

Core - Bug # 58004

Status:	Resolved	Priority:	Must have
Author:	Philipp Wrann	Category:	Categorization API
Created:	2014-04-17	Assigned To:	
Updated:	2014-04-30	Due date:	
TYPO3 Version:	6.2		
PHP Version:	5.3		
Complexity:	no-brainer		
Is Regression:	Yes		
Sprint Focus:			
Subject:	Categories not visible		
Description			
Since last update i cant see the categories tree anymore.			
I added the TCA definitions like this: \\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable			
Related issues:			
related to Core - Bug # 57869: Remove side effects form TCA cache building		Rejected	2014-04-12
related to Core - Task # 57942: Provide API to add cached TCA changes		Resolved	2014-04-15
related to Core - Bug # 58384: Installing distributions fails		Resolved	2014-05-01
related to Core - Bug # 58620: Category tab is missing in file metadata in 6.2.2		Resolved	2014-05-08

Associated revisions

Revision 4b3e3cce - 2014-04-30 00:26 - Helmut Hummel

[BUGFIX] Fix oddities of the CategoryRegistry

This change makes it again possible to call makeCategorizable from ext_tables.php files.

This is done by directly applying additions to the TCA if the default TCA has already been applied.

For details read the comparison matrix in the referenced bug report.

Resolves: #58004

Releases: 6.2

Change-Id: I0cb69d0421a0df3f930cc9cac1b1811108572530

Reviewed-on: <https://review.typo3.org/29560>

Reviewed-by: Wouter Wolters

Reviewed-by: Frans Saris

Tested-by: Frans Saris

Reviewed-by: Steffen Ritter

Tested-by: Steffen Ritter

Reviewed-by: Helmut Hummel

Tested-by: Helmut Hummel

History

#1 - 2014-04-17 11:59 - Philipp Wrann

This is the concrete call in ext_tables.php:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable($_EXTKEY,'tx_regionalobject_domain_model_regionalobject','categories'  
;
```

the field exists in ext_tables.sql and of course in the database

#2 - 2014-04-17 12:14 - Philipp Wrann

I think the TCA definitions are just built too early.

#3 - 2014-04-17 13:22 - Stefan Neufeind

He meant: works with 6.2.0, broke since 6.2.1. Might be related to the introduction of TCA-overrides (#57942) though that should not break existing things.

#4 - 2014-04-17 13:36 - Steffen Ritter

- Status changed from New to Accepted

confirmed

#5 - 2014-04-17 13:55 - Ernesto Baschny

- Is Regression changed from No to Yes

#6 - 2014-04-17 15:47 - Philipp Wrann

- File example.zip added

I added the TCA definitions like in the example attached.

#7 - 2014-04-17 17:22 - Andreas Allacher

Same problem. Most likely related to overrides.

Because from Bootstrap->executeExtTablesAdditionalFile() the applyTca() is gone.

It is now done in applyTcaOverrides.

However instead of using

```
\TYPO3\CMS\Core\Category\CategoryRegistry::getInstance()
```

The registry is injected and is therefore a different instance than the one where makeCategorizable adds the category information.

#8 - 2014-04-17 19:58 - Stefan Neufeind

CategoryRegistry is a singleton. So I guess you should receive the same instance.

#9 - 2014-04-18 00:54 - Helmut Hummel

- Status changed from Accepted to Needs Feedback

Hey guys, thanks for your feedback and sorry for any inconvenience caused by this change.

However, calling `\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable` in `ext_tables.php` is wrong and instead the call should be made in `ext_localconf.php`

It is true nonetheless that having it in `ext_tables.php` worked in TYPO3 6.1 and also 6.2.0

It worked because `CategoryRegistry->applyTca` was called so late in the bootstrap, that it was **after** `ext_tables.php` files were included **and after** the `extTables.php` file was read.

That meant that the resulting TCA could related to categories could not be modified **at all**

Additionally, adding to the registry and applying the TCA changes of the registry could not be cached, although the result was/is pretty "static"

The thing is, that the design of `\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable` changed since TYPO3 6.1 to a pure registry with no side effects when adding categorization requests. Such API is not something to be placed in `ext_tables.php` because in this file **direct manipulation** of global state (e.g. TCA) is required to be effective. Since `\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable` does **not** manipulate the global TCA. Thus, such call has to go into `ext_localconf.php`. It has been made to work by putting it late into the bootstrap, but that had the downsides described above.

My recommendation: put the `\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable` call in `ext_localconf.php` as this will work in 6.1.x and 6.2.0 if you need backwards compatibility (you can also add it to `ext_tables.php` if you do not care about logged warnings in 6.1.x or wrap it in a condition to only call if version lower than 6.1).

If you want the benefit of caching and do not need backwards compatibility to 6.1, then put it in a *.php file (choose a name as you like) in `<extKey>/Configuration/TCA/Overrides/` (see #57951)

Let me elaborate on the current order of execution during bootstrap:

1. `ext_localconf.php` files from all extensions are read
2. `<extKey>/Configuration/TCA/<table_name>.php` files are read and put into TCA[`<table_name>`]
3. `<extKey>/Configuration/TCA/Overrides/<name>.php` are read (API calls in there are meant to manipulate TCA and nothing else is useful as these files are only included when TCA is not cached)
4. `CategoryRegistry` changes are applied
5. Complete TCA is cached
6. `ext_tables.php` files from all extensions are read (Here still TCA can be manipulate for BC reasons or e.g. for changing category registry related things)
7. `extTables.php`

I post here my suggestions already made on <https://review.typo3.org/29484> If you have another option or proposal what we should do in this area please add your comments:

This is a limitation in the category registry design. We can do the following:

1. Apply category registry TCA **before** override TCA from extensions are included. Then the `makeCategorizable` calls from extension can only work in `ext_localconf.php` files and are executed in every BE and FE request.
2. Kick the registry design and apply TCA changes on every `->add` call directly and apply default core categorization before TCA overrides are included. by doing so `makeCategorizable` must **not** be placed in `ext_localconf.php` files as this would not work any more, but default TCA can be overridden and still cached.
3. Leave it like that, as it is better than before (except that now `makeCategorizable` will not work in `ext_tables.php` and `extTables.php` any more, but only in TCA overrides and `ext_localconf.php` files)
4. any other option?

#10 - 2014-04-18 08:44 - Philipp Wrann

Just my thoughts on that:

A functional change between one big release and a patch (eg. 6.2.0 -> 6.2.1) should never take place, mark it as deprecated and use the new approach in 6.3

If someone uses the new approach the benefits will be given i think, so ensure backwards compatibility within version 6.2. All Extensions that work with 6.2.0 should also work with 6.2.*.

Splitting up TCA definitions in 2 files anyway is bad, (i hated that weird dynamicConfigFile bullshit too) especially for new developers. Maybe it would be best to have something like that in your tables TCA:

```
$TCA['table']['columns']['categories']['config'] = ExtensionManagementUtility::getCategoryFieldTCAConfig('table','categories',array());
```

Like you can use:

```
ExtensionManagementUtility::getFileFieldTCAConfig
```

#11 - 2014-04-18 10:57 - Helmut Hummel

Philipp Wrann wrote:

A functional change between one big release and a patch (eg. 6.2.0 -> 6.2.1) should never take place, mark it as deprecated and use the new approach in 6.3
If someone uses the new approach the benefits will be given i think, so ensure backwards compatibility within version 6.2. All Extensions that work with 6.2.0 should also work with 6.2..*

I totally agree in general, but in this case, how should it have been done?

The category fields added were not changeable because applyTca was executed **after** ext_tables.php, if we move applyTca before ext_tables.php then we could change category fields there again, but cannot add them in ext_tables.php because TCA is already applied. See the chicken/egg problem here?

Would be great if you had an idea how to theoretically fix that without breaking something, then I'm happy to implement it.

This really is a chicken/egg problem: applyTca before ext_tables.php then you cannot register any more, applyTca after ext_tables.php then you cannot modify the applied TCA any more.

So where should we put it without breaking something?

Splitting up TCA definitions in 2 files anyway is bad, (i hated that weird dynamicConfigFile bullshit too) especially for new developers.
Maybe it would be best to have something like that in your tables TCA:

```
$TCA['table']['columns']['categories']['config'] = ExtensionManagementUtility::getCategoryFieldTCAConfig('table','categories',array());
```

Like you can use:

```
ExtensionManagementUtility::getFileFieldTCAConfig
```

Yes. This is what should have been done, it hasn't because the goal has been to also create the database field automatically (I guess). Nevertheless such API would be good to have in exchange (or in addition) to the current API.

#12 - 2014-04-18 11:35 - Helmut Hummel

Helmut Hummel wrote:

Splitting up TCA definitions in 2 files anyway is bad, (i hated that weird dynamicConfigFile bullshit too) especially for new developers. Maybe it would be best to have something like that in your tables TCA:

```
$TCA['table']['columns']['categories']['config'] = ExtensionManagementUtility::getCategoryFieldTCAConfig('table','categories',array());
```

Like you can use:

```
ExtensionManagementUtility::getFileFieldTCAConfig
```

Yes. This is what should have been done, it hasn't because the goal has been to also create the database field automatically (I guess). Nevertheless such API would be good to have in exchange (or in addition) to the current API.

Ok, thought about this again. The scope of something like `ExtensionManagementUtility::getCategoryFieldTCAConfig` and `ExtensionManagementUtility::makeCategorizable` is really different.

If you introduce a new table and want to have it categorizable, then just write the TCA for it (currently no support for you in that regard) and put it into `Configuration/TCA/<table_name>.php` and you are set.

Do not use `ExtensionManagementUtility::makeCategorizable` when introducing tables yourself.

`ExtensionManagementUtility::makeCategorizable` fits perfectly to easily add a category field to an **existing** table. This can be done in `ext_localconf.php` and `/Configuration/TCA/Overrides/`

However I would be fine introducing `ExtensionManagementUtility::getCategoryFieldTCAConfig` in 6.2.x if somebody wants to do this.

#13 - 2014-04-18 11:43 - Stefan Neufeind

When I write something that should be categorizable maybe I don't want to think about what TCA makes it categorizable but just use the function for that? That's also how extension-builder (soon to be released version) does it - independent of whether it's an existing table or if you just defined it in the very same class.

I guess you mean it just won't work? So if you define your own TCA **and** want to use `makeCategorizable`, would the `makeCategorizable()` then need to go into a separate TCA-overrides-file (inside the same extension) to make it work?

#14 - 2014-04-18 12:14 - Philipp Gampe

Ok, thought about this again. The scope of something like `ExtensionManagementUtility::getCategoryFieldTCAConfig` and `ExtensionManagementUtility::makeCategorizable` is really different.

If you introduce a new table and want to have it categorizable, then just write the TCA for it (currently no support for you in that regard) and put it into `Configuration/TCA/<table_name>.php` and you are set.

I agree. I think we need both functions. One for adding TCA to an existing table and one to help extension authors to provide category fields themselves as part of their extension (e.g. `tx_news`).

#15 - 2014-04-18 15:45 - Helmut Hummel

Stefan Neufeind wrote:

|

When I write something that should be categorizable maybe I don't want to think about what TCA makes it categorizable but just use the function for that?

True, but this never worked.

That's also how extension-builder (soon to be released version) does it - independent of whether it's an existing table or if you just defined it in the very same class.

We need to fix the core first before we can say what code the extension builder should create.

I guess you mean it just won't work? So if you define your own TCA **and** want to use makeCategorizable, would the makeCategorizable() then need to go into a separate TCA-overrides-file (inside the same extension) to make it work?

Yes. Introducing a new table and at the same time making it categorizable never worked.

#16 - 2014-04-18 16:20 - Helmut Hummel

[EDIT] Behavior also changed from 6.1.0 to 6.1.1

Here a matrix that will help on changes in that area. Any change should add more + than we had before.

Judging the change introduced in 6.2.1: it added many + but to do so we had to give up one, that is use makeCategorizable in ext_tables.php

Features	TYPO3 6.0	TYPO3 6.1.0	TYPO3 6.1.1	TYPO3 6.2.0	TYPO3 6.2.1
use makeCategorizable in ext_tables.php	+	+	+	+	-
use makeCategorizable in ext_localconf.php	-	-	+	+	+
use makeCategorizable in Overrides/file.php	-	-	-	-	+
override TCA of makeCategorizable	+	+	-	-	+
use makeCategorizable for newly introduced table1	-	-	+	+	+
use makeCategorizable for new table in same TCA file	-	-	-	-	-
have TCA of makeCategorizable cached	-	-	-	-	+
have changes to TCA of makeCategorizable cached4	-	-	-	-	-
database definition	(+)[2]	(+)[2]	(+)[2]	+	+

automatically added					
defaultCategorizedTables	-	+	+	+	+
defaultCategorizedTables overridable	-	-	-	-	+ ^[3]
defaultCategorizedTables cachable	-	-	-	-	+
defaultCategorizedTables overrides cachable	-	-	-	-	-
API to return category field configuration	-	-	-	-	-

[1] In same extension

[2] Was buggy for EM

[3] in ext_tables.php

[4] in 6.2.1 changes to TCA created by CategoryRegistry can only be overridden in ext_tables.php

#17 - 2014-04-18 19:09 - Helmut Hummel

[EDIT] [6] and [7] is fixed with #58039

[EDIT-2] Added failing case to table

[EDIT-3] Fixed failing case to table

Features	TYPO3 6.0	TYPO3 6.1.0	TYPO3 6.1.1	TYPO3 6.2.0	TYPO3 6.2.1	TYPO3 6.2.2-patch
use makeCategorizable in ext_tables.php	+	+	+	+	-	+
use makeCategorizable in ext_localconf.php	-	-	+	+	+	+
use makeCategorizable in Overrides/file.php	-	-	-	-	+	+
override TCA of makeCategorizable	+	+	-	-	+	+
use makeCategorizable for newly introduced table1	-	-	+	+	+	+
makeCategorizable new table, old stlye "dynamic" TCA1	-	-	+	+	-	+ ^[8]
use makeCategorizable for new table in same TCA file	-	-	-	-	-	+ ^[5]
have TCA of	-	-	-	-	(+) ^[7]	+

makeCategorizable cached						
have changes to TCA of makeCategorizable cached4	-	-	-	-	-	+
database definition automatically added	(+)[2]	(+)[2]	(+)[2]	+	+	+
defaultCategorized Tables	-	+	+	+	+	+
defaultCategorized Tables overridable	-	-	-	-	+ [3]	+
defaultCategorized Tables cachable	-	-	-	-	+	+
defaultCategorized Tables overrides cachable	-	-	-	-	-	+
API to return category field configuration	-	-	-	-	-	+ [5]

[1] In same extension

[2] Was buggy for EM

[3] in ext_tables.php

[4] in 6.2.1 changes to TCA created by CategoryRegistry can only be overridden in ext_tables.php

[5] new API: CategoryRegistry::getTcaFieldConfiguration()

[6] not the case for new API and also not the case when makeCategorizable is used in Overrides/file.php

[7] This is true when calling makeCategorizable from Overrides/file.php which has the downside, that the DB field will not be added during DB compare any more and has to be added manually

[8] TCA ctrl section of table must be defined before calling makeCategorizable !

#18 - 2014-04-18 19:50 - Gerrit Code Review

- Status changed from Needs Feedback 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/29560>

#19 - 2014-04-18 21:09 - 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 <https://review.typo3.org/29560>

#20 - 2014-04-18 21:31 - 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 <https://review.typo3.org/29560>

#21 - 2014-04-18 23:02 - 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 <https://review.typo3.org/29560>

#22 - 2014-04-21 02:03 - Gerrit Code Review

Patch set 5 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/29560>

#23 - 2014-04-23 21:15 - Gerrit Code Review

Patch set 6 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/29560>

#24 - 2014-04-30 00:31 - Helmut Hummel

- *Status changed from Under Review to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:4b3e3cce4f1bdcedc7f6ee23fb1bea952569a5ca.

Files

example.zip	2.2 kB	2014-04-17	Philipp Wrann
-------------	--------	------------	---------------