Core - Bug # 45221

Status:	Resolved	**Priority:**	Must have
Author:	Christian Futterlieb	**Category:**	File Abstraction Layer (FAL)
Created:	2013-02-06	**Assigned To:**	
Updated:	2014-02-27	**Due date:**	
TYPO3 Version:	6.0		
PHP Version:	5.3		
Complexity:			
Is Regression:			
Sprint Focus:			
Subject:	Images with whitespaces in their names are not stored correctly in _processed_		

Description

As of TYPO3 version 6.0.1, images with whitespaces in their names are not correct referenced in the output, because the image src arribute contains the whitespaces (as '%20') but the processed file is stored with underlines.

Example (assuming the image is wider than 50px):

```
page.5 = IMAGE
page.5.file = fileadmin/My Image.jpg
page.5.file.maxW = 50
```

will produce following img tag:

```
<img src="typo3temp/_processed_/My%20Image.jpg" width=50" ... />
```

but the image itself is stored as

```
typo3temp/_processed_/csm_My_Image{$hash}.jpg
```

I did not dig too deep into this problem because I'm not familar with FAL (yet ;)). I just saw, that the TYPO3\CMS\Core\Resource\Driver\LocalDriver calls its method sanitizeFileName() before adding the file in the method addFile(). This is new since 6.0.1.

Thanks for taking care of this one!

Related issues:

related to Core - Bug # 42925: File-Upload does not sanitize umlauts in filen...	**Resolved**	**2012-11-13**
related to Media Management - Bug # 47622: Spaces in filename of image result...	**Closed**	**2013-04-26**
related to Core - Bug # 47140: ProcessedFile/Thumbnail is always regenerated	**Resolved**	**2013-04-11**
duplicated by Core - Bug # 45694: space in filename	**Closed**	**2013-02-21**

Associated revisions

Revision f9ebcda0 - 2013-04-06 16:57 - Helmut Hummel

[BUGFIX] Fix processed files if original has special chars

Filenames of files uploaded in TYPO3 CMS before 6.0.1
can contain problematic characters, because filename
sanitizing of added files was introduced with 6.0.1.

The same happens if files are not uploaded through
the TYPO3 backend and then indexed.

The filenames of processed files are sanitized when
adding them to the storage but the not sanitized
original filename identifier is kept in the database
record of the processed files, which causes wrong
generated paths.

Update the identifier of the processed file along
with all other properties after adding them
to the storage.

Change-Id: I53e4eb42def291ba88ce18209a348b1e2f592185
Resolves: #45221
Related: #42925
Releases: 6.1, 6.0
Reviewed-on: https://review.typo3.org/18529
Reviewed-by: Benjamin Mack
Tested-by: Benjamin Mack

Revision ed988a87 - 2013-04-06 16:58 - Helmut Hummel

[BUGFIX] Fix processed files if original has special chars

Filenames of files uploaded in TYPO3 CMS before 6.0.1
can contain problematic characters, because filename
sanitizing of added files was introduced with 6.0.1.

The same happens if files are not uploaded through
the TYPO3 backend and then indexed.

The filenames of processed files are sanitized when
adding them to the storage but the not sanitized
original filename identifier is kept in the database
record of the processed files, which causes wrong
generated paths.

Update the identifier of the processed file along
with all other properties after adding them
to the storage.

Change-Id: I53e4eb42def291ba88ce18209a348b1e2f592185
Resolves: #45221
Related: #42925
Releases: 6.1, 6.0

Reviewed-on: https://review.typo3.org/18529
Reviewed-by: Benjamin Mack
Tested-by: Benjamin Mack
Reviewed-on: https://review.typo3.org/19682

History

#1 - 2013-02-07 22:05 - Andreas Wolf

- Status changed from New to Accepted

I guess this is because the changed filename is not reported back by the driver. This would be a general issue for all file-adding methods.

We clearly miss functional tests for the whole storage–driver block in FAL...

#2 - 2013-02-14 10:40 - Joschi Kuphal

I can confirm this for file names containing an "@" symbol (and probably many more characters) as well. This bug is rendering the whole 6.0.1 update completely useless for us in two cases / for two websites, as we have tons of image files with non-alphanumeric characters in their names (characters that are perfectly legal for file names in general though).

as the driver simply changes the file name (where there is no need to change it btw. in case of the "@" symbol), but doesn't report back the change (and doesn't throw an error either), there's no chance to notice the problem at all. so the calling process will continue running, in the belief of the image being generated properly, but of course displaying the image will fail, as the real name on the harddrive is different.

#3 - 2013-02-14 11:08 - Christian Futterlieb

> *This bug is rendering the whole 6.0.1 update completely useless for us*

Yes, that is the same for me, I can't use 6.0.1 because of this bug either

#4 - 2013-02-22 14:10 - Stefan Völker

same here under 6.0.2
We can't update ANY project as long as this bug is present!

#5 - 2013-02-27 23:51 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/18529

#6 - 2013-02-28 00:00 - Tilo Baller

- File s0me_Filé_with_nôn-_scii_Char-acters.jpg added

As far as I discovered from my own experience with this problem, there are three cases in general we have to differ between:
1.) You upload a file containing non-ascii (accurately: characters not matching [.a-zA-Z0-9_-]) in its name in TYPO3 Version >= 6.0.1.
No problems at all. File gets sanitized on upload.

2.) You upload or create a file containing non-ascii (accurately: characters not matching [.a-zA-Z0-9_-]) in its name in TYPO3 Version < 6.0.1.
In this case the filename does not get sanitized. See these issues: http://forge.typo3.org/issues/42925, http://forge.typo3.org/issues/42873
With v6.0.1 the fix was included which sanitizes files uploaded in the backend.
Additional Info: Sanitizing was already done before 6.0.1 when renaming a file (or creating a folder).

3.) You place a file inside your file storage somehow without uploading it in the backend (e.g. via FTP upload).
As far as I know the filename will not get sanitized at all, because the indexing/(processing?) task leaves the original file as it is.
Maybe the processing task should take care of sanitizing, but I'm not really into FAL yet, so maybe it could cause problems if the file gets referenced (e.g. in a tt_content element) close after its creation without being processed by the indexing/(processing?) task at this time.

Sorry for maybe mixing up some words like 'indexing' and 'processing'.

.

The attached patch also sanitizes the filename of the rendered processed files when the original filename contains disallowed characters, which are cause by case 2) or 3).

.

Important Note:
I used a existing but slightly different sanitizing function. I'm not sure if this could cause problems in some unknown cases. Maybe it would be better to extract the sanitizeFileName() function currently placed in LocalDriver-Class to some Utility-Class an use this one instead.

I'm also not sure if it might cause problems with other drivers than the local one (e.g. Dropbox/Amazon Drivers).

.

I did the following **tests**:

Unit test:
- added filename containing some non-ascii characters to filenameExtensionDataProvider used by getNameWithoutExtensionReturnsCorrectName() and getExtensionReturnsCorrectExtension() tests

Manual acceptance test:
- uploaded attached test file
  - checked for preview image in File-Module
  - checked for preview image in Info-Popup of File-module

- referenced image in tt_content element
  - checked for preview image in edit form of tt_content element
  - checked in frontend for rendered image

#7 - 2013-02-28 00:31 - Joschi Kuphal

Just a quick note in response to Tilo's details / patch: I personally do think that a very problematic part is the general understanding of "sanitizing" here. There might be situations (as it is the case for us) where it is **AN ABSOLUTE NO-GO** to simply change / "sanitize" file names just because someone decided that everything but [.a-zA-Z0-9_-] is illegal for file names. the simple but true reason is: it is not. although i'd never recommend using exotic characters in file names there's a pretty good chance that they are de facto legal for every modern (utf8 capable) operating / server system. in

our case the images containing an (at)-mark in their file names come out of another software system (!= TYPO3), and i simply expect TYPO3 to work with these files without imposing the absurd need of "sanitizing" file names just for the sake of it, as this is going to break the other system's functionality. again, one should take into account that an (at)-mark (being part of the ASCII range) definitely is perfectly ok for file names and even URIs. Trying to find a solution for "sanitizing" the names of files that didn't get uploaded through the backend is the wrong way imho. Instead FAL should better learn how to deal with a wider range of legal file names instead of crippling the specification.

As far as i can say right now: the proposed method still leaves us in a state where we cannot update our TYPO3 installations as it would break functionality.

#8 - 2013-02-28 21:40 - Alexander Opitz

Till yet I didn't know that the FAL will sanitize filenames on the storage. As Joschi Kuphal wrote, this is a NO-GO.

#9 - 2013-03-01 11:17 - Christian Weiske

I can reproduce issues with preview images that have special chars in file names, so sanitizing them is correct.

If you want to change the sanitation rules (e.g. to allow "@"), please open another issue.

#10 - 2013-03-02 12:35 - Joschi Kuphal

@Christian Weiske: Sorry, but you're missing the point somehow (though this might be due to me not being too specific). Let me try to clarify what I was talking about:

In the following paragraphs I will substitute the (at) symbol with § as it will mess up formating otherwise.

Say you have a bunch of image files containing §-symbols (or any other "FAL-thinks-this-character-has-to-be-sanitized-character"), e.g. "§3_1aea065a9809069107e196c918c9e75c.png". The image files could reside anywhere, but let's assume they are all inside "typo3temp/subfolder". The images got there "somehow", i.e. not via TYPO3 backend upload or anything similar, but via FTP for example. Next, you're going to author an (extbase/fluid) extension, and inside a regular controller action you're doing the following:

```
...

// Render the temporary image resource
$tempImageConfig          = array(
   'width' => max(1, intval($width)).'m',
   'height' => max(1, intval($height)).'m',
);
$tempImageInfo            = $this->configurationManager->getContentObject()->getImgResource($localImagePath, $tempImageConfig);
$tempImagePath             = trim($tempImageInfo[3]);

...
```

What would you expect tempImagePath to be in the end, assuming localImagePath was "typo3temp/subfolder/§3_1aea065a9809069107e196c918c9e75c.png"? I would expect it to be "typo3temp/_processed_/csm_§3_1aea065a9809069107e196c918c9e75c_2fee8346ac.png" (the "_2fee8346ac" suffix of course could be anything, but this is a real live example), and guess what: it is! **But the problem is: There's no such file** (at least starting with TYPO3 v6.0.1)! Instead a file "typo3temp/_processed_/csm__3_1aea065a9809069107e196c918c9e75c_2fee8346ac.png" has been created, with the §-symbol replaced by an underscore. IMHO there are two problems arising with this:

1. The controller action doesn't get notified about the renaming. Instead it gets back the expected, but non-existent file path. Any further steps

involving the file path will fail (in our example, creating a symlink and also displaying the image in the frontend). Bummer. **This definitely is a (serious) bug has to be fixed as soon as possible** (if not already the case with 6.0.2).

2. Secondly, I don't see any need why the §-symbol has to be substituted at all. Can you tell me? Is there any true reason for these strict sanitation rules, or are they just arbitrary? As i said before, almost any OS should be capable of even handling UTF-8 file names, so what's the problem about the absolutely innocent §-symbol (and those many other characters suffering the same FAL restriction)? In general I would agree that file name sanitation is a good idea. But in this special situation I think sanitation does more harm than good. At least there should be the possibility to disable it intentionally. IMHO the logic should be: If the file system contains files, that didn't get there via TYPO3, and if the file system can obviously deal with them, and if even TYPO3 seems to be able to deal with them (e.g. it can read and convert them), then there should be no need for "over-sanitizing" any of the file names involved in the whole process. Even if the sanitized file name would be returned properly (see 1.) there could be succeeding program logic relying on a particular file name structure. It should not be up to FAL to freely decide about how file names should or should not look like, without any control for the user.

#11 - 2013-03-04 09:51 - Christian Futterlieb

> ***This definitely is a (serious) bug has to be fixed as soon as possible***

Totally agree ;)

> *(if not already the case with 6.0.2).*

No, it's not

And: I'd suggest to keep the focus on the particular issue to get it solved asap.

@Joschi Kuphal: Don't misunderstand me, the discussion on filename sanitation is important in my eyes also and I agree your argumentation. But I'd like to see the current problem solved first and not blocked by a discussion, that is even more 'general-scope' and should be discussed separately therefore.

#12 - 2013-03-04 10:48 - Joschi Kuphal

@Christian Futterlieb:

> *@Joschi Kuphal: Don't misunderstand me, the discussion on filename sanitation is important in my eyes also and I agree your argumentation. But I'd like to see the current problem solved first and not blocked by a discussion, that is even more 'general-scope' and should be discussed separately therefore.*

I'm absolutely with you in this point, don't worry ;) First of all I'm interested in updating our websites to the most recent TYPO3 version, and solving this issue will enable us to do so again (hopefully). Please consider my general thoughts about file name sanitation just as a collateral mumbling ... :)

#13 - 2013-03-22 00:21 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/18529

#14 - 2013-03-22 16:43 - Andreas Wolf

I think it's obvious that something has to be done here. As not all filesystems out there are UTF-8/Unicode capable, I suggest to do the following: By default allow UTF-8 characters, but disallow them when $TYPO3_CONF_VARS[SYS][UTF8filesystem] is set to false.

To sanitize the filenames, we should not create a RegEx or other filter ourselves, but instead use filter_var() provided by PHP:

```
filter_var("<filename>", FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW); // for UTF-8
filter_var("<filename>", FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW | FILTER_FLAG_STRIP_HIGH); // ASCII only
```

Additionally, we have to strip certain other characters (e.g. *, /, \, :, ?) ourselves.

#15 - 2013-03-22 17:22 - Andreas Wolf

Alexander Opitz wrote:

> *Till yet I didn't know that the FAL will sanitize filenames on the storage. As Joschi Kuphal wrote, this is a NO-GO.*

File names were sanitized since TYPO3 i-don't-know. I found traces of it in t3lib_basicFileFunc from late 2006, which would be TYPO3 4.0/4.1. So this is nothing really new, please don't be so upset...

@Joschi: I guess you have not enabled TYPO3_CONF_VARS[SYS][UTF8filesystem], which would allow the (at) character in filenames.

The RegEx currently used in the local driver is this:

```
preg_replace('/[\\x00-\\x2C\\/\\x3A-\\x3F\\x5B-\\x60\\x7B-\\xBF]/u', '_', trim($fileName));
```

I think we could strip it down to removing the following character ranges:
- 0x00-0x1F (non-printable ASCII characters
- \, /, ?, *, :
- 0x7F-0xBF (note: 0x7F is the DEL character, 0x80-0xBF mark the second, third or fourth byte of a UTF-8 byte sequence and thus are invalid characters when encountered alone; refer to https://de.wikipedia.org/wiki/UTF-8 for details)

#16 - 2013-03-22 17:47 - Joschi Kuphal

@Alexander:

> *@Joschi: I guess you have not enabled TYPO3_CONF_VARS[SYS][UTF8filesystem], which would allow the (at) character in filenames.*

yes, you're right, UTF8filesystem wasn't / isn't enabled at our installations in question. i'd be glad if this was the solution for (at least) our problem with the (at) symbol, thanks a lot for the hint! i should have thought of it before, but i must admit that it didn't come to my mind as the (at) symbol is not a UTF8 character altogether (didn't have the time to dig into the core code as well, otherwise it might have come to my attention before ...). unfortunately i'm not able to test this instantly, but i definitely will by next week.

however, the driver not returning the correct / sanitized filename is problem of it's own - which hopefully got patched now. thanks for this one as well!

#17 - 2013-03-23 15:04 - Andreas Wolf

Joschi Kuphal wrote:

> *@Joschi: I guess you have not enabled TYPO3_CONF_VARS[SYS][UTF8filesystem], which would allow the (at) character in filenames.*
>
> *yes, you're right, UTF8filesystem wasn't / isn't enabled at our installations in question. i'd be glad if this was the solution for (at least) our problem with the (at) symbol, thanks a lot for the hint! i should have thought of it before, but i must admit that it didn't come to my mind as the (at) symbol is not a UTF8 character altogether (didn't have the time to dig into the core code as well, otherwise it might have come to my attention before ...). unfortunately i'm not able to test this instantly, but i definitely will by next week.*

Ok, would be great to get some feedback. :-)

> *however, the driver not returning the correct / sanitized filename is problem of it's own - which hopefully got patched now. thanks for this one as well!*

Yeah, I don't know why we didn't fix this earlier...

#18 - 2013-04-06 00:06 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/18529

#19 - 2013-04-06 16:58 - Gerrit Code Review

Patch set 1 for branch **TYPO3_6-0** has been pushed to the review server.
It is available at https://review.typo3.org/19682

#20 - 2013-04-06 16:58 - Benjamin Mack

- Status changed from Under Review to Resolved

#21 - 2013-04-17 11:41 - Joschi Kuphal

@Andreas: sorry for the long delay, but just to give you the feedback i promised (we did check it just yesterday ...): **YES**, activating UTF8filesystem prevents the driver from rewriting the (at)-mark, so our specific problem is solved this way. thanks again for the hint (although i still dont associate the (at)-mark with UTF8 ;))!

Files

s0me_Filé_wìth_nôn-_scii_Char-acters.jpg	752.3 kB	2013-02-28	Tilo Baller