TYPO3 CMS 7.2 - What’s New

Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

16/May/2015

TYPO3

TYPO3 CMS 7.2 - What's New

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 7.2 – The Facts

- Release date: 28 April 2015
- Release type: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Primary focus: Frontend

Introduction

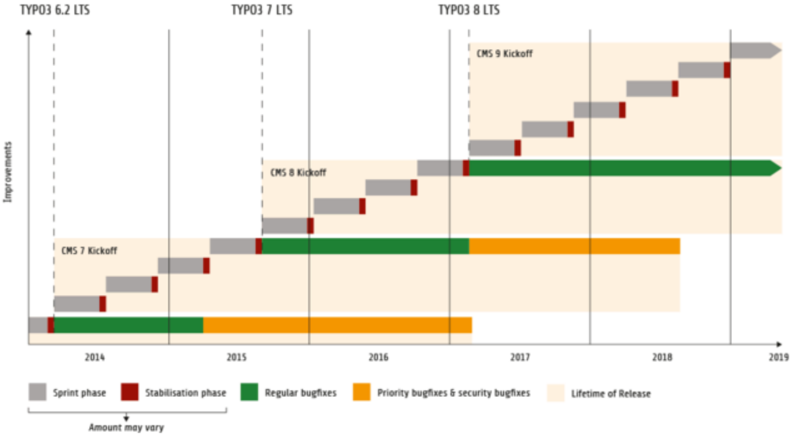
System Requirements

- 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



Introduction

TYPO3 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	09/Jun/2015	Package Ecosystem, Composer and Extension Handling
v7.4	04/Aug/2015	Backend Overhaul Vol 2
v7.5	29/Sep/2015	*(to be determined...)*
v7.6	xx/xxx/2015	**TYPO3 CMS 7 LTS** (Long Term Release)

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

http://typo3.org/news/article/embrace-and-innovate-typo3-cms-7/

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/7.2
$ tar xzf typo3_src-7.2.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.2.0 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 mlink 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_7.2
- 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)

TYPO3

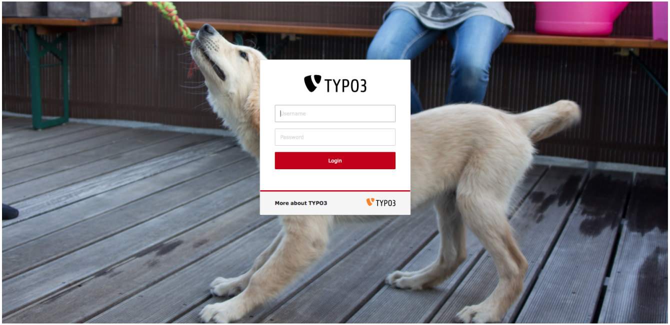
Chapter 1:

Backend User Interface

Backend User Interface

Customisable Login Form

System extension `backend` allows administrators to configure a custom background image, a logo and a colour for the backend login screen:



Backend User Interface

Image Manipulation: Cropping

An image manipulation functionality allows editors to crop images in the backend. This feature must be explicitly activated for BE users ("Exclude Fields"):

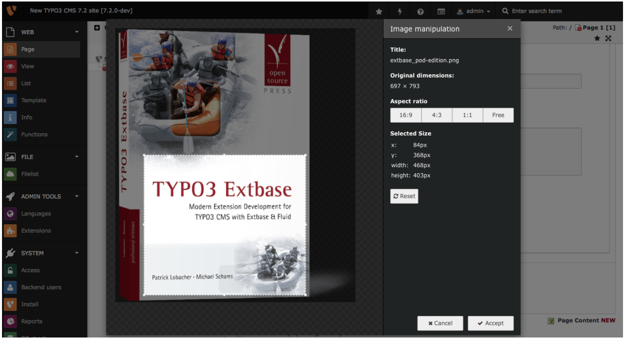


Image manipulation

Title:
extbase_pod-edition.png

Original dimensions:
697 × 793

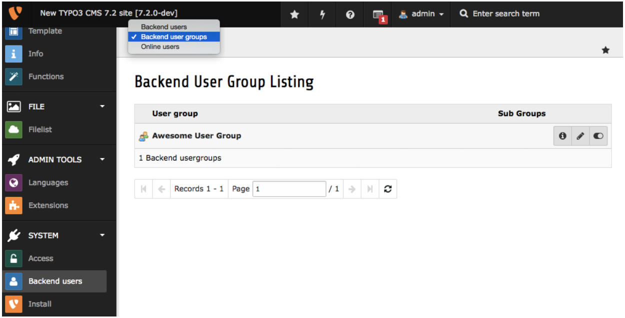
Aspect ratio

Selected Size
x: 84px
y: 368px
width: 468px
height: 403px

Backend User Interface

Backend User Groups

Backend user groups can be maintained in a submodule of the "Backend Users" module now:



Backend User Group Listing

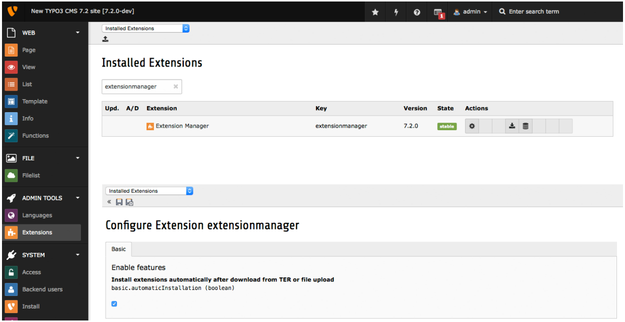
User group	Sub Groups
Awesome User Group	
1 Backend usergroups	

Records 1 - 1 Page 1 / 1

Backend User Interface

Disable Automatic Extension Installation

Administrators can configure the Extension Manager not to install downloaded extensions straight away:



Installed Extensions

extensionmanager

Upd.	A/D	Extension	Key	Version	State	Actions
		Extension Manager	extensionmanager	7.2.0	stable	

Configure Extension extensionmanager

Basic

Enable features

Install extensions automatically after download from TER or file upload

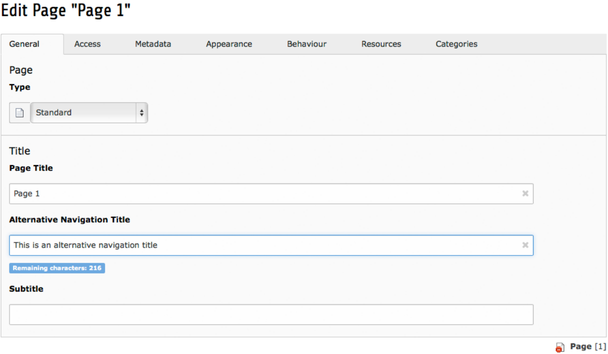
`basic.automaticInstallation (boolean)`

☑

Backend User Interface

Remaining Characters on Text Fields

The number of remaining characters is displayed below text input fields:



Edit Page "Page 1"

Page

Type

Standard

Title

Page Title

Page 1

Alternative Navigation Title

This is an alternative navigation title

Remaining characters: 216

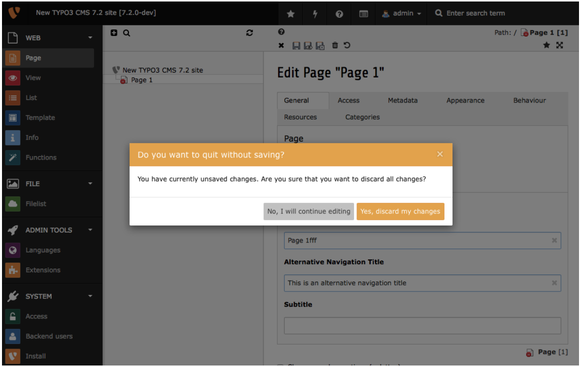
Subtitle

Page [1]

Backend User Interface

Confirm Unsaved Changes

A new confirmation dialog warns editors about loosing unsaved changes:



Do you want to quit without saving?

You have currently unsaved changes. Are you sure that you want to discard all changes?

Edit Page "Page 1"

Page 1fff

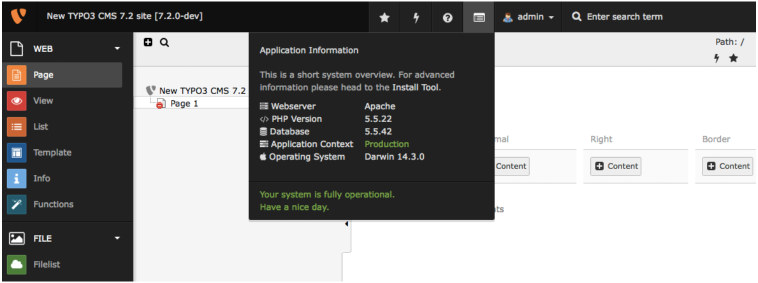
Alternative Navigation Title: This is an alternative navigation title

Subtitle:

Backend User Interface

System Information Dropdown

A dropdown menu shows several information about the system TYPO3 is installed on. The data of this dialog can be extended:
(see chapter "In-Depth Changes" for further details)



Application Information

This is a short system overview. For advanced information please head to the **Install Tool**.

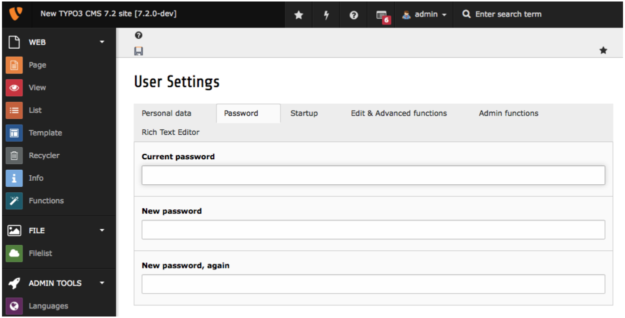
Webserver	Apache
PHP Version	5.5.22
Database	5.5.42
Application Context	Production
Operating System	Darwin 14.3.0

Your system is fully operational.
Have a nice day.

Backend User Interface

Changing Password

Backend users are forced to provide the current (old) password in order to change it to a new password:



User Settings

Current password

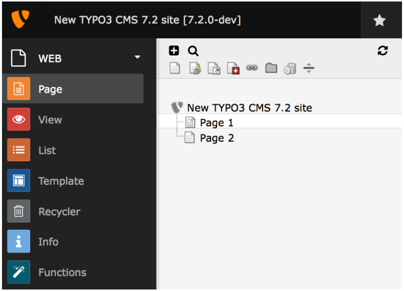
New password

New password, again

Backend User Interface

Page Icon for "Show Content from Page"

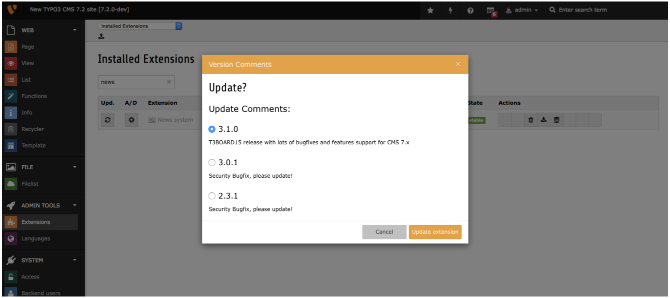
A new page icon in the page tree indicates that a page shows content from another page:



Backend User Interface

Extension Updates

When updating an extension, it is now possible to choose the target version:



Version Comments ×

Update?

Update Comments:

● 3.1.0

T3BOARD15 release with lots of bugfixes and features support for CMS 7.x

○ 3.0.1

Security Bugfix, please update!

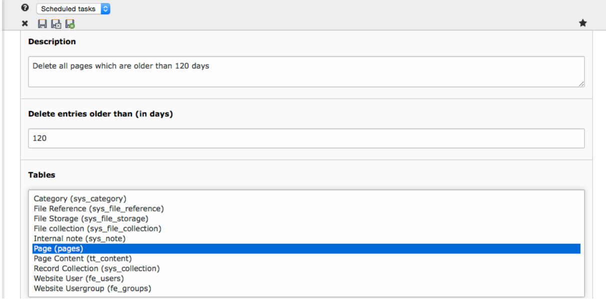
○ 2.3.1

Security Bugfix, please update!

Backend User Interface

Recycler Task

A new scheduler task for system extension `recycler` removes deleted records from content tables in the database. The maximum age and the affected tables are configurable in the task's settings.
This may also apply to files, if they are referenced in the content element.



Description

Delete all pages which are older than 120 days

Delete entries older than (in days)

120

Tables

- Category (sys_category)
- File Reference (sys_file_reference)
- File Storage (sys_file_storage)
- File collection (sys_file_collection)
- Internal note (sys_note)
- Page (pages)
- Page Content (tt_content)
- Record Collection (sys_collection)
- Website User (fe_users)
- Website Usergroup (fe_groups)

Chapter 2:

TSconfig & TypoScript

TSconfig & TypoScript

Flexible Preview URL Configuration (1)

- It is now possible to configure the preview link generated for the "save & view" button in the backend.
- A common use case is to have previews for blog or news records, but you can also define different preview pages for normal content elements.

```
TCEMAIN.preview {
  <table name> {
    previewPageId = 123
    useDefaultLanguageRecord = 0
    fieldToParameterMap {
      uid = tx_myext_pi1[showUid]
    }
    additionalGetParameters {
      tx_myext_pi1[special] = HELLO
    }
  }
}
```

Flexible Preview URL Configuration (2)

- `previewPageId`:
  UID of the page to use for preview
  (if this setting is omitted the current page will be used)
- `useDefaultLanguageRecord`:
  defines if translated records will use the UID of the default record
  (this is activated by default, value: 1)
- `fieldToParameterMap`:
  a mapping which allows to select fields of the record to be included as GET-parameters
- `additionalGetParameters`:
  allows to add arbitrary GET-parameters and even to override others

TSconfig & TypoScript

RTE Configuration: Default Target

- RTE configuration property can be used in PageTSconfig to configure a default target for links of a given type

  `buttons.link.[` *type* `].properties.target.default = ...`

- Possible link types are:
  (further types may be provided by extensions)
  - `page`
  - `file`
  - `url`
  - `mail`
  - `spec`

TSconfig & TypoScript

Strip Empty HTML Tags in HTMLparser

- A new functionality has been implemented in the HTMLparser that allows the stripping of empty HTML tags

```
stdWrap {
   // this removes all empty HTML tags
   HTMLparser.stripEmptyTags = 1
   // this removes empty h2 and h3 tags only
   HTMLparser.stripEmptyTags.tags = h2, h3
}

RTE.default.proc.entryHTMLparser_db {
   stripEmptyTags = 1
   stripEmptyTags.tags = p
   stripEmptyTags.treatNonBreakingSpaceAsEmpty = 1
}
```

Note: HTMLparser strips all unknown tags by default.
Therefore it might be useful to retain these:
`HTMLparser.keepNonMatchedTags = 1`

TYPO3

TSconfig & TypoScript

Miscellaneous

- New property `buttons.abbreviation.removeFieldsets` may be used in PageTSconfig to configure the abbreviation dialog

```
# Possible values are:
# acronym, definedAcronym, abbreviation, definedAbbreviation
buttons.abbreviation.removeFieldsets = acronym,definedAcronym
```

- Property `inlineLanguageLabel` of object `PAGE` can handle `LLL:` references now

Chapter 3:

In-Depth Changes

In-Depth Changes

SVG Support

- TYPO3 CMS core supports SVG images now ("Scalable Vector Graphics")
- When an SVG image is scaled, a record with the calculated new dimensions is stored in `sys_file_processedfile` rather than creating a processed file
  (except, is image is processed further, e.g. cropping).
- A fallback is added to determine SVG dimensions if ImageMagick/GraphicsMagick can not determine image dimensions. In this case, the contents of the XML file is read.
- SVG has also been added to the list of valid image files:
  `$GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext']`

In-Depth Changes

Extending FAL Driver

- In order to improve the performance of the file list when showing (remote) storages the FAL driver should take care of sorting, ordering and determining the number of files/folders. Two new parameters `sort` and `sortRev` have been added to allow that:

```
public function getFilesInFolder($folderIdentifier, $start = 0, $numberOfItems = 0,
  $recursive = FALSE, array $filenameFilterCallbacks = array(), $sort = '', $sortRev = FALSE);

public function getFoldersInFolder($folderIdentifier, $start = 0, $numberOfItems = 0,
  $recursive = FALSE, array $folderNameFilterCallbacks = array(), $sort = '', $sortRev = FALSE);
```

- Additionally, two new methods have been implemented:

```
public function getFilesInFolderCount($folderIdentifier, $recursive = FALSE,
  array $filenameFilterCallbacks = array());

public function getFoldersInFolderCount($folderIdentifier, $recursive = FALSE,
  array $folderNameFilterCallbacks = array());
```

In-Depth Changes

Backend Routing API (1)

- A Backend Routing API has been implemented, which manages the backend Entry Points
- Inspired by the Symfony Routing Framework, this API is compatible with it to a large extent
  (however TYPO3 uses only approx. 20% at this point in time)
- Basically three classes implement the functionality:
  - `class Route`: contains details about path and options
  - `class Router`: API to match the route
  - `class UrlGenerator`: generates the URL

In-Depth Changes

Backend Routing API (2)

- Routes are defined in the following file of an extension:
  `Configuration/Backend/Routes.php`
  (see system extension `backend` as an example)
- Further details about the Backend Routing API:
  http://wiki.typo3.org/Blueprints/BackendRouting

In-Depth Changes

New System Extension for Media Content Elements

- New system extension "`mediace`" contains the following cObjects:
  - `MULTIMEDIA`
  - `MEDIA`
  - `SWFOBJECT`
  - `FLOWPLAYER`
  - `QTOBJECT`
- Content elements `media` and `multimedia` have been moved to this system extension, too, as well as the "Media Wizard Provider"
- This extension is **not** installed by default!

In-Depth Changes

Location of Third-party Libraries

- Composer-installed third-party libraries are installed under `typo3/contrib/vendor` now (TYPO3 CMS < 7.2: in folder `Packages/Libraries`)
- This way the packaging process for releasing TYPO3 CMS as tarball or zip can trigger a fully working installation without having to ship `Packages/` for third-party libraries
- Problems may occur with installations if they were set up via composer and use `phpunit` unless composer dependencies have been completely rebuilt. To fix this, execute:

```
# cd htdocs/
# rm -rf typo3/contrib/vendor/ bin/ Packages/Libraries/ composer.lock
# composer install
```

In-Depth Changes

JavaScript Notifications

- A new JavaScript Notification API has been implemented:

```
// old and deprecated:
top.TYPO3.Flashmessages.display(TYPO3.Severity.notice)

// new and the only correct way since TYPO3 CMS 7.2:
top.TYPO3.Notification.notice(title, message)
```

- The following API functions exist:
  (parameter `duration` is optional and features a default value of 5 seconds)
  - `top.TYPO3.Notification.notice(title, message, duration)`
  - `top.TYPO3.Notification.info(title, message, duration)`
  - `top.TYPO3.Notification.success(title, message, duration)`
  - `top.TYPO3.Notification.warning(title, message, duration)`
  - `top.TYPO3.Notification.error(title, message, duration)`

In-Depth Changes

System Information Dropdown (1)

- Custom system information items can be added to the dropdown by creating a slot
- The slot must be registered in file `ext_localconf.php`:

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

$signalSlotDispatcher->connect(
  \TYPO3\CMS\Backend\Backend\ToolbarItems\SystemInformationToolbarItem::class,
  'getSystemInformation',
  \Vendor\Extension\SystemInformation\Item::class,
  'getItem'
);
```

In-Depth Changes

System Information Dropdown (2)

- Custom system information items can be added to the dropdown by creating a slot
- This requires class `Item` and its method `getItem()` in file `EXT:extension\Classes\SystemInformation\Item.php`:

```
class Item {
  public function getItem() {
    return array(array(
      ’title’ => ’The title shown on hover’,
      ’value’ => ’Description shown in the list’,
      ’status’ => SystemInformationHookInterface::STATUS_OK,
      ’count’ => 4,
      ’icon’ => \TYPO3\CMS\Backend\Utility\IconUtility::getSpriteIcon(
    ’extensions-example-information-icon’)
    ));
  }
}
```

In-Depth Changes

System Information Dropdown (3)

- Icon `extensions-example-information-icon` must be registered in `ext_localconf.php`:

```
\TYPO3\CMS\Backend\Sprite\SpriteManager::addSingleIcons(
  array(
    ’information-icon’ => \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extRelPath(
      $_EXTKEY) . ’Resources/Public/Images/Icons/information-icon.png’
    ),
   $_EXTKEY
);
```

In-Depth Changes

System Information Dropdown (4)

- Messages are shown at the bottom of the dropdown
- Extensions can provide its own slot to fill messages:

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

$signalSlotDispatcher->connect(
  \TYPO3\CMS\Backend\Backend\ToolbarItems\SystemInformationToolbarItem::class,
  'loadMessages',
  \Vendor\Extension\SystemInformation\Message::class,
  'getMessage'
);
```

In-Depth Changes

System Information Dropdown (5)

- Messages are shown at the bottom of the dropdown
- This requires the class `Message` and its method `getMessage()` in file `EXT:extension\Classes\SystemInformation\Message.php`:

```
class Message {
  public function getMessage() {
    return array(array(
      ’status’ => SystemInformationHookInterface::STATUS_OK,
      ’text’ => ’Something went wrong. Take a look at the reports module.’
    ));
  }
}
```

In-Depth Changes

Image Manipulation Configuration Options (1)

- The following TypoScript configuration options are available:

```
# disable cropping for all images
tt_content.image.20.1.file.crop =

# override or set cropping for all images
# offsetX,offsetY,width,height
tt_content.image.20.1.file.crop = 50,50,100,100
```

- Fluid also supports cropping functions:

```
# disable cropping for all images
<f:image image="{imageObject}" crop="" ></f:image>

# override or set cropping for all images
# offsetX,offsetY,width,height
<f:image image="{imageObject}" crop="50,50,100,100" ></f:image>
```

In-Depth Changes

Image Manipulation Configuration Options (2)

- TCA features the image cropping functionality, too:
  - Column Type: `image_manipulation`
  - Config `file_field`: string (default: `uid_local`)
  - Config `enableZoom`: boolean (default: `FALSE`)
  - Config `allowedExtensions`: string
    (default: `$GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext']`)
  - Config `ratios`: array, default:

```
array(
  '1.7777777777777777' => '16:9',
  '1.3333333333333333' => '4:3',
  '1' => '1:1',
  'NaN' => 'Free'
)
```

In-Depth Changes

Additional Parameters for HTMLparser userFunc

- Additional parameters can be supplied to a userFunc of the HTMLparser:

```
myobj = TEXT
myobj.value = <a href="/" class="myclass">MyText</a>
myobj.HTMLparser.tags.a.fixAttrib.class {
  userFunc = Tx\MyExt\Myclass->htmlUserFunc
  userFunc.myparam = test
}
```

- Access to these parameters in an extension as follows:

```
function htmlUserFunc(array $params, HtmlParser $htmlParser) {
  // $params[’attributeValue’] contains the attribute value "myclass"
  // $params[’myparam’] is set to "test" in this example
  ...
}
```

In-Depth Changes

Locking API (1)

- New Locking API has been introduced, which provides various locking methods (SimpleFile, Semaphore, ...)
- A locking method must implement the `LockingStrategyInterface`:

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);
$locker = $lockFactory->createLocker('someId');
$locker->acquire() || die('Could not acquire lock.');
...
$locker->release();
```

In-Depth Changes

Locking API (2)

- Some methods also support non-blocking locks:

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);
$locker = $lockFactory->createLocker(
  'someId',
  LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
    LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK
);
try {
  $result = $locker->acquire(LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
        LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK);
  catch (\RuntimeException $e) {
  if ($e->getCode() === 1428700748) {
    // some process owns the lock
    // let's do something else meanwhile
    ...
  }
}
if ($result) {
  $locker->release();
}
```

In-Depth Changes

Signal after Extension Installation

- New signal has been implemented in method `\TYPO3\CMS\Extensionmanager\Utility\InstallUtility::install()` which emits as soon as an extension has been installed and all imports/updates finished

```
// execution
$this->emitAfterExtensionInstallSignal($extensionKey);

// methode
protected function emitAfterExtensionInstallSignal($extensionKey) {
  $this->signalSlotDispatcher->dispatch(
    __CLASS__,
    'afterExtensionInstall',
    array($extensionKey, $this)
  );
}
```

In-Depth Changes

Registry for Text Extraction (1)

- Multiple text extractors can be registered to allow dealing with different file types (e.g. Office, PDF files, etc.)
- TYPO3 core ships with an extractor for plain text files
- Every registered text extractor class needs to implement the `TextExtractorInterface`
- ...and the following methods:

  `canExtractText()`

  checks if text extraction from the given file is possible

  `extractText()`

  returns the file’s text content as a string

In-Depth Changes

Registry for Text Extraction (2)

- Text extractor registration in file `ext_localconf.php`:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
      getInstance();
$textExtractorRegistry->registerTextExtractor(
  \TYPO3\CMS\Core\Resource\TextExtraction\PlainTextExtractor::class
);
```

- Usage as follows:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
      getInstance();
$extractor = $textExtractorRegistry->getTextExtractor($file);
if($extractor !== NULL) {
  $content = $extractor->extractText($file);
}
```

In-Depth Changes

Miscellaneous

- Web libraries (such as Twitter Bootstrap, jQuery, Font Awesome, etc.) use "Bower" (`http://bower.io`) and are not part of the TYPO3 core Git repository anymore

`# bower install`	executes an installation
`# bower update`	executes an update

  (file `bower.json` is located in directory `Build/`)
- Scheduler CLI has received option "`-s`" to stop a running task
- The processing folder of a (remote) storage can be outside of the storage (useful in case of a read-only storage for instance)
- It is now possible to retrieve the page ID of the originally requested page: `$TSFE->getRequestedId()`

Chapter 4:

Extbase & Fluid

Extbase & Fluid

Callouts vs. FlashMessages

- Callouts (content info boxes) replace FlashMessages in several places in the backend to display context information
- This required a new Fluid ViewHelper `be.infobox`:

```
<f:be.infobox title="Message title">
   your box content
</f:be.infobox>

<f:be.infobox
   title="Message title"
   message="your box content"
   state="-2"
   iconName="check"
   disableIcon="TRUE" />
```

Extbase & Fluid

format.case ViewHelper

- New ViewHelper `format.case` changes casing of strings:
  - `upper`: transforms a string to "UPPERCASE"
  - `lower`: transforms a string to "lowercase"
  - `capital`: transforms a string to its first letter upper-cased
  - `uncapital`: transforms a string to its first letter lower-cased
- For example:

```
// transforms to "SOME TEXT WITH MIXED CASE"
<f:format.case>Some TeXt WiTh miXed cAse</f:format.case>

// transforms to "SomeString"
<f:format.case mode="capital">someString</f:format.case>
```

Extbase & Fluid

Miscellaneous

- Parameter `cHash` is not added to action URIs if the current request is not cached and the target action is not cacheable

Chapter 5:

Deprecated/Removed Functions

Deprecated/Removed Functions

FormEngine Refactoring (1)

- A major refactoring of FormEngine classes and its sub classes happend, which impacts properties and methods
- In TYPO3 CMS 7.2 many internal properties, such as `FormEngine->$defaultInputWidth` or `FormEngine->$RTEenabled` are ignored
- properties `FormEngine->$allowOverrideMatrix` and `SuggestElement->class` are *protected* now
- If format of `type=none` is set to user in TCA, the configured userFunc no longer gets an instance of FormEngine as parent object, but an instance of NoneElement

Deprecated/Removed Functions

FormEngine Refactoring (2)

- The following methods (and more) have been classified as **deprecated**:
  - `FormEngine->renderWizards()`
  - `FormEngine->dbFileIcons()`
  - `FormEngine->getClipboardElements()`
  - `FormEngine->getSingleField_typeNone_render()`
  - `FormEngine->formMaxWidth()`
  - `FormEngine->addItems()`
  - `SuggestElement->init()`
  - ...

Tip: review the `deprecation_*.log` to find out, where these methods have been ported to, in case your extension(s) use the FormEngine.

Deprecated/Removed Functions

FormEngine Refactoring (3)

- The following methods have been **renamed**:

```
OLD:  \TYPO3\CMS\Backend\Form\Element\SuggestElement
NEW:  \TYPO3\CMS\Backend\Form\Wizard\SuggestWizard

OLD:  \TYPO3\CMS\Backend\Form\Element\SuggestDefaultReceiver
NEW:  \TYPO3\CMS\Backend\Form\Wizard\SuggestWizardDefaultReceiver

OLD:  \TYPO3\CMS\Backend\Form\Element\VaueSlider
NEW:  \TYPO3\CMS\Backend\Form\Wizard\ValueSliderWizard
```

Deprecated/Removed Functions

Backend Entry Points

- The following Backend Entry Points are changed:
  - `typo3/logout.php` (`logout`)
  - `typo3/alt_file_navframe.php` (`file_navframe`)
  - `typo3/dummy.php` (`dummy`)
  - `typo3/browser.php` (`browser`)
  - `typo3/db_new.php` (`db_new`)
  - `typo3/show_item.php` (`show_item`)
- URLs can be determined by using the following approach:

  ```
  \TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl(...)
  ```

  For example:

  ```
  \TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl('logout')
  ```

Deprecated/Removed Functions

Backend Login Refactoring

- Due to a refactoring of the backend login, Fluid is used as the template engine and the signal `LoginController::SIGNAL_RenderLoginForm` has been **removed**
- Additionally the following methods of the LoginController have been **removed**:
  - `LoginController::makeLoginBoxImage`
  - `LoginController::wrapLoginForm`
  - `LoginController::makeLoginNews`
  - `LoginController::makeLoginForm`
  - `LoginController::makeLogoutForm`

Deprecated/Removed Functions

Miscellaneous (1)

- In TYPO3 CMS < 7.2 URIs to a module which was dispatched through `mod.php` was stored as an array in a global variable `$GLOBALS['MCONF']['_']`.
  This has been **removed** without substitution and extensions need to use `BackendUtility::getModuleUrl()` instead.
- The option to retrieve the local path of a FAL file via TypoScript has been **removed**: `a.value.data = file:current:localPath`
- The following `tt_content` TCA fields have been moved to `EXT:compatibility6`:
  - `altText`
  - `imagecaption`
  - `imagecaption_position`
  - `image_link`
  - `longdescURL`
  - `titleText`

Deprecated/Removed Functions

Miscellaneous (2)

- The third-party library `websvg` has been **removed** from the TYPO3 CMS core. The according TypoScript options (`page.javascriptLibs.SVG.*`) and public methods within PageRenderer have been **removed** without substitution, too.
  For example: `$pageRenderer->loadSvg()`
- The following keys under `mod.web_info.menu.function` have been **renamed** (this has an impact on PageTSconfig):

```
tx_cms_webinfo_page -> TYPO3\CMS\Frontend\Controller\PageInformationController
tx_cms_webinfo_lang -> TYPO3\CMS\Frontend\Controller\TranslationStatusController
tx_belog_webinfo -> TYPO3\CMS\Belog\Module\BackendLogModuleBootstrap
tx_infopagetsconfig_webinfo -> TYPO3\CMS\InfoPagetsconfig\Controller\
      InfoPageTyposcriptConfigController
tx_linkvalidator_ModFuncReport -> TYPO3\CMS\Linkvalidator\Report\LinkValidatorReport
```

Deprecated/Removed Functions

Miscellaneous (3)

- Parameter `$returnHTML` of method `\TYPO3\CMS\Core\Utility\DebugUtility::debugRows()` is not used anymore and has been marked **deprecated**
- TypoScript option `config.prefixLocalAnchors` has been marked as **deprecated**, as well as the appropriate methods in TypoScriptFrontendContoller: `prefixLocalAnchorsWithScript()` and `doLocalAnchorFix()`
- Public method `$TSFE->checkFileInclude()` in the global FrontendController is **deprecated** now. Use the autoloader or `$TSFE->tmpl->getFileName()` instead.

Deprecated/Removed Functions

Miscellaneous (4)

- The two cObjects `COBJ_ARRAY` (alias for `COA`) and `CASEFUNC` (alias for `CASE`) have been moved to extension `EXT:compatibility6` (and marked as **deprecated**) and are not available by default anymore
- The DataHandler property `stripslashes_values` has been marked as **deprecated**
- The "ThumbnailView" as well as `thumbs.php` and `BackendUtility::getThumbNail()` are **deprecated** now and will be removed in TYPO3 CMS Version 8
  (see `BackendUtility::thumbCode()` for migration inspirations)

Deprecated/Removed Functions

Miscellaneous (5)

- Namespace `mod.tx_linkvalidator` of the LinkValidator Scheduler Task changed to `mod.linkvalidator` in order to make the setting consistent with TSconfig

Chapter 6:

Sources and Authors

Sources and Authors

Sources (1)

TYPO3 News:

- http://typo3.org/news

Release Infos:

- http://wiki.typo3.org/TYPO3_CMS_7.2.0
- INSTALL.md and ChangeLog
- typo3/sysext/core/Documentation/Changelog/7.2/*

TYPO3 Bug-/Issuetracker:

- https://forge.typo3.org/projects/typo3cms-core

TYPO3 Git Repositories:

- https://git.typo3.org/Packages/TYPO3.CMS.git
- https://git.typo3.org/Packages/TYPO3.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, Paul Blondiaux, Sergio Catala, Michel Mix,
Sinisa Mitrovic, Angeliki Plati, Nena Jelena Radovic und Roberto Torresani

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