

TYP03 CMS 8.6 – What's New

Summary of the new features, changes and improvements

Created by:

Patrick Lobacher and Pierrick Caillon

26/February/2017

Creative Commons BY-NC-SA 3.0



TYPO3 CMS 8.6 - What's New

Chapter Overview

Introduction

Backend User Interface

TScnfig & TypoScript

In-Depth Changes

Extbase & Fluid

Deprecated/Removed Functions

Sources and Authors

Introduction

The Facts

Introduction

TYPO3 CMS 8.6 – The Facts

- Release date: 14 February 2017
- Release type: Sprint Release
- Slogan: "Polishing"



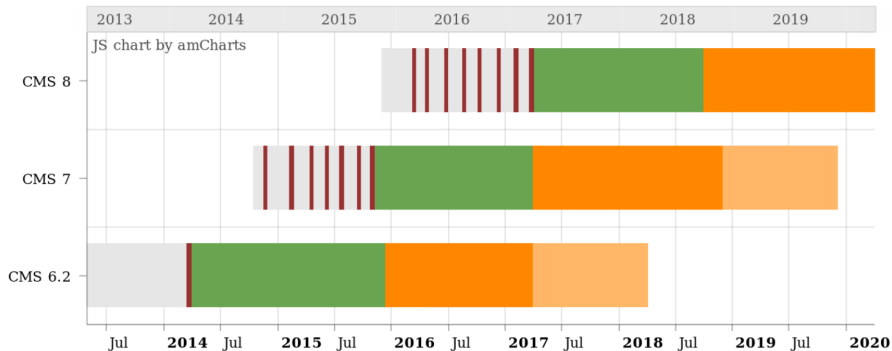
Introduction

System Requirements

- PHP: version 7
- MySQL: version 5.5 to 5.7
- Disk space: min 200 MB
- PHP settings:
 - `memory_limit` \geq 128M
 - `max_execution_time` \geq 240s
 - `max_input_vars` \geq 1500
 - compilation option `--disable-ipv6` must not be used
- The backend requires Microsoft Internet Explorer 11 or later, Microsoft Edge, Google Chrome, Firefox, Safari or any other modern, compatible browser

Introduction

Development and Release Timeline



Introduction

TYPO3 CMS Roadmap

Release dates and their primary focus:

- v8.0 22/Mar/2016 Adding last minute things
- v8.1 03/May/2016 Cloud Integration
- v8.2 05/Jul/2016 Doctrine Prerequisites
- v8.3 30/Aug/2016 Rich Text Editor
- v8.4 18/Oct/2016 Doctrine Migration + Upgrades
- v8.5 20/Dec/2016 New RTE + Integrator Support
- v8.6 14/Feb/2017 Polishing
- v8.7 04/Apr/2017 LTS Preparation

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

<https://typo3.org/news/article/kicking-off-typo3-v8-development/>

Introduction

Installation

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

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/8.6
$ tar xzf typo3_src-8.6.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-8.6.0 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 8.x

- Upgrades only possible from TYPO3 CMS 7.6 LTS
- TYPO3 CMS < 7.6 LTS should be updated to TYPO3 CMS 7.6 LTS first
- Upgrade instructions:
http://wiki.typo3.org/Upgrade#Upgrading_to_8.6
- 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)

Introduction

PHP Version 7

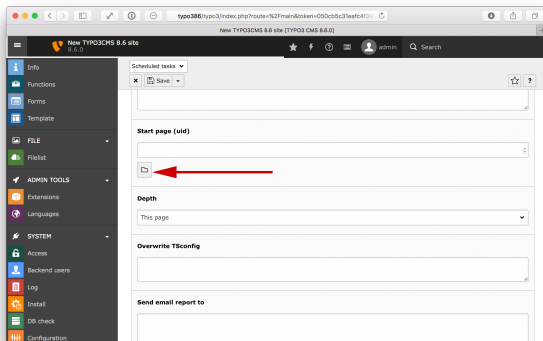
- PHP 7.0 is the minimum requirement for TYPO3 CMS 8.x
- TYPO3 will support subsequent PHP 7 releases as they come out
- This version raise gives a significant performance boost to the overall system
- Not only backend editors will notice a more fluent interface, but the new all-time record for a full cached page call in the frontend is below 7 milliseconds now, which is approximately 40% faster compared to running the very same website with PHP version 5.5
- We also started using new features from this PHP version, for instance the cryptographically secure pseudo-random generators are in active use already

Chapter 1: Backend User Interface

Backend User Interface

Scheduler Page Browser

To improve the usability of the **EXT:linkvalidator** scheduler task, the page browser was added to select the start page.



Backend User Interface

Run Manually Executed Tasks on Next Cron-run

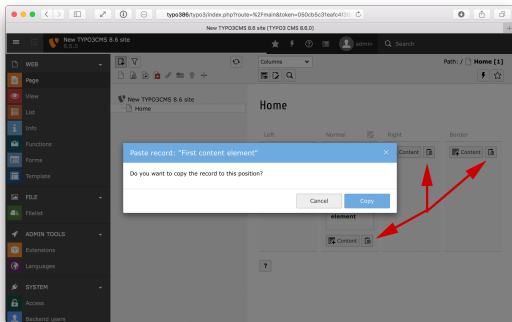
There is a new action icon to mark a task to be run by cron. Also a new button "Execute selected tasks on next cron job" has been added to mark all selected actions to be run by next cron job.

The screenshot displays the TYPO3 CMS 8.6 Backend User Interface. The left sidebar contains navigation menus for 'Info', 'Functions', 'Forms', 'Template', 'FILE', 'ADMIN TOOLS', and 'SYSTEM'. The main content area is titled 'Scheduled tasks' and features a table with the following columns: ID, Task, Type, Frequency, Parallel Execution, Last Execution, and Next Execution. A single task is listed with ID '1', Task 'Linkvalidator (linkvalidator)', Type 'Recurring', Frequency '10', and Next Execution '15-02-17 08:08'. The task's action menu includes a new icon for 'Run task on next cron job'. Below the table, there are two buttons: 'Execute selected tasks' and 'Execute selected tasks on next cron job'. A 'Server time' notification is visible at the bottom, stating 'All dates and times in the Scheduler are measured according to the server's time, as the Scheduler is run purely on the server-side. Current server time is 15-02-17 08:09 CET (Europe/Berlin, GMT +01:00)'.

Backend User Interface

Paste Icons and Modal

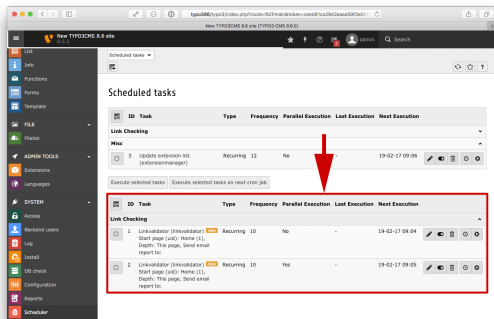
As soon as the normal clipboard contains an item, a single paste icon becomes available in the page module. When the user clicks on the icon, a modal pops up to have the user confirm the action.



Backend User Interface

Folding of Scheduler Task Groups

When task groups are used, the tasks are displayed grouped in the list of tasks. Clicking on the row with the group title hides or shows the tasks of the group now.



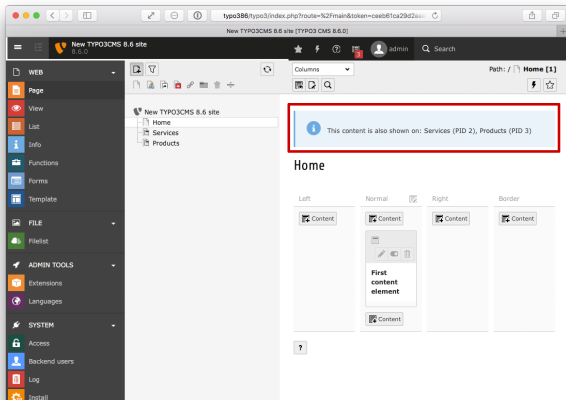
The screenshot shows the TYPO3 CMS 8.6 Backend Scheduler interface. The left sidebar contains navigation options like 'List', 'Info', 'Functions', 'Forms', 'Templates', 'File', 'Stats', 'ADMIN TOOLS', 'Extensions', 'Languages', 'SYSTEM', 'Access', 'Backend users', 'Log', 'Install', 'ID check', 'Configuration', 'Reports', and 'Scheduler'. The main content area is titled 'Scheduled tasks' and contains a table of tasks. A red arrow points to the 'Link Checking' group title, which is highlighted in a red box. The table has columns for ID, Task, Type, Frequency, Parallel Execution, Last Execution, and Next Execution.

ID	Task	Type	Frequency	Parallel Execution	Last Execution	Next Execution
Link Checking						
Misc						
3	Update extension list (extensionmanager)	Recurring	12	No	19-02-17 09:06	
Execute selected tasks Execute selected tasks on next cron job						
Link Checking						
1	Linkvalidator (linkvalidator) Start page (url): Home (1), Description: This page sends email report to:	Recurring	10	No	19-02-17 09:04	
2	Linkvalidator (linkvalidator) Start page (url): Home (1), Description: This page sends email report to:	Recurring	10	Yes	19-02-17 09:05	

Backend User Interface

Page Module Notice "Content is also shown on"

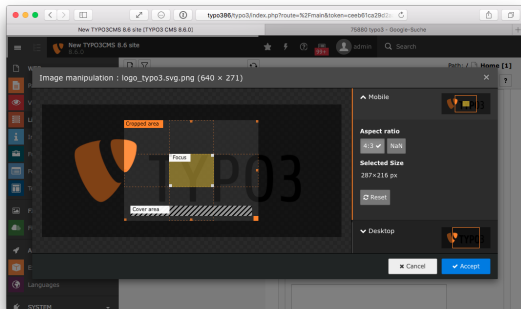
When page content is inherited from a different page via "Show content from page", a notice is displayed on the page that is pulling in content to a different page.



Backend User Interface

Image Manipulation - Multiple Cropping Variants

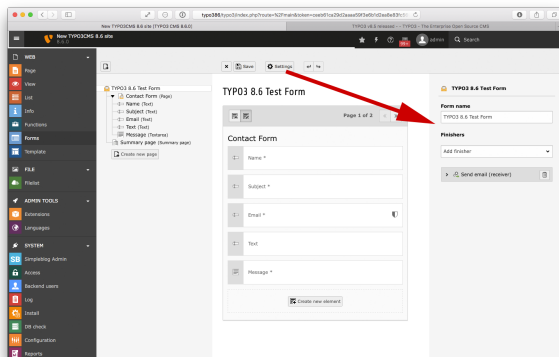
The Image-Manipulation tool is now capable of handling multiple crop variants (if configured). Users can also select a focus area, which is always inside the crop area and mark the area in the image which must be visible for the image to transport its meaning. To give editors a hint which area of the image is used by other DOM-Elements like headlines, when selecting a crop area, it is possible to define multiple so called cover areas.



Backend User Interface

EXT:form: add form settings button to module header

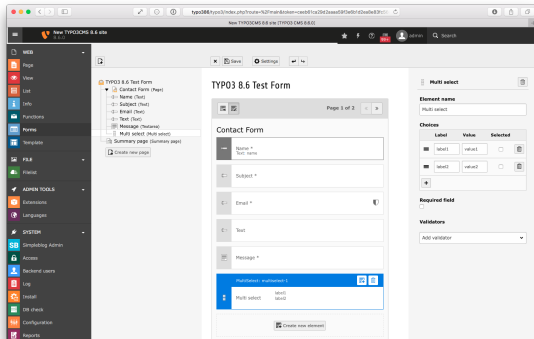
A new button has been added to the module header of the form editor. Clicking on this button shows the form settings within the inspector.



Backend User Interface

EXT:form: Add multiselect inspector editor

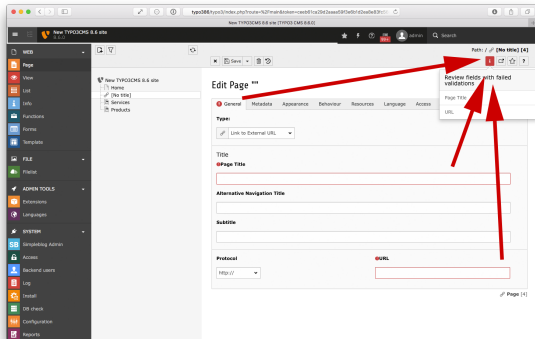
A new inspector editor, i.e. a new field type of the form editor, has been added. If applied, multi-select fields can be added to the inspector. A multi-select field allows the selection of multiple meta properties for a field and stores them in the defined property path.



Backend User Interface

Show list of failed input elements in FormEngine

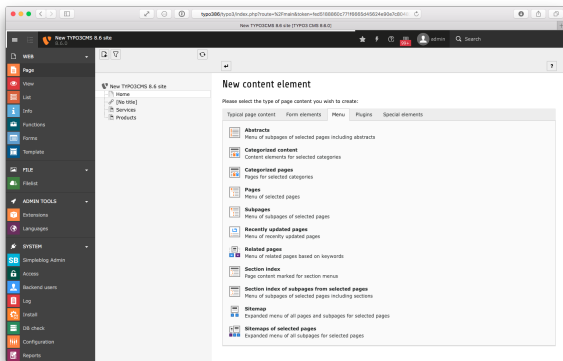
When validating input fields of the FormEngine fails, a button is now rendered into the button bar in the module document header. Clicking the button renders a list of all input elements whose validation failed. Clicking onto a field in that list automatically focuses the field in the form.



Backend User Interface

Dedicated content elements for menus

For better maintainability the currently existing content element menu has been split into dedicated content elements.



Chapter 2: TSconfig & TypoScript

TSconfig & TypoScript

Section Frame for CSS Styled Content replaced with Frame Class

- The functionality provided by Section Frame has been streamlined with Fluid Styled Content and is now available as Frame Class.
- For this, the TypoScript keys now use the right part from the CSS class `csc-frame-` instead of numbers.

■ Before:

```
tt_content.stdWrap.innerWrap.cObject.key.field = section_frame
tt_content.stdWrap.innerWrap.cObject.5 =< tt_content.stdWrap.innerWrap.cObject.default
tt_content.stdWrap.innerWrap.cObject.5.20.10.value = csc-frame csc-frame-ruler-before
```

■ After:

```
tt_content.stdWrap.innerWrap.cObject.key.field = frame_class
tt_content.stdWrap.innerWrap.cObject.ruler-before =< tt_content.stdWrap.innerWrap.cObject.default
tt_content.stdWrap.innerWrap.cObject.ruler-before.20.10.value = csc-frame csc-frame-ruler-before
```


Chapter 3: In-Depth Changes

In-Depth Changes

Page Browser for scheduler tasks

- Scheduler tasks that need a page uid can now add a button for the page browser popup.
- In the `ValidatorTaskAdditionalFieldProvider` two additional fields have to be added.
- If the additional field browser is set to page then the `SchedulerModuleController` adds a button for calling the page browser popup to the field.

```
'browser' => 'page',
```

- The `pageTitle` contains the title of the page that is shown next to the browse button.

```
'pageTitle' => $pageTitle,
```

In-Depth Changes

Synchronized field values in localized records (1/2)

- The localized record overlay behaviour has been changed to make localization-rows standalone.
- Previously, if fields in TCA columns were set to `l10n_mode exclude` or `mergeIfNotBlank`, the localized record overlay did not contain values, and those values were "pulled up" from the underlying default language records.

In-Depth Changes

Synchronized field values in localized records (2/2)

- This has been changed, the DataHandler now copies those values over to the localized record and synchronizes them if the default language record is changed.

```
'columns' => [  
  ...  
  'header' => [  
    'label' => 'My header',  
    'config' => [  
      'type' => 'input',  
      'behaviour' => [  
        'allowLanguageSynchronization' => true,  
      ],  
    ],  
  ],  
],  
],
```

In-Depth Changes

Image Manipulation Tool (1/6)

- The `imageManipulation` TCA type is now capable of handling multiple crop variants if configured.
- It is also possible to define an initial crop area. If no initial crop area is defined, the default selected crop area will cover the complete image.
- Crop areas are defined relatively with floating point numbers. The coordinates and sizes must be specified for that.

In-Depth Changes

Image Manipulation Tool (2/6)

- The following example configures two crop variants, one with the id "mobile", one with the id "desktop". The array key defines the id, which will be used when rendering an image with the image view helper.

```
'config' => [  
  'type' => 'imageManipulation',  
  'cropVariants' => [  
    'mobile' => [  
      'title' => 'Mobile',  
      'allowedAspectRatios' => [  
        '4:3' => [  
          'title' => '4:3',  
          'value' => 4 / 3  
        ],  
        ...  
      ],  
    ],  
    'desktop' => [  
      ...  
    ],  
  ],  
]
```

In-Depth Changes

Image Manipulation Tool (3/6)

- The below example has an initial crop area in the size of the previous image cropper provided by default.

```
'config' => [  
  'type' => 'imageManipulation',  
  'cropVariants' => [  
    'mobile' => [  
      'title' => 'LLL:EXT:ext_key/Resources/Private/Language/locallang.xlf:imageManipulation.  
        mobile',  
      'cropArea' => [  
        'x' => 0.1,  
        'y' => 0.1,  
        'width' => 0.8,  
        'height' => 0.8,  
      ],  
    ],  
  ],  
]
```

In-Depth Changes

Image Manipulation Tool (4/6)

- Users can also select a focus area, when configured.
- The focus area is always inside the crop area and mark the area in the image which must be visible for the image to transport its meaning.

```
'config' => [  
  'type' => 'imageManipulation',  
  'cropVariants' => [  
    'mobile' => [  
      'title' =>  
        'LLL:EXT:ext_key/Resources/Private/Language/locallang.xlf:imageManipulation.mobile',  
      'focusArea' => [  
        'x' => 1 / 3,  
        'y' => 1 / 3,  
        'width' => 1 / 3,  
        'height' => 1 / 3,  
      ],  
    ],  
  ],  
]
```


In-Depth Changes

Image Manipulation Tool (5/6)

- Very often images are used in a context, where there are overlaid with other DOM elements like a headline.
- To give editors a hint which area of the image is affected, when selecting a crop area, it is possible to define multiple cover areas.
- These areas are shown inside the crop area. The focus area cannot intersect with any of the cover areas.

```
'config' => [  
  'type' => 'imageManipulation',  
  'coverAreas' => [  
    [  
      'x' => 0.05, 'y' => 0.85,  
      'width' => 0.9, 'height' => 0.1,  
    ],  
  ],  
]
```

In-Depth Changes

Image Manipulation Tool (6/6)

- To render crop variants, the variants can be specified as argument to the image view helper:

```
<f:image image="{data.image}" cropVariant="mobile" width="800" >  
</f:image>
```

In-Depth Changes

Default Content Element Changed for Fluid Styled Content

- The default content element has been streamlined with CSS Styled Content and has been changed to "Text"
- To restore the configuration you need to set the default content element manually to your preferred choice. You can do this by simply overriding the configuration again in your `Configuration/TCA/Overrides/tt_content.php` file.

```
$GLOBALS['TCA']['tt_content']['columns']['CType']['config']['default'] = 'textmedia';  
$GLOBALS['TCA']['tt_content']['columns']['CType']['config']['default'] = 'header';
```

In-Depth Changes

TCA Changes (1/2)

- The TCA on field level has been changed.
- Nearly all column types are affected.
- In general, the sub-section `wizards` is gone and replaced by a combination of new `renderType` and a new set of configuration options.
- Wizards are now divided into three different kinds: `fieldInformation`, `fieldControl` and `fieldWizard`.

In-Depth Changes

TCA Changes (2/2)

■ Example:

```
'fieldControl' => [  
  'editPopup' => [  
    'disabled' => false,  
  ],  
  'addRecord' => [  
    'disabled' => false,  
    'options' => [  
      'setValue' => 'prepend',  
    ],  
  ],  
  'listModule' => [  
    'disabled' => false,  
  ],  
],
```

- You find further details at docs.typo3.org

In-Depth Changes

Introduce Session Storage Framework

- A new session storage framework has been introduced
- The goal of this framework is to create interoperability between different session storages (called "backends") like database, file storage, Redis, etc.
- The following session backends are available by default:
 - `\TYPO3\CMS\Core\Session\Backend\DatabaseSessionBackend`
 - `\TYPO3\CMS\Core\Session\Backend\RedisSessionBackend`

In-Depth Changes

CLI Support for T3D Imports

- EXT: `impexp` now allows to import data files (T3D or XML) via the command line interface through a Symfony Command.

- Usage:

```
./typo3/sysext/core/bin/typo3 impexp:import [<options>] <file>  
<pageId>
```

- Options:

- `--updateRecords`: Force updating existing records
- `--ignorePid`: Don't correct page ids of updated records
- `--enableLog`: log all database action

In-Depth Changes

Implement Hook in `typolink` for Modification of Page Params

- A new hook has been implemented in `ContentObjectRenderer::typoLink` for links to pages. With this hook you can modify the link configuration, for example enriching it with additional parameters or meta data from the page row.
- You can now register a hook via:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['typolinkProcessing']  
  ['typolinkModifyParameterForPageLinks'][] = \Your\Namespace\Hooks\MyHook::class;
```

- Usage:

```
public function modifyPageLinkConfiguration(  
    array $linkConfiguration, array $linkDetails, array $pageRow) : array  
{  
    $linkConfiguration['additionalParams'] .= $pageRow['myAdditionalParamsField'];  
    return $linkConfiguration;  
}
```


In-Depth Changes

Hook to Add Custom TypoScript Templates (1/2)

- A new hook in TemplateService allows to add or modify existing TypoScript templates.
- You can now register a hook via the following code in the extensions `ext_localconf.php` file:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['Core/TypoScript/TemplateService']  
    ['runThroughTemplatesPostProcessing']
```

- `EXT:my_site/Classes/Hooks/TypoScriptHook.php (1/2)`

```
namespace MyVendor\MySite\Hooks;  
class TypoScriptHook  
{  
    /**  
     * Hooks into TemplateService after  
     * @param array $parameters  
     * @param \TYPO3\CMS\Core\TypoScript\TemplateService $parentObject  
     * @return void  
     */  
    ...  
}
```

In-Depth Changes

Hook to Add Custom TypoScript Templates (2/2)

■ EXT:my_site/Classes/Hooks/TypoScriptHook.php (2/2)

```
...
public function addCustomTypoScriptTemplate($parameters, $parentObject)
{
    // Disable the inclusion of default TypoScript set via TYPO3_CONF_VARS
    $parameters['isDefaultTypoScriptAdded'] = true;
    // Disable the inclusion of ext_typoscript_setup.txt of all extensions
    $parameters['processExtensionStatics'] = false;

    // No template was found in rootline so far, so a custom "fake" sys_template record is added
    if ($parentObject->outermostRootlineIndexWithTemplate === 0) {
        $row = [
            'uid' => 'my_site_template',
            'config' =>
                '<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_site/Configuration/TypoScript/site_setup.t3s">',
            'root' => 1,
            'pid' => 0
        ];
        $parentObject->processTemplate($row, 'sys_' . $row['uid'], 0, 'sys_' . $row['uid']);
    }
}
}
```

In-Depth Changes

Plugin Preview with Fluid

- The page TSconfig to render a preview of a single content element in the Backend has been improved by allowing the rendering of plugins via Fluid as well
- All properties of the `tt_content` record are available in the template directly (e.g. UID via `{uid}`)
- Any data of the flexform field `pi_flexform` is available with the property `pi_flexform_transformed` as an array.

```
mod.web_layout.tt_content.preview.list.simpleblog_bloglisting =  
EXT:simpleblog/Resources/Private/Templates/Preview/SimpleblogPlugin.html
```

In-Depth Changes

Template Paths in BackendTemplateView

- BackendTemplateView now allows overwriting of template paths to add your own locations for templates, partials and layouts in a BackendTemplateView based backend module.

```
$frameworkConfiguration =  
    $this->configurationManager->getConfiguration(  
        ConfigurationManagerInterface::CONFIGURATION_TYPE_FRAMEWORK  
    );  
$viewConfiguration = [  
    'view' => [  
        'templateRootPaths' => ['EXT:myext/Resources/Private/Backend/Templates'],  
        'partialRootPaths' => ['EXT:myext/Resources/Private/Backend/Partials'],  
        'layoutRootPaths' => ['EXT:myext/Resources/Private/Backend/Layouts'],  
    ],  
];  
$this->configurationManager->setConfiguration(  
    array_merge($frameworkConfiguration, $viewConfiguration)  
);
```

In-Depth Changes

Miscellaneous

- The TCA config setting `maxitems` for `type=select` and `type=group` fields is now an optional setting that defaults to a high value (99999) instead of 1 as before.
- Accessing TYPO3 functionality from the command line has been simplified. Single commands no longer require single users in the database, instead all cli command use the username `_cli_`. This user is created on demand by the framework if it does not exist at the first command line call. The `_cli_` user has admin rights and no longer needs specific access rights assigned to perform specific tasks like manipulating database content using the `DataHandler`.

Chapter 4: Extbase & Fluid

Extbase & Fluid

Widget Identifier Extended

- The parameter `customWidgetId` has been introduced for fluid widgets. This string is used in the widget identifier in addition to the `nextWidgetNumber`.
- The widget identifier is used to create the GET parameter names.
- A good value for the `customWidgetId` is `contentObjectData.uid` to ensure no collisions happen.
- Allows to use the same fluid widget more than once on one page in different content elements.

```
<f:widget.paginate customWidgetId="{contentObjectData.uid}" ...>  
</f:widget.paginate>
```

FlashMessageViewHelper

- The FlashMessageViewHelper has been refactored and no longer inherits from the TagBasedViewHelper
- Remove the tag specific attributes and style the default output. If you need custom output use the possibility to render FlashMessages yourself, for example:

```
<f:flashMessages as="flashMessages">
  <dl class="messages">
    <f:for each="{flashMessages}" as="flashMessage">
      <dt>CODE!! {flashMessage.code}</dt>
      <dd>MESSAGE:: {flashMessage.message}</dd>
    </f:for>
  </dl>
</f:flashMessages>
```


Removal of Fluid Styled Content Menu ViewHelpers (1/3)

- Fetching data directly in the view is not recommended and the temporary solution of menu ViewHelpers has been replaced by its successor, the menu processor that is based on HMENU.
- Menu ViewHelpers have been moved to the `compatibility7` extension, and are replaced in the core menu content elements.

Removal of Fluid Styled Content Menu ViewHelpers (2/3)

■ Before:

```
tt_content.menu_subpages.dataProcessing {
    10 = TYPO3\CMS\Frontend\DataProcessing\SplitProcessor
    10 {
        if.isTrue.field = pages
        fieldName = pages
        delimiter = ,
        removeEmptyEntries = 1
        filterIntegers = 1
        filterUnique = 1
        as = pageUids
    }
}

<ce:menu.directory pageUids="{pageUids}" as="pages" levelAs="level">
    <f:for each="{pages}" as="page">
        ...
    </f:for>
</ce:menu.directory>
```

Removal of Fluid Styled Content Menu ViewHelpers (3/3)

■ After:

```
tt_content.menu_subpages.dataProcessing {
    10 = TYPO3\CMS\Frontend\DataProcessing\MenuProcessor
    10.special = directory
    10.special.value.field = pages
}

<f:for each="{menu}" as="page">
    ...
</f:for>
```

Extbase & Fluid

New Fluid ViewHelper `f:variable`

- A new ViewHelper `f:variable` has been added in Fluid 2.2.0, which is now minimum required dependency for TYPO3 CMS
- The ViewHelper allows variables to be assigned in the template:

```
<f:variable name="myvariable">My variable's content</f:variable>  
<f:variable name="myvariable" value="My variable's content"></f:variable>  
{f:variable(name: 'myvariable', value: 'My variable\'s content')}  
{myoriginalvariable -> f:variable(name: 'mynewvariable')}
```

Extbase & Fluid

New Default Layout for Fluid Styled Content (1/2)

- Previously, there have been three layouts you could choose from when you were defining your own custom content elements or overriding an existing template.
- To provide a better maintainability and ease of use of overrides we are reducing these to a single layout that is named `Default` with all sections optional and fallbacks if the section is not set. Also we are introducing the "DropIn" concept.
- The `Default` layout consists of five predefined sections that can be utilized to shape the output for your content rendering. In most cases you will not have to care about other section than `Main`. The sections will be rendered in that exact ordering: `Before`, `Header`, `Main`, `Footer`, `After`

Extbase & Fluid

New Default Layout for Fluid Styled Content (2/2)

- The sections `Before` and `After` are so called "DropIn" sections
- DropIns have been introduced to be able to place additional functionality to all content elements without overriding layouts or templates
- DropIns are basically placeholders/empty partials that are meant to be overridden if necessary
- DropIn Locations:
 - `Resources/Private/Partials/DropIn/Before/All.html`
 - `Resources/Private/Partials/DropIn/After/All.html`

Chapter 5: Deprecated/Removed Functions

Deprecated/Removed Functions

Optional Fluid TypeScript Template removed

- The static include file "Fluid: (Optional) default ajax configuration (fluid)" was meant as an example/showcase on how to use Fluid Widgets in FE. It is outdated and therefore removed.
- Instead include the needed file manually in your TypeScript template:

```
page.includeJSLibs {
    jquery = https://code.jquery.com/jquery-3.1.1.slim.min.js
    jquery.external = 1
    jquery.integrity = sha256-SlIrNqv8h6QGKDuNoLGA4iret+kyesCkHGzVUUUV0shc=
    jqueryUi = https://code.jquery.com/ui/1.12.1/jquery-ui.min.js
    jqueryUi.external = 1
    jqueryUi.integrity = sha256-VazP97ZCwtekAsvgPBSUwPFKdrwD3unUfSGVYrahUqU=
}

page.includeCSSLibs {
    jqueryUI = https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css
    jqueryUi.external = 1
}
```


Deprecated/Removed Functions

Lowlevel VersionsCommand Parameters Changed (1/2)

- The existing CLI command within `EXT:lowlevel` for showing and cleaning up versions (from `EXT:version / EXT:workspaces`) has been migrated to a Symfony Console command.
- The command previously available via `./typo3/cli_dispatch.phpsh lowlevel_cleaner versions` is now available via `./typo3/sysext/core/bin/typo3 cleanup:versions` and allows the following CLI options to be set:
 - `-v` and `-vv` to show more detailed information on the records affected
 - `--pid=23` or `-p=23` to only find versions with page ID 23 (otherwise "0" is taken)

Continued on next slide

Deprecated/Removed Functions

Lowlevel VersionsCommand Parameters Changed (2/2)

Continued

- ...
 - `--depth=4` or `-d=4` to only clean recursively until a certain page tree level
 - `--dry-run` to only show the records to be changed / deleted
 - `--action=nameofaction` to clean up versioned records, one of the following actions are possible:
 - `versions_in_live`: Delete versioned records in the live workspace
 - `published_versions`: Delete versions of published records
 - `invalid_workspace`: Move records inside a non-existing workspace ID into the live workspace
 - `unused_placeholders`: Remove placeholders which are not used anymore from the database

Deprecated/Removed Functions

Default Layouts for Fluid Styled Content Changed

- The content element layouts for Fluid Styled Content have been changed to provide a better maintainability and to be more flexible.
- Previously available content element layouts `ContentFooter`, `HeaderFooter` and `HeaderContentFooter` have been dropped and replaced with a single `Default` layout that is more flexible.

```
$GLOBALS['TCA']['tt_content']['columns']['CType']['config']['default'] = 'textmedia';  
$GLOBALS['TCA']['tt_content']['columns']['CType']['config']['default'] = 'header';
```

Deprecated/Removed Functions

TypoScript Standard Header (1/2)

- The TypoScript standard header rendering definition `lib.stdHeader` has been introduced in CSS Styled Content to reference it across multiple content elements to simplify maintenance.
- For Fluid Styled Content a workaround for compatibility with CMS 7 has been introduced to simplify migration. However, it only renders the header and misses all frames, and additional options are necessary to generate a streamlined rendering output if the content element layout was not implemented correctly.

Deprecated/Removed Functions

TypoScript Standard Header (2/2)

■ Output now:

```
tt_content.simple_content = COA
tt_content.simple_content {
    10 < lib.stdHeader
    20 = TEXT
    20.field = bodytext
}

<header>
    <h1>Nunc vel libero dignissim</h1>
</header>
<p>
    ...
</p>
```

Deprecated/Removed Functions

Miscellaneous (1/4)

- The following method has been removed:
`FlashMessage->getMessageAsMarkup()`
- `EXT:felogin` doesn't add default CSS styles anymore because it may break the frontend output, for example if CSS frameworks are used.
- The frontend specific TypoScript setup for `EXT:form` isn't loaded automatically anymore and must be added manually through static includes. With this change a TYPO3 integrator can easier decide where the extension TypoScript is included.
- The setting `noCopy` has been removed without replacement from the list of possible values of the TCA column property `l10n_mode`.
- The setting `mergeIfNotBlank` has been removed without replacement from the list of possible values of the TCA column property `l10n_mode`.

Deprecated/Removed Functions

Miscellaneous (2/4)

- The TypoScript setting `config.sys_language_softMergeIfNotBlank` has been removed without any replacement. This is a result of removing the TCA setting `mergeIfNotBlank` from the list of possible values for `l10n_mode`.
- The database field `pages.url_scheme` functionality has been moved to the `compatibility7` extension. The field allows to force the HTTP or HTTPS protocol for a specific page to be set by an editor in the page properties on a per-page basis. However, it is common today to ensure (if a SSL certificate is available) to use HTTPS for a whole website or even only for a specific area (inc. subpages) to force the protocol.

Deprecated/Removed Functions

Miscellaneous (3/4)

- The function `QuickEdit` in the page module has been moved to `EXT:compatibility7` and will not be developed further. See [TYPO3 Extension Repository \(TER\)](#).
- In order to streamline CSS Styled Content and Fluid Styled Content, several options of CSS Styled Content have been dropped without replacement: `TCA image_compression`, `TCA image_effects`, `TCA image_noRows`, `TypoScript IMAGE noRows`, `TypoScript IMAGE noCols`, `TypoScript IMAGE noRowsStdWrap`, `TypoScript IMGTEXT captionAlign`
- The field `select_key` of the table `tt_content` is not used in the core and has been removed.

Deprecated/Removed Functions

Miscellaneous (4/4)

- The following methods and properties within `FlashMessage::class` have been marked as deprecated:
 - `FlashMessage->classes`
 - `FlashMessage->icons`
 - `FlashMessage->getClass()`
 - `FlashMessage->getIconName()`
- Method `ArrayUtility::inArray()` has been marked deprecated
- CSS Styled Content is now deprecated
(will be removed with TYPO3 CMS version 9)

Chapter 6: Sources and Authors

Sources and Authors

Sources

TYPO3 News:

- <http://typo3.org/news>

Release Infos:

- http://wiki.typo3.org/TYPO3\CMS_8.6.0
- [INSTALL.md](#) and [Changelog](#)
- `typo3/sysex/core/Documentation/Changelog/8.6/*`

TYPO3 Bug-/Issuetracker:

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

TYPO3 and Fluid Git Repositories:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://github.com/TYPO3/Fluid>

Sources and Authors

TYPO3 CMS What's New Team:

Pierrick Caillon, Sergio Catala, Richard Haeser, Jigal van Hemert,
Patrick Lobacher, Michel Mix, Sinisa Mitrovic,
Nena Jelena Radovic, Michael Schams and Roberto Torresani

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

Licensed under Creative Commons BY-NC-SA 3.0

