TYPO3 CMS 7.6 – What’s New
Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

10/November/2015
Creative Commons BY-NC-SA 3.0
Chapter Overview

Introduction

Backend User Interface

TSconfig & TypoScript

In-Depth Changes

Extbase & Fluid

Deprecated/Removed Functions

Sources and Authors
Introduction

The Facts
Introduction

TYPO3 CMS 7.6 – The Facts

- Release date: 10 November 2015
- Release type: Long Term Support (LTS) Release
- Vision: Embrace, Innovate, Deliver
Introduction

System Requirements

- **PHP**: v5.5.0 - v5.6.x
- **MySQL**: v5.5.x - v5.6.x (no strict mode)
- **Disk space**: min 200 MB
- **PHP settings**:
  - memory_limit >= 128M
  - max_execution_time >= 240s
  - max_input_vars >= 1500
  - compilation option --disable-ipv6 must **not** be used
- **Backend** requires IE >= 9 or any other modern browser

*) Further details: PHP Minimum Requirements for TYPO3 CMS 7
Introduction

Development and Release Timeline
Introduction

TYPO3 CMS Roadmap

Release dates and their primary focus:

- v7.0 02/Dec/2014 Backend Overhaul Vol 1
- v7.1 24/Feb/2015 Core Cleanup & Streamlining
- v7.2 28/Apr/2015 Frontend
- v7.3 16/Jun/2015 Package Ecosystem, Composer
- v7.4 04/Aug/2015 Backend Overhaul Vol 2
- v7.5 29/Sep/2015 Finalization
- v7 LTS 10/Nov/2015 **TYPO3 CMS 7 LTS** (Long Term Support)

https://typo3.org/typo3-cms/roadmap/
Introduction

Installation

- **Official installation procedure under Linux/Mac OS X**
  (DocumentRoot for example `/var/www/site/htdocs`):

  ```
  $ cd /var/www/site
  $ wget --content-disposition get.typo3.org/7.6
  $ tar xzf typo3_src-7.6.0.tar.gz
  $ cd htdocs
  $ ln -s ../typo3_src-7.6.0 typo3_src
  $ ln -s typo3_src/index.php
  $ ln -s typo3_src/typo3
  $ touch FIRST_INSTALL
  ```

- **Symbolic links under Microsoft Windows:**
  - Use junction under Windows XP/2000
  - Use mklink under Windows Vista and Windows 7
Introduction

Upgrade to TYPO3 CMS 7.x

- Upgrades only possible from TYPO3 CMS 6.2 LTS
- TYPO3 CMS < 6.2 should be updated to TYPO3 CMS 6.2 LTS first

- Upgrade instructions:
  - http://wiki.typo3.org/Upgrade#Upgrading_to_7.6
- Official TYPO3 guide "TYPO3 Installation and Upgrading":
  - http://docs.typo3.org/typo3cms/InstallationGuide

- General approach:
  - Check minimum system requirements (PHP, MySQL, etc.)
  - Review deprecation_*.log in old TYPO3 instance
  - Update all extensions to the latest version
  - Deploy new sources and run Install Tool -> Upgrade Wizard
  - Review startup module for backend users (optionally)
Chapter 1:

Backend User Interface
Backend User Interface

Workspaces Notification Settings (1)

Meaning and behavior of notification settings have been streamlined (an upgrade wizard helps to upgrade the settings to the new definitions)
Backend User Interface

Workspaces Notification Settings (2)

Stage publishing execute received configuration options

Stage "publishing execute":
Settings dialog

- show dialog
- changeable preselection

Preselection

- owners
- members
- editors

Default notification mail recipients

- Backend user
- Backend usergroup
Backend User Interface

Search Functionality in Element Browser

File search has been added to the TYPO3 Element Browser (works recursively)
Chapter 2: TSconfig & TypoScript
indexed_search: Parameters

The following TypoScript properties can now be configured for indexed_search:

- `titleCropAfter = 50`
- `titleCropSignifier = ...`
- `summaryCropAfter = 180`
- `summaryCropSignifier =`
- `hrefInSummaryCropAfter = 60`
- `hrefInSummaryCropSignifier = ...`
- `markupSW_summaryMax = 300`
- `markupSW_postPreLgd = 60`
- `markupSW_postPreLgd_offset = 5`
- `markupSW_divider = ...`

Keys can be:
- `plugin.tx_indexedsearch.results`
- `plugin.tx_indexedsearch.settings.results`

Every property has `stdWrap`-functionality
TSconfig & TypoScript

indexed_search: Configurable Path Separator

- New TypoScript configuration option `breadcrumbWrap` has been added.
- This allows to configure page path separator in `indexed_search` results.
- This option supports the TypoScript `option split` syntax. Default configuration is "/":

```
plugin.tx_indexedsearch.settings.breadcrumbWrap = / || / 
```
New TypoScript configuration option has been added: `forwardSearchWordsInResultLink.no_cache`

This controls whether the `no_cache` parameter should be added to page links for `indexed_search`.

- For Indexed Search Extbase plugins:
  ```
  plugin.tx_indexedsearch.settings.forwardSearchWordsInResultLink.no_cache = 1
  ```

- For plugins based on AbstractPlugin:
  ```
  plugin.tx_indexedsearch.forwardSearchWordsInResultLink.no_cache = 1
  ```
Chapter 3: In-Depth Changes
In-Depth Changes

Bootstrap for Install Tool (1)

- The Install Tool is now based on Bootstrap – for the installation part:
In-Depth Changes

Bootstrap for Install Tool (2)

The Install Tool is now based on Bootstrap – for the configuration:

Change configuration values

Changed values are written to LocalConfiguration.php. The optional file AdditionalConfiguration.php is not controlled by the TYPO3 CMS core and may override single settings again. Administrators must maintain AdditionalConfiguration.php on their own and should use it with care.

Filter by: ImageMag

Backend [BE]

[BE]\(\text{enable}_\text{exec\_function}\) = 0

- Boolean: Don’t use exec() function (except for ImageMagick which is disabled by [FX]\(\text{im\_exec}\)). If set, all file operations are done by the default PHP functions. This is necessary under Windows! On Unix the system commands by exec() can be used, unless this is disabled.

Database [DB]

Extension Installation [EXT]

Frontend [FE]

Image Processing [FX]

[FX]\(\text{im\_path\_exec}\) = /usr/bin/

Path to the IM tool ‘convert’ with LZW enabled! See ‘gifcompress‘. If your version 4.3.5 of ImageMagick is compiled with LZW you may leave this field blank AND disable the flag ‘gif\_compress’. Tip: You can call LZW ‘convert’ with a prefix like ‘myver\_convert’ by setting this path with it, e.g. ‘/usr/bin/myver\_’ instead of just ‘/usr/bin’.

[FX]\(\text{im\_version}\) = 4.3.5

Write configuration

TYPO3 CMS 7.6 - What’s New
In-Depth Changes

CSRF Protection for Frontend Plugins

- New class allows usage of the FormProtection API in the frontend
- This implements a CSRF protection (Cross-Site Request Forgery)

```
$formToken = \TYPO3\CMS\Core\FormProtection\FormProtectionFactory::get()->getFormProtection()->
genenerateToken('news', 'edit', $uid);
if (
    $dataHasBeenSubmitted
    && \TYPO3\CMS\Core\FormProtection\FormProtectionFactory::get()->validateToken(
        \TYPO3\CMS\Core\Utility\GeneralUtility::_POST('formToken'), 'User setup', 'edit')) {
    // processes the data
} else {
    // invalid token!
}
```
In-Depth Changes

Tabs for LinkBrowser (1)

- This new feature allows to extend the LinkBrowser with new tabs.
- Each tab is handled by a so called "LinkHandler", which has to implement the following Interface:
  \TYPO3\CMS\Recordlist\LinkHandler\LinkHandlerInterface
- LinkHandlers are registered in PageTSconfig as follows:

```php
file {
    handler = TYPO3\CMS\Recordlist\LinkHandler\FileLinkHandler
    label = LLL:EXT:lang/locallang_browse_links.xlf:file
    displayAfter = page
    scanAfter = page
    configuration {
        customConfig = passed to the handler
    }
}
```
In-Depth Changes

Tabs for LinkBrowser (2)

- Options `displayBefore` and `displayAfter` define the positions of the tabs
- Options `scanBefore` and `scanAfter` define the order in which handlers are executed when scanning existing links

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['LinkBrowser']['hooks'] = [
    'handler' => 
        \Vendor\Ext\MyClass::class,
    'before' => [], // optional
    'after' => [] // optional
];
```
In-Depth Changes

Module Template API (1)

- A new Module Template API aims to normalize the implementation of DocHeaders
- Example 1: adding a button

```php
$openInNewWindowButton = $this->moduleTemplate->getDocHeaderComponent()->getButtonBar()->makeLinkButton()->setHref('#')->setTitle($this->getLanguageService()->sL('LLL:EXT:lang/locallang_core.xlf:labels.openInNewWindow', TRUE))->setIcon($this->iconFactory->getIcon('actions-window-open', Icon::SIZE_SMALL))->setOnClick($aOnClick);

$this->moduleTemplate->getDocHeaderComponent()->getButtonBar()->addButton($openInNewWindowButton, ButtonBar::BUTTON_POSITION_RIGHT);
```
Example 2: adding a menu with menu items

```php
$lang = $this->moduleTemplate->getDocHeaderComponent()->getModuleMenuRegistry()->makeMenu()->setIdentifier('_langSelector')->setLabel($this->getLanguageService()->sL('LLL:EXT:lang/locallang_general.xlf:LGL.language', TRUE));

$menuItem = $languageMenu->makeMenuItem()->setTitle($lang['title'] . $newTranslation)->setHref($href);

if ((int)$lang['uid'] === $currentLanguage) {
    $menuItem->setActive(TRUE);
}

$languageMenu->addMenuItem($menuItem);
$this->moduleTemplate->getDocHeaderComponent()->getModuleMenuRegistry()->addMenu($languageMenu);
```
PSR-7 Routing for Backend AJAX Requests

To add a route for an AJAX request, file `Configuration/Backend/AjaxRoutes.php` can be created with the following content:

```php
return [
    // do something
    'unique_route_name' => [
        'path' => '/toolcollection/some-action',
        'target' => \Vendor\Controller\SomeController::class . '::myAction',
    ],
];
```
In-Depth Changes

OpenID Hook `getUserRecord`

Two hooks have been added to the OpenID service (1/2)

- **Hook 1:**
  ```
  $GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['openid']['getUserRecord']
  ```
  - Modifies the user record after it has been fetched, or:
  - Create a new record if none was found
  - Parameters `record`, `response` and `authInfo` are passed to the hook
In-Depth Changes

OpenID Hook authRequest

Two hooks have been added to the OpenID service (2/2)

- Hook 2:
  
  ```php
  $GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['openid']['authRequest']
  ```

  - Modifies the Authentication Request, before it is sent
  - Can be used to request additional attributes such as a nickname from the OpenID Server for example
  - Parameters authRequest and authInfo are passed to the hook
In-Depth Changes

Hooks and Signals (1)

- It is now possible to change the upload folder returned by `BackendUserAuthentication::getDefaultUploadFolder()`.
- Register the hook in file `ext_localconf.php` as follows:

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_userauthgroup.php']['getDefaultUploadFolder'][] = \Vendor\MyExtension\Hooks\DefaultUploadFolder::class . '->getDefaultUploadFolder';
```
In-Depth Changes

Hooks and Signals (2)

Example:

```php
<?php
namespace Vendor\MyExtension\Hooks;
use TYPO3\CMS\Core\Authentication\BackendUserAuthentication;
use TYPO3\CMS\Core\Resource\Folder;

/**
 * Class DefaultUploadFolder
 */
class DefaultUploadFolder {

    /**
     * Get default upload folder
     * If there is a folder present with the same name as the last part of the table name use that folder.
     * @param array $params
     * @param BackendUserAuthentication $backendUserAuthentication
     * @return Folder
     */
    public function getDefaultUploadFolder($params, BackendUserAuthentication $backendUserAuthentication) {
        [...]
Example (continued):

```php
/** @var Folder $uploadFolder */
$uploadFolder = $params['uploadFolder'];
$pid = $params['pid'];
$table = $params['table'];
$field = $params['field'];

(matches = [];
if (!empty($uploadFolder) && preg_match('/(_([a-z]+)$/', $table, $matches)) {
    $folderName = $matches[1];
    if ($uploadFolder->hasFolder($folderName)) {
        $uploadFolder = $uploadFolder->getSubfolder($folderName);
    }
} return $uploadFolder;
```

TYPO3 CMS 7.6 - What’s New
In-Depth Changes

Miscellaneous

- Using the TCA field type `select` requires to specify an option `renderType`

- Valid values are:

  ```
  'renderType' => 'selectMultipleSideBySide',
  'renderType' => 'selectCheckBox',
  'renderType' => 'selectSingle',
  'renderType' => 'selectSingleBox',
  'renderType' => 'selectTree',
  ```
Chapter 4:
Extbase & Fluid
Extbase & Fluid

Same Table Relations

It is now possible to use a domain model where an object is connected to another object of the same class directly

```php
namespace \Vendor\Extension\Domain\Model;

class A {
    /**
     * @var \Vendor\Extension\Domain\Model\A
     */
    protected $parent;
}
```

```php
namespace \Vendor\Extension\Domain\Model;

class A {
    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $x;

    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $y;
}
```
Extbase & Fluid

Option absolute for Image-ViewHelpers

- New option absolute forces ImageViewhelper and Uri/ImageViewHelper to output an **absolute** URL
- **Example 1 (ImageViewhelper):**

  ```html
  <f:image image="{file}" width="400" height="375" absolute="1" />
  // Output
  <img alt="alt set in image record"
       src="http://example.com/fileadmin/_processed_/323223424.png"
       width="400" height="375" />
  ```

- **Example 2 (Uri/ImageViewHelper):**

  ```html
  <f:uri.image image="{file}" width="400" height="375" absolute="1" />
  // Output
  http://example.com/fileadmin/_processed_/323223424.png
  ```
Extbase & Fluid

Strip Whitespace between HTML Tags

- New ViewHelper `spaceless` removes redundant spaces between HTML tags while preserving the whitespace that may be inside HTML tags:

```php
<f:spaceless>
<div>
  <div>
    <div>
      text
    </div>
  </div>
</div>
```

- **Output:**

```html
<div><div><div>text

text</div></div></div>
```
Extbase & Fluid

RootLevel Configuration

- The RootLevel of a table can be configured in TCA now (this define where records of a table can be found in the system)
  - 0: page tree only
  - 1: on the root page only (PID 0)
  - -1: both, root page and page tree

- TCA configuration:

  ```php
  $GLOBALS['TCA']['tx_myext_domain_model_record']['ctrl']['rootLevel'] = -1;
  ```
Chapter 5:

Deprecated/Removed Functions
Deprecated/Removed Functions

**AJAX Routes**

- AJAX handlers registered in the core by `ExtensionManagementUtility::registerAjaxHandler()` have been replaced by AJAX routes, which are registered under `Configuration/Backend/AjaxRoutes.php`

- Complete list (old/new) is available at: [Breaking-69916-RegisteredAJAXHandlersReplacedByRoutes.rst](Breaking-69916-RegisteredAJAXHandlersReplacedByRoutes.rst)
Deprecated/Removed Functions

Miscellaneous

- Extension mediace has been moved to the TYPO3 Extension Repository
- TCA configuration `dynamicConfigFile` within the `ctrl` section of a table has been **deprecated**. The table configuration is located as `Configuration/TCA/<table_name>.php` now.
- The `jumpurl` function has been removed from the core and is available as an extension in TYPO3 Extension Repository
Chapter 6: Sources and Authors
Sources and Authors

Sources

**TYPO3 News:**
- [http://typo3.org/news](http://typo3.org/news)

**Release Infos:**
- [http://wiki.typo3.org/TYP03_CMS_7.6.0](http://wiki.typo3.org/TYP03_CMS_7.6.0)
- **INSTALL.md** and **Changelog**
- [typo3/sysext/core/Documentation/Changelog/7.6/*](http://typo3.org/sysext/core/Documentation/Changelog/7.6/*)

**TYPO3 Bug-/Issuetracker:**

**TYPO3 Git Repositories:**
- [https://git.typo3.org/Packages/TYP03.CMS.git](https://git.typo3.org/Packages/TYP03.CMS.git)
- [https://git.typo3.org/Packages/TYP03.Fluid.git](https://git.typo3.org/Packages/TYP03.Fluid.git)
Sources and Authors

**TYPO3 CMS What’s New Slides:**

Patrick Lobacher  
(Research, Information Gathering and German Version)

Michael Schams  
(Project Leader and English Version)

**Translations by:**
Andrey Aksenov, Pierrick Caillon, Sergio Catala, Jigal van Hemert, Michel Mix, Sinisa Mitrovic, Angeliki Plati, Nena Jelena Radovic, Roberto Torresani

http://typo3.org/download/release-notes/whats-new

Licensed under Creative Commons BY-NC-SA 3.0