TYPO3 CMS 7.5 – What’s New

Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

11/October/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 7.5 - The Facts

- Release date: 29 September 2015
- Releasetype: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Primary focus: Finalization
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
  - 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
TYPOP3 CMS Roadmap

Estimated 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 Oct/Nov/2015  **TYPO3 CMS 7 LTS** (Long Term Release)

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/7.5
  $ tar xzf typo3_src-7.5.0.tar.gz
  $ cd htdocs
  $ ln -s ../typo3_src-7.5.0 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
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_7.5](http://wiki.typo3.org/Upgrade#Upgrading_to_7.5)

- Official TYPO3 guide "TYPO3 Installation and Upgrading":

- 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
WEB->View features a dropdown to select a language for the page preview. (can be deactivated by mod.SHARED.view.disableLanguageSelector = 1)
Content Element textmedia

A new Content Element "Text & Media" combines the elements text, image and textpic.

New content element

Please select the type of page content you wish to create:

<table>
<thead>
<tr>
<th>Typical page content</th>
<th>Special elements</th>
<th>Form elements</th>
<th>Plugins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header Only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adds a header only.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Images Only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any number of images aligned in columns and rows with a caption.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bullet List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A single bullet list.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A simple table.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Text &amp; Media</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any number of media wrapped right around a regular text element.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Links</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Backend User Interface

YouTube and Vimeo Files

Content Element "Text & Media" allows editors to include external YouTube and Vimeo files, as well as local files.
Backend User Interface

Search in Module Filelist

Module "Filelist" features a search by file name now (recursively from the current folder).
Chapter 2:

TSconfig & TypoScript
INCLUDE_TYPOSCRIPT has an extra (optional) property "condition" now, which includes the file/directory only, if the condition is met.

// only include TypoScript, if current user is logged in:
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/feuser.ts"
condition="[loginUser = *]">

// include TypoScript depending on application context:
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/staging.ts"
condition="applicationContext = /^Production//Staging//Server\d+$/">
TCA-Option: Display Date Offset

- TCA option `disableAgeDisplay` disables the display of the age (for example: "2015-08-30 (-27 days)"

  ```
  $GLOBALS['TCA']['tt_content']['columns']['date']['config']['disableAgeDisplay'] = true;
  ```

- As a precondition, type of the field has to be input and `eval` has to be set to `date`
Inline Language Label Files with TypoScript (1)

- XLF language files can be read and converted into an inline array
- This enables accessing language labels in JavaScript for example
- The following three optional parameters are supported:
  - `selectionPrefix`: only label keys that start with this prefix will be included
  - `stripFromSelectionName`: string that will be removed from any included label key
  - `errorMode`: error mode if file could not be found:
    - 0: syslog entry (default), 1: ignore, 3: throw an exception
Example:

```plaintext
type = PAGE
page.inlineLanguageLabelFiles {
    someLabels = EXT:myExt/Resources/Private/Language/locallang.xlf
    someLabels.selectionPrefix = idPrefix
    someLabels.stripFromSelectionName = strip_me
    someLabels.errorMode = 2
}
```

Output:

```html
<script type="text/javascript">
/*<![CDATA[*/
var TYPO3 = TYPO3 || {};
TYPO3.lang = {
    "firstLabel": [{"source":"first Label","target":"erstes Label"}],
    "secondLabel": [{"source":"second Label","target":"zweites Label"}]
/*]]></script>
```
TSconfig & TypoScript

Workspace Preview by TSconfig

- TYPO3 CMS generates preview links only for tables `tt_content`, `pages` and `pages_language_overlay` by default.
- This can be configured using PageTSconfig now:

  # use page 123 for previewing workspaces records (in general)
  options.workspaces.previewPageId = 123

  # use the pid field of each record for previewing (in general)
  options.workspaces.previewPageId = field:pid

  # use page 123 for previewing workspaces records (for table `tx_myext_table`)
  options.workspaces.previewPageId.tx_myext_table = 123

  # use the pid field of each record for previewing (or table `tx_myext_table`)
  options.workspaces.previewPageId.tx_myext_table = field:pid
Image Quality of sourceCollection

- Image quality of each sourceCollection entry can be configured now.
- This setting takes precedence over configuration in Install Tool (as stored in file LocalConfiguration.php).
- Example:

  `# for small retina images
tt_content.image.20.1.sourceCollection.smallRetina.quality = 80`

  `# for large retina images
tt_content.image.20.1.sourceCollection.largeRetina.quality = 65`
Count Elements in a List

- A new property `returnCount` has been added to the `stdWrap` property `split`.
- This allows to count the number of elements in a comma-separated list.
- The following code returns 9 for example:

```plaintext
define 1 = TEXT
1 {
    value = x,y,z,1,2,3,a,b,c
    split.token = ,
    split.returnCount = 1
}
```
Handling of Backend Layouts (1)

- Handling of backend layouts has been simplified for the frontend
- New option `pagelayout` can be used in TypoScript

Example:

```plaintext
page.10 = FLUIDTEMPLATE
page.10 {
  file.stdWrap.cObject = CASE
  file.stdWrap.cObject {
    key.data = pagelayout
    default = TEXT
    default.value = EXT:sitepackage/Resources/Private/Templates/Home.html
    3 = TEXT
    3.value = EXT:sitepackage/Resources/Private/Templates/1-col.html
    4 = TEXT
    4.value = EXT:sitepackage/Resources/Private/Templates/2-col.html
  }
}
```

(continue on next page)
Handling of Backend Layouts (2)

...where key.data = pagelayout replaces the following code:

field = backend_layout
ifEmpty.data = levelfield:-2,backend_layout_next_level,slide
ifEmpty.ifEmpty = default
TSconfig & TypoScript

Miscellaneous

- stdWrap-Funktion bytes has been introduced in TYPO3 CMS 7.4
- The ability to set the base has been added in TYPO3 CMS 7.5, which allows to define whether to use a base of 1000 or 1024 to calculate with

  ```
  bytes.labels = " | K| M| G"
  bytes.base = 1000
  ```
Chapter 3:

In-Depth Changes
In-Depth Changes

Fluid-based Content Elements (1)

- New system extension "Fluid-based Content Elements" has been implemented
- Fluid templates are used for the rendering of content elements rather than TypoScript
- Could be an alternative to CSS Styled Content at one point in the future
- Include the following static templates in order to use this feature:
  - Content Elements (fluid_styled_content)
  - Content Elements CSS (optional) (fluid_styled_content)
In-Depth Changes

Fluid-based Content Elements (2)

- In addition, the following PageTSconfig template has to be added to the page properties:
  Fluid-based Content Elements (fluid_styled_content)

- Overwrite default templates by adding own paths in TypoScript setup:

  ````
  lib.fluidContent.templateRootPaths.50 = EXT:site_example/Resources/Private/Templates/
  lib.fluidContent.partialRootPaths.50 = EXT:site_example/Resources/Private/Partials/
  lib.fluidContent.layoutRootPaths.50 = EXT:site_example/Resources/Private/Layouts/
  ```
In-Depth Changes

Fluid-based Content Elements (3)

- Migrate from CSS Styled Content to Fluid-based Content Elements:
  - Uninstall extension css_styled_content
  - Install extension fluid_styled_content
  - Use the Upgrade Wizard in the Install Tool to migrate Content Elements text, image and textpic to textmedia

Note: "Fluid-based Content Elements" is still in an early stage and breaking changes are possible until TYPO3 CMS 7 LTS. Also some conflicts regarding CSS Styled Content possibly still exist.
In-Depth Changes

SELECTmmQuery Method

- New method SELECT_mm_query has been added to class DatabaseConnection
- Extracted from exec_SELECT_mm_query to separate the building and execution of M:M queries.
- This enables the use of the query building in the database abstraction layer

```
$query = SELECT_mm_query('*, ', 'table1', 'table1_table2_mm', 'table2', 'AND table1.uid = 1', '', 'table1.title DESC');
```
Optimize Database Tables in MySQL

- New scheduler task to run the MySQL command `OPTIMIZE TABLE` on selected database tables
- This command reorganizes the physical storage of table data and associated index data to reduce storage space and improve I/O efficiency
- The following types of tables are supported: MyISAM, InnoDB and ARCHIVE
- Using this task with DBAL and other DBMS is not supported due to the fact that the commands used are MySQL-specific

Note: optimizing tables is an I/O intensive process. Also in MySQL < 5.6.17 the process locks the tables while it is running, which may impact the website.
In-Depth Changes

Handling of Online Media (1)

- External medias (online media) are supported by default now
- As examples, the support for YouTube and Vimeo videos has been implemented in the core
- Resources can be added as URLs using content element "Text & Media" for example
- Matching helper class fetches the meta data and supplies an image that will be used as the preview if available
In-Depth Changes

Handling of Online Media (2)

The following URL syntaxes are possible:

**YouTube:**
- youtu.be/<code>
- www.youtube.com/watch?v=<code>
- www.youtube.com/v/<code>
- www.youtube-nocookie.com/v/<code>
- www.youtube.com/embed/<code>

**Vimeo:**
- vimeo.com/<code>
- player.vimeo.com/video/<code>
Accessing the resources using Fluid can be achieved as follows:

```html
<!-- enable js api and set no-cookie support for YouTube videos -->
<f:media file="{file}" additionalConfig="{enablejsapi:1, 'no-cookie': true}" ></f:media>

<!-- show title and uploader for YouTube and Vimeo before video starts playing -->
<f:media file="{file}" additionalConfig="{showinfo:1}" ></f:media>
```

- **Custom configuration options for YouTube videos:**
  - autoplay, controls, loop, enablejsapi, showinfo, no-cookie

- **Custom configuration options for Vimeo videos:**
  - autoplay, loop, showinfo
Handling of Online Media (4)

To register your own online media service, you need an `OnlineMediaHelper` class that implements `OnlineMediaHelperInterface` and a `FileRenderer` class that implements `FileRendererInterface`

// register your own online video service (the used key is also the bind file extension name)
$GLOBALS['TYPO3_CONF_VARS']['SYS']['OnlineMediaHelpers']['myvideo'] = 
   \\MyCompany\Myextension\Helpers\MyVideoHelper::class;

$rendererRegistry = \TYPO3\CMS\Core\Resource\Rendering\RendererRegistry::getInstance();
$rendererRegistry->registerRendererClass(
   \\MyCompany\Myextension\Rendering\MyVideoRenderer::class
);

// register an custom mime-type for your videos
$GLOBALS['TYPO3_CONF_VARS']['SYS']['FileInfo']['fileExtensionToMimeType']['myvideo'] = 'video/myvideo';

// register your custom file extension as allowed media file
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] .= ',myvideo';
In-Depth Changes

Backend Routing

■ New routing component has been added to the TYPO3 backend which handles addressing different calls/modules inside TYPO3 CMS

■ Routes can be defined in the following class:
  Configuration/Backend/Routes.php

```php
return [
  'myRouteIdentifier' => [
    'path' => '/document/edit',
    'controller' => Acme\MyExtension\Controller\MyExampleController::class . '::methodToCall'
  ]
];
```

■ Called method contains PSR-7 compliant request and response objects:

```php
public function methodToCall(ServerRequestInterface $request, ResponseInterface $response) {
...
}
```
In-Depth Changes

Autoload Definition in ext_emconf.php

- Extensions may provide one or more PSR-4 definitions in file ext_emconf.php now.
- This was already possible in composer.json, but with this new feature, extension developers do not need to provide a composer file just for this anymore.

```php
$EM_CONF[$_EXTKEY] = array(  
    'title' => 'Extension Skeleton for TYPO3 CMS 7',
    ...
    'autoload' =>
        array(  
            'psr-4' => array(  
                'Helhum\ExtScaffold\' => 'Classes'
            )
        );
```

(this is the new recommended way to register classes in TYPO3)
In-Depth Changes

New Icon Factory (1)

- Logic for working with icons, icon sizes and icon overlays is now bundled into the new class IconFactory.
- The new icon factory will replace the old icon skinning API step by step.
- All core icons will be registered directly in the IconRegistry class.
- Extensions must use IconRegistry::registerIcon() to override existing icons or add additional icons to the icon factory:

```php
IconRegistry::registerIcon(
    $identifier,
    $iconProviderClassName,
    array $options = array()
);
```
The TYPO3 CMS core implements three icon provider classes: BitmapIconProvider, FontawesomeIconProvider and SvgIconProvider.

Example usage:
```php
$iconFactory = GeneralUtility::makeInstance(IconFactory::class);
$iconFactory->getIcon($identifier, Icon::SIZE_SMALL, $overlay, IconState::cast(IconState::STATE_DEFAULT))->render();
```

Valid values for `Icon::SIZE_...` are:
- SIZE_SMALL, SIZE_DEFAULT and SIZE_LARGE

Valid values for `Icon::STATE_...` are:
- STATE_DEFAULT and STATE_DISABLED
In-Depth Changes

New Icon Factory (3)

- The TYPO3 CMS core provides a Fluid ViewHelper which makes it easy to use icons within a Fluid view:

```php
{namespace core = TYPO3\CMS\Core\ViewHelpers}

<core:icon identifier="my-icon-identifier"></core:icon>

<!-- use the "small" size if none given -->
<core:icon identifier="my-icon-identifier"></core:icon>
<core:icon identifier="my-icon-identifier" size="large"></core:icon>
<core:icon identifier="my-icon-identifier" overlay="overlay-identifier"></core:icon>

<core:icon identifier="my-icon-identifier" size="default" overlay="overlay-identifier">
</core:icon>

<core:icon identifier="my-icon-identifier" size="large" overlay="overlay-identifier">
</core:icon>
```
In-Depth Changes

Hooks and Signals

- New signal has been added to LinkValidator, which allows for additional processing upon initialization of a specific record (e.g. getting content data from plugin configuration in record)

- Registering the signal in file `ext_localconf.php`:

```php
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class
);

$signalSlotDispatcher->connect(
    \TYPO3\CMS\Linkvalidator\LinkAnalyzer::class,
    'beforeAnalyzeRecord',
    \Vendor\Package\Slots\RecordAnalyzerSlot::class,
    'beforeAnalyzeRecord'
);
```
In-Depth Changes

JumpUrl as System Extension (1)

- Generation and handling of JumpURLs has been moved to a new system extension `jumpurl`
- New hooks were introduced that allow custom URL generation and handling (see next page)

BREAKING CHANGE!
In-Depth Changes

JumpUrl as System Extension (2)

**Hook 1: manipulating URLs during link generation**

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlHandlers']['myext_myidentifier']['handler'] = \Company\MyExt\MyUrlHandler::class;

// class needs to implement the UrlHandlerInterface:
class MyUrlHandler implements \TYPO3\CMS\Frontend\Http\UrlHandlerInterface {
    ...
}
```

**Hook 2: handling of links**

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlProcessors']['myext_myidentifier']['processor'] = \Company\MyExt\MyUrlProcessor::class;

// class needs to implement the UrlProcessorInterface:
class MyUrlProcessor implements \TYPO3\CMS\Frontend\Http\UrlProcessorInterface {
    ...
}
```
In-Depth Changes

Command Line Interface (CLI)

- Calling `typo3/cli_dispatch.phpsh` via the command line now shows a colored error message if an invalid or no CLI key as first parameter was given.
- Extbase command controllers can now reside in arbitrary subfolders within the `Command` folder.
- Example:

  Controller in file:
  my_ext/Classes/Command/Hello/WorldCommandController.php
  ...can be called via CLI:
  typo3/cli_dispatch.sh extbase my_ext:hello:world <arguments>
In-Depth Changes

Miscellaneous (1)

- The move buttons of the TCA type group can now be explicitly disabled by using option `hideMoveIcons = TRUE`

- Method `makeCategorizable` has been extended with a new parameter `override` to set a new category configuration for an already registered table/field combination

- Example:

  ```php
  \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable(    'css_styled_content', 'tt_content', 'categories', array(), TRUE );
  ```

  Last parameter (here: TRUE) forces override (default value is FALSE).
In-Depth Changes

Miscellaneous (2)

- New function generates a unique ID
  
  $uniqueId = \TYPO3\CMS\Core\Utility\StringUtility::getUniqueId('Prefix');

- The file type .typoscript has been added to the list of valid plain text file types

- New configuration option defines file extensions of media files
  
  $GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] = 'gif,jpg,jpeg,bmp,png,pdf,svg,ai,mov,avi';

BREAKING CHANGE!
Chapter 4:

Extbase & Fluid
Severity Filtering for FlashMessageQueue

- In TYPO3 CMS < 7.5, all messages of the FlashMessageQueue can be fetched and/or removed only.
- In TYPO3 CMS >= 7.5, this can be done for a specific severity:

  FlashMessageQueue::getAllMessages($severity);
  FlashMessageQueue::getAllMessagesAndFlush($severity);
  FlashMessageQueue::removeAllFlashMessagesFromSession($severity);
  FlashMessageQueue::clear($severity);
Query Support for "between" added

- Support for `between` has been added to the Extbase Query object
- There is no performance benefit due to the fact that the DBMS converts "between" internally anyway: `min <= expr AND expr <= max`
- The new Extbase feature replicates the DBMS’ behavior by building a logical AND condition, so this works across all DBMS

```php
$query->matching(
    $query->between('uid', 3, 5)
);
```
Multiple FlashMessage Queues

Is it now possible to implement multiple FlashMessageQueues:

```php
$queueIdentifier = 'myQueue';
$this->controllerContext->getFlashMessageQueue($queueIdentifier);
```

Access using Fluid works as follows:

```html
<f:flashMessages queueIdentifier="myQueue" ></f:flashMessages>
```
In order to easily render video, audio and all other file types with a registered Renderer class in the frontend, the MediaViewHelper has been implemented.

MediaViewHelper first checks if there is a Renderer present for the given file - if not, it falls back and renders an image tag.

Examples:

```html
<code title="Image Object">
   <f:media file="{file}" width="400" height="375" ></f:media>
</code>

<output>
   <img alt="alt set in image record" src="fileadmin/_processed_/323223424.png" width="396" height="375" />
</output>
```
Examples (continued):

```xml
<code title="MP4 Video Object">
    <f:media file="{file}" width="400" height="375" ></f:media>
</code>

<output>
    <video width="400" height="375" controls>
        <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
    </video>
</output>

<code title="MP4 Video Object with loop and autoplay option set">
    <f:media file="{file}" width="400" height="375"
        additionalConfig="{loop: '1', autoplay: '1'}">
    </f:media>
</code>

<output>
    <video width="400" height="375" controls loop>
        <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
    </video>
</output>
```
System Extension form (1)

- System extension form (including the custom data model, controller logic, property validation, views and templating) has been adopted to support the Extbase/Fluid MVC stack.
- This allows better customization and control of the generated behavior and markup by simply modifying Fluid templates or utilizing own custom view helper logic.
- Each form element uses its own Partial, which can also be configured by the TypoScript option partialPath = ...
The following three new ViewHelpers exist:

- AggregateSelectOptionsViewHelper (for optgroup tags)
- SelectViewHelper (for optgroup tags)
- PlainMailViewHelper (to render plain text mails)

In addition, there are three Views:

- show (the form itself)
- confirmation (the confirmation page)
- postProcessor/mail (the email)

Template paths and visibility of fields can be customized for each View individually.
Annotation @cli

- By using the new annotation @cli, commands in an Extbase CommandController can be marked as CLI-commands only.
- These commands are excluded from the scheduler command selection.
- Typical use cases are commands such as `extbase:help:help` for example.
Chapter 5:
Deprecated/Removed Functions
Deprecated/Removed Functions

Slash-related Methods in GeneralUtility

- The following methods of class GeneralUtility have been marked as deprecated:
  - GeneralUtility::addSlashesOnArray()
  - GeneralUtility::stripSlashesOnArray()
  - GeneralUtility::slashArray()
Deprecated/Removed Functions

CLI-related Constants and Methods

- Logic regarding regular CLI-based scripts with the CLIkey option has been moved to the CliRequestHandler and therefore, the following method marked as deprecated:

```php
BackendUserAuthentication->checkCLIuser()
```

- Additionally, the following constants and global parameters have been marked deprecated:

```php
const TYPO3CliKey
const TYPO3CliInclude
$GLOBALS['MCONF']['name']
$GLOBALS['temp_cliScriptPath']
$GLOBALS['temp_cliKey']
```
Deprecated/Removed Functions

IconUtility

- Interface IconUtilityOverrideResourceIconHookInterface as well as class IconUtility will be removed in TYPO3 CMS 8
- All public methods of this class have been marked as deprecated:
  
  ```
  IconUtility::skinImg()
  IconUtility::getSpriteIcon()
  IconUtility::getSpriteIconForRecord()
  IconUtility::getSpriteIconClasses()
  ```

  (some methods have been moved to IconFactory)

- In addition, the following PageTSConfig setting has been marked as deprecated:

  ```
  mod.wizards.newContentElement.wizardItems.*.elements.*.icon
  ```
Deprecated/Removed Functions

**Deprecated HtmlParser Methods**

- Marker substitution functionality has been moved from `core/Classes/Html/HtmlParser.php` to its own class `core/Classes/Service/MarkerBasedTemplateService.php`

- The following methods within `HtmlParser` will be removed in TYPO3 CMS 8 and have been marked as **deprecated** now:
  - `HtmlParser::getSubpart()`
  - `HtmlParser::substituteSubpart()`
  - `HtmlParser::substituteSubpartArray()`
  - `HtmlParser::substituteMarker()`
  - `HtmlParser::substituteMarkerArray()`
  - `HtmlParser::substituteMarkerAndSubpartArrayRecursive()`
Deprecated/Removed Functions

System extension form

- System extension form is based on Extbase/Fluid now
- Therefore all TypoScript-based layout settings have been marked as deprecated and using layout is not recommended anymore:

```
10 = FORM
10 {
    layout {
        containerWrap = <div><elements ></elements></div>
        elementWrap = <div><element ></element></div>
    }
}
```
Deprecated/Removed Functions

Deprecated ViewHelpers and Methods

- The following ViewHelpers have been marked as deprecated:
  \TYPO3\CMS\Fluid\ViewHelpers\Be\Buttons\IconViewHelper
  \TYPO3\CMS\Backend\ViewHelpers\SpriteManagerIconViewHelper

- The following methods have been marked as deprecated:
  BackendUtility::getExcludeFields()
  BackendUtility::getExplicitAuthFieldValues()
  BackendUtility::getSystemLanguages()
  BackendUtility::getRegisteredFlexForms()
  BackendUtility::exec_foreign_table_where_query()
  BackendUtility::replaceMarkersInWhereClause()
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/TYPO3_CMS_7.5.0](http://wiki.typo3.org/TYPO3_CMS_7.5.0)
- INSTALL.md and ChangeLog
- [typo3/sysext/core/Documentation/Changelog/7.5/*](http://typo3.org/sysext/core/Documentation/Changelog/7.5/*)

**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