOrFolderObject("uploads/tx_news/TeamTour_Bewerbung.jpeg")

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\extbase\Classes\Service\ImageService.php:
00128:  } else {
00129:   // We have a combined identifier or legacy (storage 0) path
00130:   $image = $this->resourceFactory->retrieveFileOrFolderObject($src);
00131:  }
00132:  return $image;

208 TYPO3\CMS\Extbase\Service\ImageService::getImageFromSourceString("uploads/tx_news/TeamTour_Bewerbung.jpeg", boolean)

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\extbase\Classes\Service\ImageService.php:
00096: public function getImage($src, $image, $treatIdAsReference) {
00097:  if (is_null($image)) {
00098:   $image = $this->getImageFromSourceString($src, $treatIdAsReference);
00099:  } elseif (is_callable(array($image, 'getOriginalResource'))) {
00100:   // We have a domain model, so we need to fetch the FAL resource object from there

207 TYPO3\CMS\Extbase\Service\ImageService::getImage("uploads/tx_news/TeamTour_Bewerbung.jpeg", NULL, boolean)

C:\xampp\htdocs\typo3_bulls.de\typo3conf\ext\con_bulls\Classes\ViewHelpers\ResponsiveImageViewHelper.php:
00073: ) {
00074:  $imageUris = '';
00075:  $image = $this->imageService->getImage($src, $image, $treatIdAsReference);
00076:
00077:  if ($this->arguments['breakpoints'] && is_array($this->arguments['breakpoints'])) {

206 Tx_ConBulls_ViewHelpers_ResponsiveImageViewHelper::render("uploads/tx_news/TeamTour_Bewerbung.jpeg", "231c", "231c", NULL,
NULL, NULL, NULL, boolean, NULL)
205 call_user_func_array(array, array)

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\fluid\Classes\Core\ViewHelper\AbstractViewHelper.php:
00246:
00247:  try {
00248:   return call_user_func_array(array($this, 'render'), $renderMethodParameters);
00249:  } catch (\TYPO3\CMS\Fluid\Core\ViewHelper\Exception $exception) {
00250:   // @todo [BW] rethrow exception, log, ignore.. depending on the current context

204 TYPO3\CMS\Fluid\Core\ViewHelper\AbstractViewHelper::callRenderMethod()

C:\xampp\htdocs\typo3_src-6.2.6\typo3\sysext\fluid\Classes\Core\ViewHelper\AbstractViewHelper.php:
```

```
00228:   $this->initialize();
00229:
00230:   return $this->callRenderMethod();
00231:  }
00232:
```

#4 - 2014-12-01 10:08 - Tomas Houska

- File Bildschirmfoto 2014-12-01 um 09.47.50.png added

- File Bildschirmfoto 2014-12-01 um 09.48.01.png added

i have found a simple solution for the problem #1319455097.
After i upgrade to Typo3 6.2.6 the Filelist was not accessible. The solution was:
Look to List->File Storage-> click to "uploads (auto created)" -> change the path type to "relative" -> active the option "is online?"

#5 - 2015-01-27 07:59 - Frans Saris

- Sprint Focus set to On Location Sprint

#6 - 2015-01-29 21:41 - Mathias Schreiber

- Status changed from New to Resolved

can't reproduce

#7 - 2015-02-11 21:18 - Michel Mix

This error is very annoying.

We work with different DTAP environments. Upon working on a project I'll check it out from SVN and I download the current production database to have my system up-to-date. Even if I would have had FTP access (which I do not have), I don't want to download 500 or more images from the production server!

No problem to throw an error in the system log when a file is missing, but it should definitely NOT be fatal.

#8 - 2015-03-06 11:25 - Ronald Klomp

Please look into this problem. It is a big problem when you develop sites with more than one developer.

The state is resolved now but thats not correct. I'm working with TYPO3 version 6.2.9 and 6.2.10 and in both version the error is present.

#9 - 2015-03-23 13:54 - Henrik Ziegenhain

Jep, it is still not solved.
Example: I have an extension with simple project data like name and image.

If the selected image got deleted on filesystem, Fluid throws an Exception #1314516810: File /uploads/tx_example/940.jpg/ does not exist.

#10 - 2015-03-30 10:35 - Sascha Egerer

- Status changed from Resolved to Accepted

I reopened this issue as dicussed in slack.

Looks like this one is fixed in TYPO3 7.x but it should really be fixed in 6.2 as it affects the frontend. I think this is a problem in many installations.

The Exception should be catched and logged. -> Check how it is implmented in TYPO3 7

#11 - 2015-03-30 10:59 - Sascha Egerer

Looks like this doesn't effect content elements but for example the f:uri.image viewhelper.

Reproduce: Add {f:uri.image(src: 'invalidFilePath.jpg')} to a fluid template and clear caches.

#12 - 2015-03-30 11:56 - Sascha Egerer

- Status changed from Accepted to Needs Feedback

Ok this is the same behavior as in the current master. I would say that this is related to an integrator error as a non existing file is referenced in TypoScript or in a Template. So I'm fine with throwing a exception in this case.

Any comments on that?

#13 - 2015-03-30 12:07 - Henrik Ziegenhain

The reproduction steps are correct. But I dont think it is an Integrator issue.
Think about this usecase: An editor inserts an image in an extension (maybe news) and afterwards deletes the image via ftp/filelist.

This is something an integrator can't handle.
Usally throwing an exception would be OK, but here it is very annoying.

What about simply "hiding" the image and remove the <img> markup

#14 - 2015-03-30 23:06 - Grigori Prokhorov

Hi Sascha!

This is definitely not an integrator issue.
This issue is also not new in any way[1].
After all, we are building websites for end users and editors, not for developers.
This exception handling is also inconsistent - why does the core not throw an exception for a missing stylesheet, for example?

Anyhow, it should not be the integrator's responsibility to catch an exception ever, since it's not even possible, technically, to properly catch it[2].
One could only work around it by creating a condition which avoids the exception being thrown.

We should not allow for a case where an editor's mistake can kill off a complete page inside a website for no good reason.

A simple text message would be more than enough.

I do understand that from a developer's standpoint the core's work is done as soon as it has notified the affected component of the exception but in this case EXT:fluid should have proper exception handling then.

Sascha Egerer wrote:

> *The Exception should be catched and logged. -> Check how it is implmented in TYPO3 7*

This is by far the most correct and easiest approach to this, although I feel that a local error message (confined to the output of the method) should also be included.

I can provide a patch in a few days if it helps the cause.

Best regards,
Grigori

---

[1] https://forge.typo3.org/issues/9211

[2] http://wiki.typo3.org/Fluid

#15 - 2015-03-31 03:09 - Michel Mix

Sascha, please, common: this is NOT an integrator issue!

First of all: an exception should by definition never occur: "Exceptions should be reserved for conditions that are truly exceptional - in other words, for conditions that cannot be addressed by other coding practices. Exceptions are used in similar circumstances to assertions - for events that are not just infrequent but for events that should *never* occur" (*Code Complete*, my coding Bible ;-)). In this case (a) are we dealing with the error on a daily basis, and (b) it is possible to address the issue.

Now about the use cases:

1. An editor inserts an image in an extension (maybe news) and afterwards deletes the image via ftp/filelist (mentioned by Henrik Ziegenhains)
2. DTAP environment where a developer downloads the production database to the development environment (I mentioned this before).
3. Developers working in seperate development environments (mentioned by Ronald Klomp)
4. Environments with a load balancer; the files on the main server are synced to the other server, but that takes a few minutes.

So, catch the exception and log an error, if you want. But please: DON'T throw uncaught exceptions! The implementation of the connection between database and file system is simply too tight and should be loosened.

#16 - 2015-07-06 11:21 - Alexander Opitz

- Status changed from Needs Feedback to New

Files

Bildschirmfoto 2014-12-01 um 09.47.50.png	70.9 kB	2014-12-01	Tomas Houska
Bildschirmfoto 2014-12-01 um 09.48.01.png	47.7 kB	2014-12-01	Tomas Houska