

EXT: Improved overlays

Extension Key: overlays

Language: en

Keywords: forDevelopers, forBeginners

Copyright 2008-2010, François Suter, <typo3@cobweb.ch>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

EXT: Improved overlays.....	1	Introduction.....	4
Introduction.....	3	Getting overlaid records in a single call.....	4
What does it do?.....	3	The full API.....	4
Questions and support.....	3	Known problems.....	7
Credits.....	3	To-Do list.....	8
Keeping the developer happy.....	3	ChangeLog.....	9
Developer's manual.....	4		

Introduction

What does it do?

This extension makes it easier to get a set of records from the TYPO3 database, properly overlaid for sites with multiple languages. Furthermore it provides in a central place a set of tools related to database calls that are sometimes spread over several classes in the TYPO3 architecture.

It also aims to reduce the number of database calls. In a scenario where many records are fetched and overlaid the number of requests to the database will be much smaller, albeit at the cost of increased PHP processing (there's no free lunch).

Questions and support

If you have any questions about this extension, please ask them in the TYPO3 English mailing list, so that others can benefit from the answers. Please use the bug tracker on forge.typo3.org to report problem or suggest features (<http://forge.typo3.org/projects/extension-overlays/issues>).

Credits

Most of the overlay code itself comes from the TYPO3 core and was not developed by myself. It is encapsulated in various methods for easier use.

Keeping the developer happy

If you like this extension, do not hesitate to rate it. Go the Extension Repository, search for this extension, click on its title to go to the details view, then click on the "Ratings" tab and vote (you need to be logged in). Every new vote keeps the developer ticking. So just do it!

You may also take a step back and reflect about the beauty of sharing. Think about how much you are benefiting and how much yourself is giving back to the community.

Developer's manual

Introduction

Traditionally getting a set of records from the database in a TYPO3 application (typically a FE plugin) is done by calling a method like `t3lib_db::exec_SELECTquery()`. Then – if your site uses multiple languages and your records are localized using standard TYPO3 mechanisms – you have to overlay each record with its translation by calling `t3lib_page::getRecordOverlay()`. It is also up to you to check whether the translation succeeded or not. Here is how your code may look like, taking “tt_news” as an example and doing a simple rendering with a bullet list:

```
// Restrict records to default language
$languageCondition = $GLOBALS['TCA']['tt_news']['ctrl']['languageField'] . ' IN (0, -1)';
// Also take records that exist only in chosen language
if ($GLOBALS['TSFE']->sys_language_content > 0) {
    $languageCondition .= ' OR (tt_news.' . $GLOBALS['TCA']['tt_news']['ctrl']['languageField'] . " =
'" . $GLOBALS['TSFE']->sys_language_content . "' AND tt_news." . $GLOBALS['TCA']['tt_news']['ctrl']
['transOrigPointerField'] . " = '0')";
}

// Add enable fields check
$where = '(' . $languageCondition . ') ' . $GLOBALS['TSFE']->sys_page->enableFields('tt_news');
$news = $GLOBALS['TYPO3_DB']->exec_SELECTgetRows('uid, pid, title', 'tt_news', $where);

$content = '<ul>';
foreach ($news as $item) {
    // Perform overlay on each record
    $overlay = $GLOBALS['TSFE']->sys_page->getRecordOverlay('tt_news', $item, $GLOBALS['TSFE']->
sys_language_content, $GLOBALS['TSFE']->sys_language_contentOL);
    // Check if record was not unset (in translation strict mode)
    if (isset($overlay)) {
        $content .= '<li>' . $overlay['title'] . '</li>';
    }
}
$content .= '</ul>';
```

Getting overlaid records in a single call

Using the “overlays” library the above code is reduced to the following:

```
$news = tx_overlays::getAllRecordsForTable('title', 'tt_news');
$content .= '<ul>';
foreach ($news as $item) {
    $content .= '<li>' . $item['title'] . '</li>';
}
$content .= '</ul>';
```

The `tx_overlays::getAllRecordsForTable()` method takes the same arguments as `t3lib_db::exec_SELECTquery()`, but takes care of all the enable fields and all the translation process.

On top of this the “overlays” library is so structured that all translation overlays are gotten in a single database request. The normal TYPO3 process generates one query per record to overlay. Thus “overlays” plays much nicer to your database. This comes at the price of some additional PHP processing.

The full API

On top of the method demonstrated above, class `tx_overlays` provides a full toolbox related to getting overlaid records in the TYPO3 frontend. This API is described below.

The extension comes with an autoloader declaration, so there's no need to include the `tx_overlays` class if you're using TYPO3 4.3 or above.

`tx_overlays` is a fully static class. As such it cannot be XCLASSed.

Method:	Parameters:	Description:
getAllRecordsForTable	\$selectFields (string) Comma-separated list of fields to select from the table. \$fromTable (string) Table from which to select. \$whereClause (string) Additional WHERE clause for the query. Enable fields are handled automatically. Optional, defaults to empty string. \$groupBy (string) GROUP BY field(s). Optional, defaults to empty string. \$orderBy (string) ORDER BY field(s). Optional, defaults to empty string. \$limit (string) LIMIT value ([begin,]max). Optional, defaults to empty string.	<p>This method returns an array with all records properly overlaid with their translations and filtered according to the table's enable fields.</p> <p>NOTE: this will not work properly using implicit joins (i.e. <code>SELECT * FROM foo,bar...</code>) as whether enable fields, nor translations will be cleanly applied to both tables.</p>
getEnabledFieldsCondition	\$table (string) Name of the table to build the condition for. \$showHidden (boolean) Set to TRUE to force the display of hidden records. \$ignoreArray (array) Defines which enable fields to ignore. Use keys like "disabled", "starttime", "endtime", "fe_group" (i.e. keys from "enablecolumns" in TCA) and set values to TRUE to exclude corresponding conditions from WHERE clause.	<p>This method returns the complete SQL condition needed to respect all enable fields as defined in the TCA of the table. Essentially this method is a wrapper around <code>t3lib_page::enableFields()</code>. The only difference is that it removes the " AND " at the beginning of the condition, so that it's not necessary to prepare for such condition by adding things like "1=1" to your SQL statements.</p> <p>For more details look at what happens inside <code>t3lib_page::enableFields()</code>.</p>
getLanguageCondition	\$table (string) Name of the table to build the condition for.	<p>This method returns the SQL condition necessary to select the correct records with records to their language, based on the table's TCA and the currently selected language in the FE.</p>
getWorkspaceCondition	\$table (string) Name of the table to build the condition for.	<p>In its current state this method ensures that only live records are returned. Workspace preview is currently not supported.</p>
selectOverlaysFields	\$table (string) Table from which to select. \$selectFields (string) List of fields to select from the table.	<p>This method makes sure that the query will include the necessary fields for the language overlay process (e.g. uid, pid and the fields related to localization). It returns the original list of selected fields plus any other necessary fields that were missing.</p>
overlayRecordSet	\$table (string) Table name \$recordset (array) Full recordset to overlay. Must contain uid, pid and <code>\$TCA[\$table]['ctrl']['languageField']</code> . \$currentLanguage (integer) Uid of the currently selected language in the FE. \$overlayMode (string) Overlay mode. If "hideNonTranslated" then records without translation will not be returned un-translated but removed instead.	<p>This method performs the translation overlays on the given recordset, according to the language and overlay settings.</p> <p>NOTE: if overlay mode is "hideNonTranslated", the overlaid recordset may contain less records than the original one, if some records were missing a translation.</p>
getOverlayRecords	\$table (string) Name of the table for which to fetch the records. \$uids (array) Array of all uid's of the original records for which to fetch the translation. \$currentLanguage (integer) uid of the system language to translate to.	<p>This method is a wrapper around <code>tx_overlays::getLocalOverlayRecords()</code> and <code>tx_overlays::getForeignOverlayRecords()</code>. It will call the appropriate method depending on the translation structure (i.e. translations in same table or in foreign table). It returns the same result as these two methods.</p>
getLocalOverlayRecords	\$table (string) Name of the table for which to fetch the records. \$uids (array) Array of all uid's of the original records for which to fetch the translation. \$currentLanguage (integer) uid of the system language to translate to.	<p>This method gets all translation overlay records for the chosen language and for all records indicated by the list of uid's. It will work for tables that contain their own translations.</p> <p>It returns all overlay records organized by uid and pid so that the overlay process can take place properly afterward.</p>
getForeignOverlayRecords	\$table (string) Name of the table for which to fetch the records. \$uids (array) Array of all uid's of the original records for which to fetch the translation. \$currentLanguage (integer) uid of the system language to translate to.	<p>This method gets all translation overlay records for the chosen language and for all records indicated by the list of uid's. It will work for tables that use a foreign table to store their translations.</p> <p>It returns all overlay records organized by uid and pid so that the overlay process can take place properly afterward.</p>

Method:	Parameters:	Description:
overlaySingleRecord	\$table (string) Name of the table for which the operation is taking place \$record (array) Record to overlay \$overlay (array) Overlay of the record	This method actually performs the translation overlay of a single record taking into account all fine-grained translation settings at field level (i.e. "l10n_mode"). It returns the properly overlaid record.

Known problems

- The extension currently doesn't support workspace preview. The records that can be fetched with `tx_overlays::getAllRecordsForTable()` are always the live versions.

To-Do list

It is planned to integrate workspace preview support in the near future.

For other feature requests, please use the bug tracker on forge.typo3.org (<http://forge.typo3.org/projects/extension-overlays/issues>).

ChangeLog

Version	Changes:
1.0.0	First public release