TYPO3 Core - Bug #75881

Improper usage of processed files

2016-04-23 12:45 - Philipp Gampe

Status:	Closed		**Start date:**	2016-04-23
Priority:	Should have		**Due date:**	
Assignee:			**% Done:**	0%
Category:	File Abstraction Layer (FAL)		**Estimated time:**	0.00 hour
Target version:				
TYPO3 Version:	7		**Complexity:**	medium
PHP Version:			**Is Regression:**	No
Tags:			**Sprint Focus:**	

Description

$file->process(...) returns a ProcessedFile object, even it could not process the image.
The ProcessedFile has a method isProcessed() that needs to be used to check the result, otherwise ProcessedFile returns the properties of the original file.

History

#1 - 2016-04-23 12:47 - Gerrit Code Review

- Status changed from New 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/47871

#2 - 2016-08-30 23:25 - Philipp Gampe

- Status changed from Under Review to New

- Assignee deleted (Philipp Gampe)

- Complexity set to medium

#3 - 2019-03-06 21:20 - Susanne Moog

- Status changed from New to Needs Feedback

After revisiting this topic a few years later, I've come to the conclusion that I either don't get what the issue is anymore or there is no issue. Basically, the naming of "processed" is a bit ambivalent as in this case it simply means "hey, I've seen that file at some point" and not "I've done something with it". From the (current) code, I see no issue. Do you?

#4 - 2019-03-09 14:48 - Philipp Gampe

I did not verify lately, but at least the back in the days, the API returned an unprocessed file object in case of errors. If this is the expected behavior, then this should be documented.

Ambivalence in the API is an own bug category :/

You can as well go ahead and close this one. If nobody else bother since several years now, it must be good enough and time may be better spend elsewhere.

#5 - 2020-03-20 15:42 - Susanne Moog

- Status changed from Needs Feedback to Closed

Thanks for the feedback, closing here (just as a side note: language itself is pretty ambivalent ;))