

## Core - Bug # 60650

<b>Status:</b>	Resolved	<b>Priority:</b>	Should have
<b>Author:</b>	Jost Baron	<b>Category:</b>	
<b>Created:</b>	2014-07-29	<b>Assigned To:</b>	
<b>Updated:</b>	2014-12-14	<b>Due date:</b>	
<b>TYPO3 Version:</b>	6.2		
<b>PHP Version:</b>			
<b>Complexity:</b>			
<b>Is Regression:</b>	No		
<b>Sprint Focus:</b>			
<b>Subject:</b>	ConfigurationManager::getConfigurationValueByPath ignores AdditionalConfiguration.php		
<b>Description</b>	Judging by the uses in the core, it should query the final \$GLOBALS['TYPO3_CONF_VARS']-array.		
<b>Related issues:</b>			
related to Core - Bug # 52557: Show message and disallow modification of non ...		<b>Under Review</b>	<b>2013-10-06</b>
related to Core - Bug # 62301: ConfigurationManager setters make system unstable		<b>Rejected</b>	<b>2014-10-17</b>
related to Core - Bug # 60294: If a local file storage is outside the documen...		<b>Closed</b>	<b>2014-07-13</b>
duplicates Core - Bug # 53417: ConfigurationManager->getConfigurationValueByP...		<b>Rejected</b>	<b>2013-11-07</b>

### Associated revisions

**Revision 2d574641 - 2014-12-14 14:51 - Markus Klein**

[TASK] Add note about AdditionalConfiguration.php to Install Tool

Add a note to clarify that AdditionalConfiguration.php is always overruling any setting made within the Install Tool.

Furthermore we add a note to the ConfigurationManager that it is limited to Core internal usage only.

Resolves: #60650

Releases: master, 6.2

Change-Id: Icaf7d4b1ce5b14ad0618e38e92a49a33405997f7

Reviewed-on: <http://review.typo3.org/35409>

Reviewed-by: Christian Kuhn <[lolli@schwarzbu.ch](mailto:lolli@schwarzbu.ch)>

Tested-by: Christian Kuhn <[lolli@schwarzbu.ch](mailto:lolli@schwarzbu.ch)>

Reviewed-by: Anja Leichsenring <[leichsenring@ab-softlab.de](mailto:leichsenring@ab-softlab.de)>

Tested-by: Anja Leichsenring <[leichsenring@ab-softlab.de](mailto:leichsenring@ab-softlab.de)>

Reviewed-by: Wouter Wolters <[typo3@wouterwolters.nl](mailto:typo3@wouterwolters.nl)>

Tested-by: Wouter Wolters <[typo3@wouterwolters.nl](mailto:typo3@wouterwolters.nl)>

Reviewed-by: Alexander Opitz <[opitz.alexander@gmail.com](mailto:opitz.alexander@gmail.com)>

Tested-by: Alexander Opitz <[opitz.alexander@gmail.com](mailto:opitz.alexander@gmail.com)>

**Revision 26077111 - 2014-12-14 15:01 - Markus Klein**

[TASK] Add note about AdditionalConfiguration.php to Install Tool

Add a note to clarify that AdditionalConfiguration.php is always

overruling any setting made within the Install Tool.

Furthermore we add a note to the ConfigurationManager that it is limited to Core internal usage only.

Resolves: #60650

Releases: master, 6.2

Change-Id: Icaf7d4b1ce5b14ad0618e38e92a49a33405997f7

Reviewed-on: <http://review.typo3.org/35412>

Reviewed-by: Anja Leichsenring <[leichsenring@ab-softlab.de](mailto:leichsenring@ab-softlab.de)>

Tested-by: Anja Leichsenring <[leichsenring@ab-softlab.de](mailto:leichsenring@ab-softlab.de)>

## History

---

### #1 - 2014-07-29 01:32 - Markus Klein

So you're using the ConfigurationManager in your code somewhere to retrieve configuration, right?

### #2 - 2014-07-29 10:52 - Jost Baron

Yes, only for read access - is it not meant to be done that way? Since the core seems to do it, too

### #3 - 2014-07-29 13:07 - Markus Klein

- Status changed from New to Needs Feedback

Where does the Core use it too?

AFAIK ConfigurationManager from Core is to be used only internally (currently only in bootstrap and install tool) in order to manage the various configuration files.

All other Core parts as well as extensions should really use the global TYPO3\_CONF\_VARS array which hold the real configuration (including everything from AdditionalConfiguration.php)

Be aware that there is another ConfigurationManager from extbase, which is intended to be used by extensions too.

### #4 - 2014-07-29 14:50 - Jost Baron

There is one place on the extension manager using this method (fetching the current extension configuration under EXT/extConf/extkey), and several instances in the install tool.

The usage in the extension manager prevents changed extension settings from AdditionalConfiguration.php to be shown in the configuration area of the extension manager. This could be a problem, since the value that is actually used and the value that is shown value are different. To reproduce, add this line to your AdditionalConfiguration.php

```
$GLOBALS['TYPO3_CONF_VARS']['EXT']['extConf']['rsaauth'] = 'a:1:{s:18:"temporaryDirectory";s:0:"";}';
```

and change the setting via extension manager. Then the changed setting is shown, but the above setting is used.

I'm not sure about the usages in the install tool, some seem legit (at least under certain assumptions, for example that there is no AdditionalConfiguration.php to be respected while the initial setup is running), about some others I'm not so sure. For example, a configuration preset is marked 'inactive', if its settings are there, but only in AdditionalConfiguration.php.

## #5 - 2014-08-04 22:20 - Christian Kuhn

Well, yeah.

The core reads some information directly from LocalConfiguration to be sure that the overload file AdditionalConfiguration is ignored at some places.

The problem with AdditionalConfiguration is, that we can not reliably detect or find out what exactly is changed there:

The related issue #52557 tries to detect in "all configuration" in install tool if a value is changed by Additional configuration by comparing the setting within LocalConfiguration and the one in \$GLOBALS['TYPO3\_CONF\_VARS']. If it finds a difference, then it knows "ok, this value was probably changed in AdditionalConfiguration". But this is not reliable since there may be code that only hits if the system is production or dev, or if it is 1 o'clock in the morning or other stuff. Even "grepping" in AdditionalConfiguration for code may not be reliable since we do not know if someone included other files.

So neither executing nor scanning this file gives us a definite knowledge if someone change something in this file.

Therefore, AdditionalConfiguration is currently "ignored" by the core. Core basically says: If you have funny code in this file, that change settings, then those settings may not be settable by core handling code anymore, since we never know what and how you are doing things in there. Thus, it is in developers hands. If a developer overwrites settings there (eg. based on context environment variable), then these settings are basically out-of-support by core code like install tool or em. If a dev forgets about overwritten stuff in this file, and then wonders why his settings changed by install tool or em does not work, well, i just see no good option to help there. In my opinion we can not even help to detect whether an option was changed or not, any code we would come up with is probably doomed to fail in some situations.

Thus, I tend to say: If you use AdditionalConfiguration, then you **must** take care of those options on your own, together with the according side effects it may have.

I'd close this issue as rejected for these reasons, and I tend to also vote down Sascha's pending patch for #52557.

Maybe, if you align with this view, we could think about how to hint devs better about this fact. Maybe we could add hints (to install tool or/and em) that settings changed within AdditionalConfiguration are ignored by the install tool / em code and changing them within these modules may not have the desired effect. This way, we would at least achieve a better information strategy about this situation.

## #6 - 2014-08-04 22:49 - Jost Baron

Hi Christian,

I still have some concerns:

1. Because of some blog posts and other stuff easily found with Google, many people use the ConfigurationManager::getConfigurationValueByPath() method (and also the setters). That's how I came to use it, too. There is no clear indication that it should not be used outside the core, and there is obviously the need for a service like it, which behaves as expected, and which can also be mocked easily.

Is it a good idea to add another "configuration manager" to the core, but for use in extensions to query and modify the \$GLOBALS['TYPO3\_CONF\_VARS'] array? Maybe even replacing the current ConfigurationManager with that one, and moving the current ConfigurationManager to some place else and giving it a more speaking name, like LocalConfigurationFileManager?

2. It's clear that the install tool can't ever analyze which options are changed by the AdditionalConfiguration.php. That problem is probably equivalent to the halting problem. But I think what Sascha does makes (partly) sense. There should be a warning that a settings value is not taken from the current LocalConfiguration.php, but comes from somewhere else (in the current context). Just a warning, to make debugging easier. Same thing for the extension manager.

## #7 - 2014-08-12 20:03 - Christian Kuhn

1. I think writing the LocalConfiguration file should be a task done by em / install tool (=core) only. Extensions should rely on \$GLOBALS['TYPO3\_CONF\_VARS'] only, not on using some API (BTW: this array can be easily set/mocked within unit test, core tests do that at a lot of places). Maybe we should mark the class and its methods as "@internal, do not use in extensions, use \$GLOBALS['TYPO3\_CONF\_VARS'] directly" to hint devs about this core-internal class, and is not public API.

2. Yep, better hinting in the em / install tool that stuff in AdditionalConfiguration may break "All configuration" (and maybe more?) is a good idea.

#### #8 - 2014-12-12 23:09 - Christian Kuhn

- Status changed from Needs Feedback to Accepted

Tasks to do:

- State more clearly in ConfigurationManager that this is a lowlevel core API that is **usually** only touched by core, and that extensions usually shouldn't operate with it.
- Add something to the install tool that changing AllConfiguration (and more? maybe presets) may not have the desired effect if AdditionalConfiguration overwrites this stuff again.

#### #9 - 2014-12-14 13:22 - 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/35409>

#### #10 - 2014-12-14 13:59 - Gerrit Code Review

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

It is available at <http://review.typo3.org/35409>

#### #11 - 2014-12-14 14:29 - Gerrit Code Review

Patch set 3 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <http://review.typo3.org/35409>

#### #12 - 2014-12-14 14:30 - Gerrit Code Review

Patch set 4 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <http://review.typo3.org/35409>

#### #13 - 2014-12-14 15:00 - Markus Klein

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:2d5746416724d77ce3839eb8248b5219d39a1145.

**#14 - 2014-12-14 15:01 - Gerrit Code Review**

- *Status changed from Resolved to Under 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/35412>

**#15 - 2014-12-14 15:30 - Markus Klein**

- *Status changed from Under Review to Resolved*

Applied in changeset commit:2607711179ccc232eeda705f446317ec1e1d8cd3.