

## TYPO3 Core - Bug #45873

### querySettings setRespectSysLanguage or setSysLanguageUid does not work

2013-02-27 01:29 - Dominic Garms

<b>Status:</b>	Closed	<b>Start date:</b>	2013-02-27
<b>Priority:</b>	Must have	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Extbase + l10n	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Candidate for Major Version	<b>Complexity:</b>	
<b>TYPO3 Version:</b>	6.2	<b>Is Regression:</b>	No
<b>PHP Version:</b>		<b>Sprint Focus:</b>	
<b>Tags:</b>			

#### Description

Hello,  
currently its not possible to get a translated record after update from 4.7 to 6.0 or 6.1. This was working in 4.7:

```
$query = $this->createQuery();
$query->getQuerySettings()->setRespectSysLanguage(false);
$result = $query->matching(
    $query->logicalAnd($query->equals('sys_language_uid', $translatedLangUid), $query->equals('l10n_
parent', $uidOfDefaultLang))
    ->execute();
$myObject = $result->getFirst();
```

In 6.x its currently impossible to get the translated record with the data query, because the function in typo3DBbackend.php doLanguageAndWorkspaceOverlay() always overlays the record with the default language record.

If I try to set setSysLanguageUid(\$translatedLangUid) then I get the translated record but weirdly the UID gets overwritten with the default language UID and inside of the object I find \_localizedUid and \_languageUid. As I can't reach this two attributes I have no idea how to correctly update or add the object into my repository.

As in 4.7 I could get this working I think it is a bug, or some major changes happens where I couldn't find any information.

Regards,  
Dominic

#### Related issues:

Related to TYPO3 Core - Bug #32216: OrderBy on translated records doesn't wor...	<b>Closed</b>	<b>2011-11-30</b>
Related to TYPO3 Core - Bug #47192: setRespectSysLanguage(FALSE) doesn't prev...	<b>Closed</b>	<b>2013-04-13</b>
Related to TYPO3 Core - Bug #56855: Extbase tries to overlay pages_language_o...	<b>Closed</b>	<b>2014-03-13</b>
Related to TYPO3 Core - Bug #59992: Persistence session doesn't take overlays...	<b>Accepted</b>	<b>2014-06-30</b>
Related to TYPO3 Core - Bug #86405: querySetting setRespectSysLanguage (wrong...	<b>New</b>	<b>2018-09-27</b>

#### Associated revisions

##### Revision 8d4fe802 - 2018-09-20 15:39 - Tymoteusz Motylewski

[TASK] Provide more tests for Extbase rendering in languages

Some tests are just documenting current, buggy behavior.  
So once bugs are fixed assertions needs to be updated.  
See inline comments for details (and list of related issues below).

Resolves: #86327  
Related: #45873  
Related: #68672  
Releases: master  
Change-Id: I780f77fa95d3824a2dafd58a1a73df2bb5fac9e4  
Reviewed-on: <https://review.typo3.org/58276>  
Tested-by: TYPO3com <[no-reply@typo3.com](mailto:no-reply@typo3.com)>  
Reviewed-by: Anja Leichsenring <[leichsenring@ab-softlab.de](mailto:leichsenring@ab-softlab.de)>

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

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

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

## Revision 9089da60 - 2018-09-21 20:36 - Tymoteusz Motylewski

[TASK] Provide more tests for Extbase rendering in languages

Some tests are just documenting current, buggy behavior.  
So once bugs are fixed assertions needs to be updated.  
See inline comments for details (and list of related issues below).

Resolves: #86327

Related: #45873

Related: #68672

Releases: master

Change-Id: I780f77fa95d3824a2dafd58a1a73df2bb5fac9e4

Reviewed-on: <https://review.typo3.org/58357>

Tested-by: TYPO3com <[no-reply@typo3.com](mailto:no-reply@typo3.com)>

Reviewed-by: Markus Klein <[markus.klein@typo3.org](mailto:markus.klein@typo3.org)>

Tested-by: Markus Klein <[markus.klein@typo3.org](mailto:markus.klein@typo3.org)>

Reviewed-by: Tymoteusz Motylewski <[t.motylewski@gmail.com](mailto:t.motylewski@gmail.com)>

Tested-by: Tymoteusz Motylewski <[t.motylewski@gmail.com](mailto:t.motylewski@gmail.com)>

## History

---

### #1 - 2013-03-03 09:36 - Anja Leichsenring

- Target version changed from 6.0.3 to Extbase 6.1

### #2 - 2013-04-10 04:32 - Dominic Garms

Here some more detailed information:

```
$logicAnd[] = $query->equals('city.country', $country);  
$logicAnd[] = $query->equals('travelType', $travelType);
```

\$travelType and \$country is an object of the correct type form their model. The travelType equals statement works in default language and all other languages.

But the joined city.country is not working for any other language than the default language.

This worked in 4.7 and does not work anymore in 6.0.4

### #3 - 2013-04-12 13:19 - Georg Ringer

- Subject changed from *querySettings setRespectSysLanguage or setSysLanguageUid does not work in 6.0 and 6.1* to *querySettings setRespectSysLanguage or setSysLanguageUid does not work in 6.0 and 6.1*

### #4 - 2013-04-12 14:05 - Alexander Schnitzler

- Status changed from New to Needs Feedback

### #5 - 2013-04-23 22:59 - Alexander Schnitzler

- Target version changed from Extbase 6.1 to Extbase 6.2

### #6 - 2013-04-24 08:14 - Johannes C. Schulz

I also run into this problem.

Our site has german and english objects. On other language-pages I need the english objects, but the doLanguageAndWorkspaceOverlay only gives me the german. I tried to force the query giving me english objects, but it doesn't work:

```
if($GLOBALS['TSFE']->sys_language_uid != 0){  
    $query->getQuerySettings()->setRespectSysLanguage(FALSE);  
    $query->getQuerySettings()->setSysLanguageUid(1);  
    $object = $query->matching($query->equals('l18n_parent',$hotel))->execute()->getFirst();  
    return $object;  
}
```

Only if I delete row 281 from Typo3DbBackend.php the needed functionality is provided.  
Please fix as soon as possible!

Johannes

**#7 - 2013-05-01 22:14 - Alexander Schnitzler**

@Felix: Can you drop a line on this one?

**#8 - 2013-05-31 15:28 - Anja Leichsenring**

- Target version changed from Extbase 6.2 to Extbase 6.3

**#9 - 2013-06-09 12:16 - Alexander Schnitzler**

- Target version changed from Extbase 6.3 to Extbase 6.2

**#10 - 2013-06-30 11:02 - Alexander Schnitzler**

- Status changed from Needs Feedback to Accepted

Testet and confirmed as bug.

**#11 - 2013-06-30 11:34 - Gerrit Code Review**

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/21744>

**#12 - 2013-07-01 17:34 - Marc Bastian Heinrichs**

- Status changed from Under Review to Needs Feedback

Dominic or Johannes, could you post your typoscript config for language overlay, please?

**#13 - 2013-07-01 20:37 - Helmut Hummel**

```
$query = $this->createQuery();
$query->getQuerySettings()->setRespectSysLanguage(false);
$query->getQuerySettings()->setRespectStoragePage(false);
$query->getQuerySettings()->setSysLanguageUid(1);
$result = $query->matching(
    $query->logicalAnd(
        $query->equals('sys_language_uid', 1),
        $query->equals('l10n_parent', 1)
    )
)->execute();
```

works fine for me. Please note that in contrary to Extbase 4.7 the following line is needed since Extbase 6.0 if the requested page language is **not** set to the desired translation to be fetched (which is uncommon but there might be usecases for that):

```
$query->getQuerySettings()->setSysLanguageUid(1);
```

There is still a difference in the handling as Extbase before 6.0 did only perform an overlay on non translated records. Now if a translated record is fetched, Extbase fetches the original language record and performs an overlay with the language id set in the query settings.

This is the related change: <https://review.typo3.org/10188>

**#14 - 2013-07-02 01:54 - Dominic Garms**

Hello,

here my language setup:

```
config {
    linkVars = L
    sys_language_uid = 0
    language = de
    locale_all = de_DE.UTF-8
    htmlTag_langKey = de-DE
    sys_language_mode = content_fallback
    sys_language_overlay = hideNonTranslated
}
[globalVar = GP:L = 4]
```

```
config.sys_language_uid = 4
config.language = en
config.locale_all = en-GB.UTF-8
config.htmlTag_langKey = en-GB
[global]
```

This is my usecase:

Model A

-->relation n:1 Model B

I have an admin tool where the fe user should be able to translate records of A and also select their object B. Because of the overlay, even if I set setSysLanguage(4) its for me currently impossible to add B, its only getting related to the sys\_language(0).

Regards,  
Dominic

#### #15 - 2013-07-02 08:07 - Johannes C. Schulz

Hello

my language-setup:

```
config {
    htmlTag_langKey = de
    metaCharset = utf-8
    linkVars = l,m
    sys_language_uid = 0
    language = de
    locale_all = de_DE
    [...]
}
# English
[globalVar = GP:l = 1] && [PIDupinRootline = {$pid_tourismus}]
    config.sys_language_uid = 1
    config.language = en
    config.locale_all = english
[global]

# Francais
[globalVar = GP:l = 2] && [PIDupinRootline = {$pid_tourismus}]
    config.sys_language_uid = 2
    config.language = fr
    config.locale_all = fr_FR
[global]

# Dutch
[globalVar = GP:l = 3] && [PIDupinRootline = {$pid_tourismus}]
    config.sys_language_uid = 3
    config.language = nl
    config.locale_all = nl_NL
[global]
```

Because 75% are only in german, there is no more configuration inside the Main-TS. Inside the subtree with other languages we use TS-Configuration:

```
TCAdefaults.pages.118n_cfg = 2
```

--

@Helmuth: I tried the things in the way you described - for me this not really worked.

#### #16 - 2013-07-12 11:57 - Nicolas Forgeot

In waiting, the bug is resolved, I fix with this :

```
//for each repository
public function initializeObject() {
    $this->defaultQuerySettings->setRespectSysLanguage(TRUE);

    $language = $_GET['L'];
    if($language) {
        $this->defaultQuerySettings->setSysLanguageUid($language);
    }
}
```

```
}  
}
```

But `sys_language_overlay hideNonTranslated` not work anymore. It's induced by the bug or it's an other bug ?

In `Typo3DbBackend`, it not would be judicious to add a condition on the overlay ?

```
protected function addSysLanguageStatement($tableName, array &$sql, $querySettings) {  
    ...  
  
    //this  
    if (isset($GLOBALS['TCA'][$tableName]['ctrl']['transOrigPointerField'])  
        && $querySettings->getSysLanguageUid() > 0  
        && $GLOBALS['TSFE']->sys_language_contentOL != "hideNonTranslated"  
    ) {  
  
    //instead  
    if (isset($GLOBALS['TCA'][$tableName]['ctrl']['transOrigPointerField'])  
        && $querySettings->getSysLanguageUid() > 0  
    ) {  
  
    // add this sql clause :  
    // OR (tx_EXT_domain_model_TABLE.sys_language_uid=0  
    // AND tx_EXT_domain_model_TABLE.uid NOT IN (  
    //     SELECT tx_atolactacontenu_domain_model_fiche.l10n_parent  
    //     FROM tx_EXT_domain_model_TABLE  
    //     WHERE tx_atolactacontenu_domain_model_fiche.l10n_parent>0  
    // AND tx_EXT_domain_model_TABLE.sys_language_uid>0  
    // AND tx_EXT_domain_model_TABLE.deleted=0))  
    }  
}
```

#### #17 - 2013-08-11 22:38 - Felix Oertel

- Tags set to `ecs12ws`

#### #18 - 2013-08-11 22:39 - Felix Oertel

- Status changed from *Needs Feedback* to *Accepted*

#### #19 - 2013-08-11 22:42 - Felix Oertel

- Priority changed from *Should have* to *Must have*

- Tags changed from `ecs12ws` to `ecs13ws`

Right selection of translated records will be part of `#ecs13ws`.

#### #20 - 2013-11-29 12:02 - Marc Steiner

Hello,

We developed an extension to generate a language menu ([http://typo3.org/extensions/repository/view/grds\\_language\\_menu](http://typo3.org/extensions/repository/view/grds_language_menu)).

For this we need to know, which languages the current page is translated to, so that we can add a css class "nonTranslated" to the link if the current page isn't available in that language.

We did this by mapping the table "pages\_language\_overlay" to a class named "Translation". Then we could get all translations for a page by pid from our "TranslationRepository", setting "respectSysLanguage" to false.

This isn't working anymore because of the bug mentioned above.

Although in our (arguably very uncommon) use case the problem goes further. Extbase tries to load the original "pages\_language\_overlay" record with a `sys_language_id` of 0.

But obviously we get an empty result, because the original resides in the table "pages" and not in the table "pages\_language\_overlay".

To make things worse,

```
count($translations)
```

returns the correct number of translations. I guess this is because of the lazyloading feature.

Maybe this is a problem of `typo3/extbase` or maybe we need to change our approach to get translations of a page. Your help would be appreciated.

Regards,  
Marc

**#21 - 2014-01-09 11:42 - Johannes C. Schulz**

@Felix, Anja & Alexander:

What's up with this issue?! Should be fixed on #ecs13ws but wasn't! Will it be fixed sometime or not?

best regards  
Johannes

**#22 - 2014-03-13 16:16 - Stanislas Rolland**

Marc Steiner wrote:

Hello,

We developed an extension to generate a language menu ([http://typo3.org/extensions/repository/view/grds\\_language\\_menu](http://typo3.org/extensions/repository/view/grds_language_menu)). For this we need to know, which languages the current page is translated to, so that we can add a css class "nonTranslated" to the link if the current page isn't available in that language.

We did this by mapping the table "pages\_language\_overlay" to a class named "Translation". Then we could get all translations for a page by pid from our "TranslationRepository", setting "respectSysLanguage" to false.

This isn't working anymore because of the bug mentioned above.

Although in our (arguably very uncommon) use case the problem goes further. Extbase tries to load the original "pages\_language\_overlay" record with a sys\_language\_id of 0.

But obviously we get an empty result, because the original resides in the table "pages" and not in the table "pages\_language\_overlay".

Please test this change: <https://review.typo3.org/#/c/28333/>

Regards,  
Stanislas

**#23 - 2014-09-23 15:10 - Johannes C. Schulz**

Did anyone found a solution of hint?

I tried now with 6.2.4, but it did not work.

Trying to display all translations of one item on a page. Using "statement" sux, but works:

```
$query->statement(' SELECT * FROM tx_news_domain_model_news WHERE (uid="'. $uid. '" OR l10n_parent="'. $uid. '" ) AND uid > 0 AND hidden=0 AND deleted=0 AND (starttime=0 OR starttime<='. time(). ' ) AND ( endtime=0 OR endtime>='. time(). ' ) ORDER BY sys_language_uid ASC');
```

This delivers me all translations of a news-object.

If I try

```
$query->getQuerySettings()->setRespectSysLanguage(FALSE);  
$query->getQuerySettings()->setRespectStoragePage(FALSE);  
$query->matching(  
    $query->logicalOr(  
        $query->equals('uid', $uid),  
        $query->equals('l10n_parent', $uid)  
    )  
);
```

I only get the "german" object two times :-( .

It's sad, to be forced for using "statement"-queries :-(

Johannes

**#24 - 2014-09-24 10:42 - Alexander Opitz**

- Project changed from Extbase MVC Framework to TYPO3 Core
- Category changed from Extbase: Generic Persistence to Extbase
- Status changed from Accepted to New
- Target version changed from Extbase 6.2 to 7.0
- TYPO3 Version set to 6.2
- Is Regression set to No

argh

So this won't be fixed in 6.2?

Is there at least a workaround for the following:

```
$result = $query->statement('
    SELECT
        tx_myext_domain_model_category.*
    FROM
        tx_myext_domain_model_category
    WHERE
        tx_myext_domain_model_category.l10n_parent = ' . (int) $uid . ' AND
        tx_myext_domain_model_category.sys_language_uid = ' . (int) $sys_language_uid .
        \TYPO3\CMS\Backend\Utility\BackendUtility::BEenableFields(
'tx_myext_domain_model_category') .
        \TYPO3\CMS\Backend\Utility\BackendUtility::deleteClause(
'tx_myext_domain_model_category')
    )->execute();
```

None of the following options seem to change the output:  
\$query->getQuerySettings()->setRespectSysLanguage(FALSE);  
\$query->getQuerySettings()->setLanguageOverlayMode(FALSE);  
\$query->getQuerySettings()->setLanguageMode('ignore');  
\$query->getQuerySettings()->setLanguageUid(1);

I always get the default language record instead.  
If i change the query to \$query->getQuerySettings()->setReturnRawQueryResult(TRUE); i will get the expected result.

Background:

We are using two sysFolders (EN,DE) both only have records with sys\_language\_uid=0. We are doing this due to overlay limitations. However related records like this tx\_myext\_domain\_model\_category are still translated and always as sys\_language\_uid=0 attached. Now we would like to fetch this translations in Backend context (CommandController aka Task) but this fails always.

It seems like the record gets correctly translated in

```
/**
 * Returns the object data matching the $query.
 *
 * @param QueryInterface $query
 * @return array
 */
public function getObjectDataByQuery(QueryInterface $query) {
    $statement = $query->getStatement();
    if ($statement instanceof Statement) {
        $rows = $this->getObjectDataByRawQuery($statement);
    } else {
        $rows = $this->getRowsByStatementParts($query);
    }

    $rows = $this->doLanguageAndWorkspaceOverlay($query->getSource(), $rows, $query->getQuerySettings());
    return $rows;
}
```

but is later again translated or the result discharged somewhere here or nearby?

```
/**
 * Maps a single row on an object of the given class
 *
 * @param string $className The name of the target class
 * @param array $row A single array with field_name => value pairs
 * @return object An object of the given class
 */
protected function mapSingleRow($className, array $row) {
    if ($this->identityMap->hasIdentifier($row['uid'], $className)) {
        $object = $this->identityMap->getObjectByIdentifier($row['uid'], $className);
    } else {
        $object = $this->createEmptyObject($className);
        $this->identityMap->registerObject($object, $row['uid']);
        $this->thawProperties($object, $row);
        $object->_memorizeCleanState();
    }
}
```

```

        $this->persistenceSession->registerReconstitutedEntity($object);
    }
    return $object;
}

```

## #26 - 2014-12-23 19:46 - Mathias Schreiber

- Target version changed from 7.0 to 7.1 (Cleanup)

## #27 - 2015-01-23 21:57 - B. St.

Vitoandre is right: The function "mapSingleRow" uses \$row['uid'] to look up an object in the identityMap. So the original record is found, not the one with overlaid data, which has the same uid, but overlaid data. The difference is, that the overlay record row has a key "\_LOCALIZED\_UID" which should be used at this point if set.

We implemented a custom DataMapper class which overwrites the "mapSingleRow" method.

1. HACK extbase in your extension (in your ext\_localconf.php) to register a "CustomQueryResult" class:

```

// The code below is NO PUBLIC API!
/** @var $extbaseObjectContainer \TYPO3\CMS\Extbase\Object\Container\Container */
$extbaseObjectContainer = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance('TYPO3\CMS\Extbase\Object\Container\Container');

$extbaseObjectContainer->registerImplementation('TYPO3\CMS\Extbase\Persistence\QueryResultInterface', 'YOURVENDOR\YOUREXT\Persistence\Storage\CustomQueryResult');
unset($extbaseObjectContainer);

```

2. Implement a simple CustomQueryResult class:

```

class CustomQueryResult extends \TYPO3\CMS\Extbase\Persistence\Generic\QueryResult {

    /**
     * @var \YOURVENDOR\YOUREXT\Persistence\Storage\CustomDataMapper
     * @inject
     */
    protected $dataMapper;

}

```

3. Implement the CustomDataMapper class and overwrite the method "mapSingleRow":

```

class CustomDataMapper extends \TYPO3\CMS\Extbase\Persistence\Generic\Mapper\DataMapper {

    /**
     * Maps a single row on an object of the given class
     *
     * @param string $className The name of the target class
     * @param array $row A single array with field_name => value pairs
     * @return object An object of the given class
     */
    protected function mapSingleRow($className, array $row) {
        $suid = isset($row['_LOCALIZED_UID']) ? $row['_LOCALIZED_UID'] : $row['uid'];
        if ($this->identityMap->hasIdentifier($suid, $className)) {
            $object = $this->identityMap->getObjectByIdentifier($suid, $className);
        } else {
            $object = $this->createEmptyObject($className);
            $this->identityMap->registerObject($object, $suid);
            $this->thawProperties($object, $row);
            $object->_memorizeCleanState();
            $this->persistenceSession->registerReconstitutedEntity($object);
        }
        return $object;
    }

}

```

That did the trick for us! But I would like to know if there is another way to handle this problem, besides patching the 6.2.x source.

## #28 - 2015-01-26 19:24 - Helmut Hummel

Dear reporters. Thanks for the input so far. However I fear we must become more specific in this topic to achieve progress.

Please give feedback in the following form:

=====

## Use case

Short description what your use case is for disabling the sysLanguageHandling.  
Especially explicitly mention if your use case needs to cover only **reading** objects or additionally also **writing** objects

## Minimal (but complete) example code

Please provide **all necessary** code (Models, sql, TCA ...) which is needed to reproduce the issue. The less code it is, the better. Please **only** include code or configuration which is **mandatory** to reproduce the issue

## Describe the actual result you see on your system

## Describe the expected result you want to see

=====

**#29 - 2015-01-26 19:40 - Helmut Hummel**

B. St. wrote:

Vitoandre is right: The function "mapSingleRow" uses \$row['uid'] to look up an object in the identityMap. So the original record is found, not the one with overlaid data, which has the same uid, but overlaid data.

The question is: How did the original record end up in the identity map?

can you provide a use case description and example code like specified above?

3. Implement the CustomDataMapper class and overwrite the method "mapSingleRow":

That did the trick for us! But I would like to know if there is another way to handle this problem, besides patching the 6.2.x source.

Question is: What other use cases does such a change break? At first sight it may look reasonable for your case, but changing that, will break other cases.

The only chance to make progress here is to collect use cases, write them down and lastly nail them down in functional tests

**#30 - 2015-01-27 09:41 - Johannes C. Schulz**

Hello Helmut

Like you wish, I will make again a sketch of my use-case.

## what we need

our site has 4 languages: 0=german, 1=english, 2=french, 3=dutch  
we use an extension for showing hotels, all items are available in german and english  
on french and dutch pages we like to show the english items, but the page around our hotel-item (menu, etc) should be in french or dutch

this works for LIST with following code

```
...
if($GLOBALS['TSFE']->sys_language_uid != 0){
    $query->getQuerySettings()->setRespectSysLanguage(FALSE);
    $query->getQuerySettings()->setSysLanguageUid(1);
    $langsearch = $query->logicalAnd($query->in('sys_language_uid',array(1)));
    $constraint[] = $langsearch;
}
...
```

**but this works not for SINGLE: everytime we get the german items**

```
$query = $this->createQuery();
$query->getQuerySettings()->setRespectStoragePage(FALSE);
if($GLOBALS['TSFE']->sys_language_uid != 0){
    $query->getQuerySettings()->setRespectSysLanguage(FALSE);
    $query->getQuerySettings()->setSysLanguageUid(1);
    $query = $query->matching(
        $query->logicalAnd(
            $query->equals('t3_origuid', $hotel),
```

```

        $query->in('sys_language_uid', array(1))
    )
    );
} else {
    $query = $query->matching(
        $query->equals('uid', $hotel)
    );
}
return $query->execute()->getFirst();

```

## what we need

the possibility to force another language inside the query.

## what happens

if I debug the query it runs fine till doLanguageOverlay overwrites my query by his own and switches back to fallback-language (german).

Hoping my worst english lets you understand what we're looking for. If not, please ask me for more details.

Johannes

### #31 - 2015-02-04 10:54 - martin pfister

hi there

i have currently the same issue.

## use-case

we build a import-extension which loads products from a external sqlite database and stores them as extbase "product"-objects.

there is a automatic import each day. instead of deleting all the objects with every new import, i would like to update them.

while working with the standard language, there is no problem. i use a method like "findOneByProductNr", get the entry, set the desired values and finally update the object.

if i want to get and update the translated objects i run into the problem which is described on this page.

## example code

### *ProductRepository.php*

```

public function findOneByProductNrAndLanguage($productNr, $lang) {
    $query = $this->createQuery();

    $query->getQuerySettings()->setRespectSysLanguage(FALSE);
    $query->getQuerySettings()->setSysLanguageUid($lang);

    $constraints = array();
    $constraints[] = $query->equals('productNr', $productNr);
    $constraints[] = $query->equals('sysLanguageUid', $lang);

    if (!empty($constraints)) {
        $query->matching($query->logicalAnd($constraints));
    }

    return $query->execute()->getFirst();
}

```

## result

the method above only returns the object in the standard language.

## expected result

if i call the method for example with findOneByProductNrAndLanguage(1,2) i want to get the object with productNr 1 in the language 2 (not 0)

edit to be clear: i work in a backend module and i dont have problems to write the objects. i create the translated objects in my Service Class like this:

```

if ($importLang === $standardLang) {
    $product->setL1OnParent(0);
} else {

```

```

$product->setL1OnParent($config['parent']->getUid());
$product->setSysLanguageUid($importLang);
}

```

i can't say if im able to update the object, because i never got a translated and persistent one to work with.

### #32 - 2015-04-09 14:14 - Jonas Eberle

Sorry, I cannot report in more detail right now.

But with `sys_language_mode = strict` the behaviour is like expected. This workaround is ok for us and `content_fallback` is not necessary, I think.

### #33 - 2015-06-11 12:29 - Peter Majmesku

I found an easy ~~hack~~ workaround for my Extbase repository.

Set

```
public $languageUid = 0;
```

in

```
typo3/sysext/extbase/Classes/Persistence/Generic/Typo3QuerySettings.php
```

In

```
typo3_src/typo3/sysext/extbase/Classes/Persistence/Repository.php
```

I set a static property:

```
public static $workaround_languageUid = false;
```

Then I check in `createQuery()`; if `self::$workaround_languageUid` isn't false. If so, I set the `languageUid` in `$this->defaultQuerySettings`.

```

/**
 * Returns a query for objects of this repository
 *
 * @return \TYPO3\CMS\Extbase\Persistence\QueryInterface
 * @api
 */
public function createQuery($language_uid = false) {
    $query = $this->persistenceManager->createQueryForType($this->objectType);
    if ($this->defaultOrderings !== array()) {
        $query->setOrderings($this->defaultOrderings);
    }
    if ($this->defaultQuerySettings !== NULL) {
        /**
         * This static property is a workaround for switching the language uid
         * via the query builder from Extbase.
         *
         * @see https://forge.typo3.org/issues/45873
         */
        if (self::$workaround_languageUid !== false) {
            $this->defaultQuerySettings->languageUid = self::$workaround_languageUid;
        }
        $query->setQuerySettings(clone $this->defaultQuerySettings);
    }
}

return $query;
}

```

In my custom code method I switch `\TYPO3\CMS\Extbase\Persistence\Repository::$workaround_languageUid = 2;` to whatever it should be. Since I don't need instances from `\TYPO3\CMS\Extbase\Persistence\Repository` it does the job.

What do you think?

### #34 - 2015-06-15 17:44 - Benni Mack

- Target version changed from 7.1 (Cleanup) to 7.4 (Backend)

### #35 - 2015-08-05 10:28 - Susanne Moog

- Target version changed from 7.4 (Backend) to 7.5

### #36 - 2015-09-24 19:21 - Benni Mack

- Target version deleted (7.5)

### #37 - 2015-10-01 14:23 - Alex Kellner

It seems still relevant for 6.2 LTS.

In a backend module a customer wants to list records in default language and in separate lines other languages. Playing with querySettings seems not to help

An (very ugly) workaround is to overwrite the TCA languageField. That disables the overlay in extbase. Maybe it helps if someone needs an urgent solution.

```
public function getLocalizedProjectsFromDefaultProject($project) {
    // Disable extbase language overlay if languageField is empty
    $languageField = $GLOBALS['TCA']['tx_projectlist_projects']['ctrl']['languageField'];
    unset($GLOBALS['TCA']['tx_projectlist_projects']['ctrl']['languageField']);

    $query = $this->createQuery();
    $query->getQuerySettings()->setRespectSysLanguage(FALSE);

    $result = $query->execute();

    // Enforce query and enable extbase language overlay from here
    $result->toArray();
    $GLOBALS['TCA']['tx_projectlist_projects']['ctrl']['languageField'] = $languageField;

    return $result;
}
```

### #38 - 2015-10-07 15:11 - Christoph Bessei

As Alex Kellner said this issue is still relevant. I use the following workaround:

```
$query->matching( $query->equals( 'uid', $uid ) );

//https://forge.typo3.org/issues/45873
//Set transOrigPointerTable skip the record overlay (which always returns the default language entry)
$GLOBALS['TCA']['tx_news_domain_model_news']['ctrl']['transOrigPointerTable'] = '';

$result = $query->execute()->getFirst();
```

Maybe this helps someone.

### #39 - 2015-12-14 09:39 - Franz Kugelman

Thanks all! and especially Christoph Bessei,

setting

```
$GLOBALS['TCA']['tx_news_domain_model_news']['ctrl']['transOrigPointerTable'] = '';
```

just for the script-run within my ContentRepository Class solved my problem, as far i can see.

I used a migrateCommandController to manipulate tt\_content records (just once while upgrading to 7.6) and could not manipulate translated tt\_content elements because the uid shown was always the uid of default language. In my case i do not care which language the content is in, i just want to set some fields to different values and leave all the rest as is.

### #40 - 2016-07-29 01:25 - Markus Klein

- Status changed from New to Needs Feedback

Not reading the whole story here, is this still an issue on 7.6?

### #41 - 2016-07-29 09:05 - Johannes C. Schulz

Markus Klein wrote:

Not reading the whole story here, is this still an issue on 7.6?

Hello Markus.

Yes, it's still a problem. Like Mathias Schreiber said, translation-handling will be processed and rewritten in TYPO3 9 or 10 :-(.

#### #42 - 2016-11-07 11:09 - Alexander Opitz

- Status changed from Needs Feedback to New
- Target version set to Candidate for Major Version

#### #43 - 2017-06-27 11:40 - Markus Mächler

I found a related problem:

When e.g. creating a query in a repository with

```
$query = $this->createQuery();
$query->getQuerySettings()
    ->setRespectStoragePage(true)
    ->setLanguageUid($languageId)
    ->setLanguageMode('strict')
    ->setIgnoreEnableFields(true);

return $query->execute();
```

The "->setIgnoreEnableFields(true)" is ignored for fetching the localized record, since in

<https://github.com/TYPO3/TYPO3.CMS/blob/master/typo3/sysext/extbase/Classes/Persistence/Generic/Storage/Typo3DbBackend.php#L615>

the query settings are not passed to the PageRepository, which in turn will still include the enableFields restriction although it should not, as specified in the querySettings

<https://github.com/TYPO3/TYPO3.CMS/blob/master/typo3/sysext/frontend/Classes/Page/PageRepository.php#L542>

Which results in no translated records being fetched when they are hidden.

This is an issue in TYPO3 7.6 and TYPO3 8

A workaround for this is temporarily disabling the TCA enableFields and reenabling it.

```
$tmpEnableFields = $GLOBALS['TCA'][$table]['ctrl']['enablecolumns'];
unset($GLOBALS['TCA'][$table]['ctrl']['enablecolumns']);
```

#### #44 - 2018-02-02 18:46 - Sybille Peters

- File 45873-annotated.png added

I can still reproduce this problem with 8.7.9.

Since it's been a while and I am not sure if everyone is talking about the same thing, I'll summarize what I did:

## Reproduce

- create a content element on a page
- translate it **using "linked" not "copied" translation**
- In your extension, setRespectSysLanguage(false)
- Retrieve your records

This problem is not reproducible if you use content elements that have been translated as "copy" (not linked).

## Expected results

Get all 2 records (default language + translated).

## Actual results

Get the content element in the default language twice (same uid)

I have uploaded a simplified demo extension: <https://github.com/sypets/t3i45873/blob/master/Classes/Domain/Repository/ContentRepository.php>

I used it with my steps to reproduce and got this result (see screenshot).

The first table shows the elements on the current page using a simple approach with setRespectSysLanguage(false).

The second table uses the workaround by Alex Kellner, note 37.

#### #45 - 2018-05-22 14:59 - Markus Mächler

Christoph Bessei wrote:

As Alex Kellner said this issue is still relevant.  
I use the following workaround:

[...]

Maybe this helps someone.

As for TYPO3 8 LTS you need to unset transOrigPointerField instead of transOrigPointerTable.

```
$transOrigPointerField = $GLOBALS['TCA']['tx_news_domain_model_news']['ctrl']['transOrigPointerField'];  
$GLOBALS['TCA']['tx_news_domain_model_news']['ctrl']['transOrigPointerField'] = '';
```

```
//execute your query
```

```
$GLOBALS['TCA']['tx_news_domain_model_news']['ctrl']['transOrigPointerField'] = $transOrigPointerField;
```

#### #46 - 2018-09-18 21:36 - Tymoteusz Motylewski

- Related to Bug #59992: Persistence session doesn't take overlays into account added

#### #47 - 2018-09-18 21:39 - Tymoteusz Motylewski

- Category changed from Extbase to Extbase + I10n

@Sybille, @Markus Is this issue then a duplicate of <https://forge.typo3.org/issues/59992> ?

I've prepared a functional test demonstrating the current wrong behavior here:

<https://review.typo3.org/#/c/58276/6/typo3/sysext/extbase/Tests/Functional/Persistence/QueryLocalizedDataTest.php>

test "postsWithoutRespectingSysLanguage"

So when fixing this issue the test need to be adapted.

#### #48 - 2018-10-03 22:09 - Tymoteusz Motylewski

- Related to Bug #86405: querySetting setRespectSysLanguage (wrong implementation of "strict"-mode) added

#### #49 - 2018-10-07 14:34 - Sybille Peters

- Subject changed from querySettings setRespectSysLanguage or setSysLanguageUid does not work in 6.0 and 6.1 to querySettings setRespectSysLanguage or setSysLanguageUid does not work

#### #50 - 2019-03-24 14:24 - Tymoteusz Motylewski

- Status changed from New to Resolved

Should be resolved with new consistent language handling available since v9

#### #51 - 2019-05-06 09:53 - Ingo Fabbri

Tymoteusz Motylewski wrote:

Should be resolved with new consistent language handling available since v9

Considering this issue was created 2013 - there should be at least a FIX for 8.7. This is a serious bug.

#### #52 - 2019-05-07 12:17 - Benni Mack

- Status changed from Resolved to Closed

## Files

---

45873-annotated.png	38.4 KB	2018-02-02	Sybille Peters
---------------------	---------	------------	----------------