

## Core - Bug # 48998

<b>Status:</b>	Resolved	<b>Priority:</b>	Must have
<b>Author:</b>	Klaus Hinum	<b>Category:</b>	File Abstraction Layer (FAL)
<b>Created:</b>	2013-06-10	<b>Assigned To:</b>	
<b>Updated:</b>	2013-11-16	<b>Due date:</b>	
<b>TYPO3 Version:</b>	6.1		
<b>PHP Version:</b>			
<b>Complexity:</b>			
<b>Is Regression:</b>	No		
<b>Sprint Focus:</b>			
<b>Subject:</b>	FAL files get reindexed in storage null when used with tt_content		
<b>Description</b>			
<p>As described in the fixed bug 40669, the FAL files are reindexed in storage 0 again when a page is shown. <a href="http://forge.typo3.org/issues/40669#change-168691">http://forge.typo3.org/issues/40669#change-168691</a> E.g. one image indexed in storage 1 (fileadmin) as /test/test.jpg and as soon as it is added to a tt_content and previewed in the frontend /fileadmin/test/test.jpg is also added to the sys_file storage (with storage 0) and not referenced in sys_file_reference.</p> <p>The bug breaks replication all the time (if the page was not previewed on the backend server first) and of course bloats the database with nearly twice the amount of sys_file entries than needed.</p> <p>Tested with Typo3 6.1.1, and may be a follow up since the bug was fixed that prevented the adding of <i>processed</i> files to the sys_file table.</p>			
<b>Related issues:</b>			
related to Core - Bug # 40669: FAL files get reindexed in storage null when u...		<b>Resolved</b>	<b>2012-09-06</b>
related to Core - Bug # 47211: ProcessedFile is persisted in sys_file as well		<b>Resolved</b>	<b>2013-04-14</b>
related to Core - Bug # 49212: File links from tt_content bodytext are not/wr...		<b>Closed</b>	<b>2013-06-18</b>

### History

#### #1 - 2013-06-17 11:17 - Klaus Hinum

- % Done changed from 100 to 0

#### #2 - 2013-07-02 18:10 - Ernesto Baschny

- Status changed from New to Accepted

- Target version set to next-patchlevel

According to Steffen this was indeed fixed in 6.0 and appears again in 6.1 (and thus 6.2). Remains unclear still what (re-)introduced this bug.

#### #3 - 2013-07-02 18:38 - Ernesto Baschny

- Status changed from Accepted to Needs Feedback

I was unable to reproduce that on 6.2 "introduction package" standard installation. Could you please

a) test with 6.2 / master

b) check if there aren't any extensions influencing this?

#### #4 - 2013-07-04 11:02 - Nico de Haen

I just tested it with 6.1.1 and 6.2 and can confirm that it is somehow fixed in 6.2.

Although I have no idea where, since the involved files RteHtmlParser and ResourceFactory did not change in relevant parts.

Maybe the change happened somewhere else in FAL? (Perhaps a case for git-bisect)

#### #5 - 2013-07-04 11:48 - Nico de Haen

I was wrong, the bug is not yet fixed!

The problem is:

The parser calls ResourceFactory->retrieveFileOrFolderObject(\$input) where input is the filepath.

The comment says: input can be "uploads/myfile.png" (backwards-compatibility, storage "0")

Thus the file is added with storage 0 and "lost". No idea how storage with uid 0 should be handled...

#### #6 - 2013-07-04 16:09 - Alexander Opitz

- Status changed from Needs Feedback to New

#### #7 - 2013-07-07 00:07 - Steffen Ritter

- Status changed from New to Needs Feedback

Puh, I tried to reproduce as I think I had this some weeks ago, too.

Currently I cannot reproduce.

Please give more information about, when this happens...

#### #8 - 2013-07-07 08:02 - Nico de Haen

It happens when opening a content element containing a file link with a not converted reference like

<link fileadmin/download/myfile.txt >Download</link> in the backend.

Then the RteHtmlParser->TS\_links\_rte converts the link into a FAL reference by calling ResourceFactory->retrieveFileOrFolderObject as described above and a sys\_file entry with storage 0 is added.

With the RteFileLinksUpdateWizard Francois pushed yesterday to Gerrit <https://review.typo3.org/#/c/22002/> such links should not occur anymore after migration, but it should be fixed anyway. Perhaps with the same routine as in the Wizards.

#### #9 - 2013-07-07 09:59 - Klaus Hinum

I can reproduce it with a 6.1.1 installation.

Just add a new content element text w. image to a page, add a new image (best upload it in the window). As soon as you do a preview of the page, the image is added again to the sys\_file table without any storage (already in it with storage 1). As you can see of the DB dump below, two identical entries are in the sys\_file table, one with storage 1 (fileadmin) and the other with storage 0 and fileadmin prepended.

```
INSERT INTO `sys_file` (`uid`, `pid`, `tstamp`, `crdate`, `cruser_id`, `deleted`, `t3ver_oid`, `t3ver_id`, `t3ver_wsid`, `t3ver_label`, `t3ver_state`,
```

Btw it also happens for old extensions that still use the uploads directory, the sys file is added when previewed.

still happening with 6.1.5 all the time (and therefore breaking our replication every day)

- Status changed from Needs Feedback to New
- Is Regression set to No

looked into it closer now, I cant reproduce it with 6.1.5 (therefore the above mentioned case wont happen anymore), but still the replication broke a couple of times due to sys\_file items created on the frontend server. Will look into it and update this as soon as it happens again and I find out more.

- Status changed from New to Needs Feedback

- Status changed from Needs Feedback to Resolved

the double indexation should be fixed with the "findBestMatchingSTorage"