

## TYPO3 Core - Bug #91430

Task # 86141 (Closed): Remove superfluous database constraint in DataMapProcessor

### Unrelated tt\_content associated to a news after translating it

2020-05-18 14:35 - Xavier Perseguers

<b>Status:</b>	Closed	<b>Start date:</b>	2020-05-18
<b>Priority:</b>	Should have	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Complexity:</b>	medium
<b>TYPO3 Version:</b>	8	<b>Is Regression:</b>	
<b>PHP Version:</b>		<b>Sprint Focus:</b>	
<b>Tags:</b>			

#### Description

Context is the following:

- Context: TYPO3 v8, EXT:news v7.3.1
- tt\_content allowed for news items
- Multilingual website
- No workspaces

#### How-to reproduce

- #. Create a news record, fill-in "title" and create an empty CE (e.g. text & images), save and close
- #. Click the button to translate to another language (in my case French, sys\_language\_uid = 2)
- #. DO NOT TOUCH ANYTHING, within the "content elements" I have a relation to some completely other tt\_content element

#### Analysis

I dumped the tables tt\_content and tx\_news\_domain\_model\_news before creating the news record, so that I can rollback and start again. Here is what happens exactly:

- My news in default language has uid=4773
- My tt\_content related element has uid=31995

Useful columns of this tt\_content row:

- uid = 31995
- pid = 7715
- t3\_origuid = 0
- CType = textmedia
- sys\_language\_uid = 0
- l18n\_parent = 0
- tx\_news\_related\_news = 4773
- l10n\_source = 0
- l10n\_state = NULL

The record that is referenced in my translated news is following:

- uid = 31842
- pid = 9074
- t3\_origuid = 0
- CType = textmedia
- sys\_language\_uid = 2
- l18n\_parent = 28440
- tx\_news\_related\_news = 3525
- l10n\_source = 0
- l10n\_state = NULL

I tracked down the bug to method \TYPO3\CMS\Core\DataHandling\Localization\DataMapProcessor::resolveAncestorId().

This method is called from \TYPO3\CMS\Core\DataHandling\Localization\DataMapProcessor::synchronizeInlineRelations() when preparing \$dependentIdMap with the call to \TYPO3\CMS\Core\DataHandling\Localization\DataMapProcessor::fetchDependentIdMap() with following parameters:

```
$tableName = 'tt_content'
```

```
$ids = [  
    0 => 31995  
]
```

```
$desiredLanguage = 2
```

The call to fetchTranslatedValues() is done with

```
$tableName = 'tt_content'
```

```
$fieldNames = [  
    'l10n_state' => 'l10n_state',  
    'language' => 'sys_language_uid',  
    'parent' => 'l18n_parent',  
    'source' => 'l10n_source',  
    'uid' => 'uid'  
]
```

```
$ids = [  
    0 => 31995  
]
```

Output of this method is:

```
$translationValues = [  
    31995 => [  
        'l10n_source' => 0,  
        'l10n_state' => null,  
        'l18n_parent' => 0,  
        'sys_language_uid' => 0,  
        'uid' => 31995  
    ]  
]
```

Then there's a call to buildElementAncestorIdMap() which will finally call resolveAncestorId with following parameters:

```
$fieldNames = [  
    'l10n_state' => 'l10n_state',  
    'language' => 'sys_language_uid',  
    'parent' => 'l18n_parent',  
    'source' => 'l10n_source',  
    'uid' => 'uid'  
]
```

```
$element = [  
    'l10n_source' => 0,  
    'l10n_state' => null,  
    'l18n_parent' => 0,  
    'sys_language_uid' => 0,  
    'uid' => 31995  
]
```

First block of the method is not evaluated:

```
    // implicit: having source value different to parent value, use source pointer  
    if (  
        !empty($fieldNames['source'])  
        && $element[$fieldNames['source']] !== $element[$fieldNames['parent']]  
    ) {
```

```
        return (int)$fieldNames['source'];
    }
}
```

Thus next block is tested: the one that does:

```
if (!empty($fieldNames['parent'])) {
    // implicit: use parent pointer if defined
    return (int)$element[$fieldNames['parent']];
}
```

`$fieldNames['parent'] = 'l18n_parent'` thus it is evaluated, it thus takes `$element['l18n_parent']` which is not set (thus leads to null) and casts it to int which means 0.

This leads to method `buildElementAncestorIdMap` to test 0 as not null and thus add this ancestor to the mapping:

```
$ancestorIdMap = [
    0 => [
        0 => 31995
    ]
];
```

this leads to having the dependent elements being uids = 31995 and 0.

Now the call to `fetchDependentElements` will build following query:

```
SELECT
    `uid`, `l10n_state`, `sys_language_uid`, `l18n_parent`, `l10n_source`
FROM
    `tt_content`
WHERE (`sys_language_uid` > 0) AND (`l18n_parent` > 0) AND ((`l18n_parent` IN (31995,0)) OR (`l10n_source` IN (31995,0))) AND ((`tt_content`.`deleted` = 0) AND ((`tt_content`.`t3ver_wsid` = 0) AND (`tt_content`.`pid` <> -1)));
```

which results in

uid	l10n_state	sys_language_uid	l18n_parent	l10n_source
31904	NULL	2	24951	0
25577	NULL	3	24951	0
25576	NULL	4	24951	0
26764	NULL	1	26761	0
26767	NULL	3	26761	0
26770	NULL	4	26761	0
26765	NULL	1	26762	0
26771	NULL	4	26762	0
26766	NULL	1	26763	0
26774	NULL	1	26773	0
31842	NULL	2	28440	0
31844	NULL	3	28440	0
31843	NULL	4	28440	0
31848	NULL	3	31845	0
31847	NULL	4	31845	0

In turn, every element in that array will be tested for the ancestorId and if that id is found in the dependant elements (31995 and 0) then it is kept. We clearly sees that this 0 is going to hurt us:

After the first loop, we have:

```
$dependantIdMap = [
    31995 => 31904
];
```

A few records are skipped because not of the targeted language 2, then the map is updated to be

```
$dependantIdMap = [
    31995 => 31842
];
```

and stays like that because there are no more matching records with targeted language.

And BANG! we have a totally wrong dependent object because of that "0" which was wrongly returned.

## Solution

We must ensure we don't cast a null (non-existing source of information) to 0 in the first place. The patch is thus:

```
diff --git a/typo3/sysex/core/Classes/DataHandling/Localization/DataMapProcessor.php b/typo3/sysex/core/Classes/DataHandling/Localization/DataMapProcessor.php
index 66ffd16d88..02deed2d33 100644
--- a/typo3/sysex/core/Classes/DataHandling/Localization/DataMapProcessor.php
+++ b/typo3/sysex/core/Classes/DataHandling/Localization/DataMapProcessor.php
@@ -1213,7 +1213,7 @@ class DataMapProcessor
    ) {
        return (int)$fieldNames['source'];
    }
-   if (!empty($fieldNames['parent'])) {
+   if (!empty($fieldNames['parent']) && isset($fieldNames[$fieldNames['parent']])) {
        // implicit: use parent pointer if defined
        return (int)$element[$fieldNames['parent']];
    }
}
```

### Related issues:

Has duplicate TYPO3 Core - Bug #94056: Selecting wrong translation while sync...

Closed

2021-05-04

## History

### #1 - 2020-05-18 14:43 - 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/c/Packages/TYPO3.CMS/+64515>

### #2 - 2020-05-19 09:24 - Xavier Perseguers

This patch has a side effect to totally mess up the file relations when a news has no content elements but 1 file relation (banner) and is then translated. The translation on a dev machine never (?) ends and duplicates 1000+ times the file relation.

Additional info:

```
config.sys_language_mode = content_fallback
config.sys_language_overlay = 1
```

### #3 - 2020-05-25 08:38 - 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/c/Packages/TYPO3.CMS/+64515>

### #4 - 2020-05-25 20:54 - 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/c/Packages/TYPO3.CMS/+64515>

### #5 - 2020-10-06 22:35 - 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/c/Packages/TYPO3.CMS/+64515>

### #6 - 2020-11-10 14:48 - Markus Klein

In the description you have

```
$element = [
    'l10n_source' => 0,
    'l10n_state' => null,
    'l18n_parent' => 0,
    'sys_language_uid' => 0,
    'uid' => 31995
]
```

and then you write

it thus takes \$element['l18n\_parent'] which is not set

That's a contradiction. Either your input data is wrong here in the ticket, or your conclusion.

Second question: The mentioned side effect is still a problem with the latest patch set?

**#7 - 2020-11-12 09:16 - Xavier Perseguers**

Mmh, yes, you're right. It is actually set, strange, I just tested again and indeed I didn't write it wrong. So the actual bug is not "unset" but "empty". This is really the problem.

**#8 - 2020-11-12 09:38 - 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/c/Packages/TYPO3.CMS/+64515>

**#9 - 2021-05-04 23:01 - Alexander Opitz**

- Has duplicate Bug #94056: *Selecting wrong translation while synchronizeInlineRelations added*

**#10 - 2021-05-05 07:14 - 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/c/Packages/TYPO3.CMS/+64515>

**#11 - 2021-05-05 07:29 - Gerrit Code Review**

Patch set 7 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server. It is available at <https://review.typo3.org/c/Packages/TYPO3.CMS/+64515>

**#12 - 2021-05-05 07:39 - Alexander Opitz**

As IMHO not only the issue with empty/unset ['source']/['parent'] field we also have the wrong

```
return (int)$fieldNames['source'];
```

which leads everytime, when parent !== source, to an int(0) and as followup inside fetchDependentElements can produce wrong results (as Xavier explained it above).

**#13 - 2021-05-05 09:22 - Oliver Hader**

- Parent task set to #86141

**#14 - 2021-05-05 09:23 - Oliver Hader**

- Status changed from Under Review to Closed

→ please see [#86141](#)