

# **TYP03 Version 10.0 – What's New**

Summary of the new features, changes and improvements

Created by:  
Michael Schams

# TYPO3 Version 10.0 - What's New

---

## Chapter Overview

Introduction

Backend User Interface

Changes for Integrators and Developers

Changes for Integrators

Changes for Developers

Sources and Authors

# Introduction

## The Facts

# Introduction

---

## TYP03 Version 10.0 - The Facts

- Release date: 23 July 2019
- Release type: Sprint Release
- Development time: approx. 6 months



# Introduction

---

## Executive Summary

TYPO3 version 10.0 is the first sprint release on the way to the LTS-version (long-term support) in 2020.

Since the main focus of version 10.0 is on clean-up tasks, it is not surprising that a large number of breaking changes have been introduced in this release.

This approach allows us to introduce new libraries, modern concepts and streamline APIs at an early stage of the development to make sure that TYPO3 remains one of the best enterprise content management systems on the market.

A number of exciting [initiatives](#) have also been formed to make long-term improvements in a specific area of TYPO3.

# Introduction

---

## System Requirements

- PHP version 7.2 or 7.3
- PHP settings:
  - `memory_limit`  $\geq$  256M
  - `max_execution_time`  $\geq$  240s
  - `max_input_vars`  $\geq$  1500
  - compilation option `--disable-ipv6` must not be used
- Most database servers supported by **Doctrine DBAL** also work with TYPO3. Tested DB engines are for example:

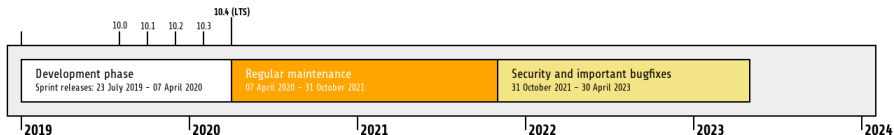


# Introduction

---

## Development, Release and Maintenance Timeline

### TYPO3 v10



### Extended Support

The [TYPO3 GmbH](#) offers further support options for TYPO3 v10 LTS even after 30 April 2023 for up to two additional years.

# Introduction

---

## TYP03 v10 Roadmap

Release dates and their primary focus:

- v10.0 23/July/2019 Pave the way for exciting new concepts and APIs
- v10.1 01/Oct/2019 Routing Improvements and Site Handling v2
- v10.2 03/Dec/2019 Fluid/Rendering Engine Improvements
- v10.3 04/Feb/2020 Feature Freeze
- v10.4 07/Apr/2020 LTS Release (Long-term Support)

<https://typo3.org/article/typo3-v10-roadmap/>

<https://typo3.org/article/typo3-v10-safe-and-sound/>



# 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/10.0
$ tar xzf typo3_src-10.0.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-10.0.0 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 and higher

# Introduction

---

## Installation Using `composer`

- Installation using `composer` under Linux, Mac OS X and Windows 10:

```
$ cd /var/www/site/  
$ composer create-project typo3/cms-base-distribution typo3v10 ^10
```

- Alternatively, create your custom `composer.json` file and run:

```
$ composer install
```

Further details and examples for `composer.json` files are available at:

<https://composer.typo3.org>

## Chapter 1: Backend User Interface

# Backend User Interface

---

## Filelist Sorting

Files can now be sorted by their meta data title in the "File Links" content element.

The screenshot displays the configuration interface for the "File Links" content element. A dropdown menu is open, showing sorting options: "no sorting", "by file extension [extension]", "by filename [name]", "by file type [type]", "by file size [size]", "by file creation date [creation\_date]", "by file modification date [modification\_date]", "by file metadata title [title]" (highlighted), and "by file metadata title [title]" (selected). Below the dropdown, the "Sorting direction" field is set to "Ascending" and the "Target" field is set to "None".

**Filelinks Layout**

**Display File Size Information** [filelink\_size]  [0]

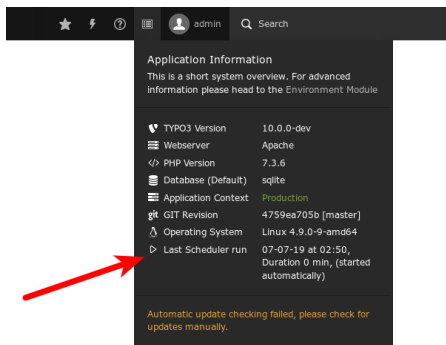
**Display description** [uploads\_description]  [0]

**Display file/Icon/thumbnall** [uploads\_type]

# Backend User Interface

## System Information Toolbar

The system information toolbar now shows information about the TYPO3 scheduler.



The screenshot shows the System Information Toolbar in the TYPO3 Backend User Interface. The toolbar is dark grey and contains several icons: a star, a lightning bolt, a question mark, a list icon, a user profile icon labeled 'admin', and a search icon labeled 'Search'. Below the toolbar, the 'Application Information' section is displayed. It contains a short system overview and a list of system details. A red arrow points to the 'Last Scheduler run' entry.

Application Information	
This is a short system overview. For advanced information please head to the Environment Module	
TYPO3 Version	10.0.0-dev
WebsERVER	Apache
PHP Version	7.3.6
Database (Default)	sqlite
Application Context	Production
git GIT Revision	4759ea705b [master]
Operating System	Linux 4.9.0-9-amd64
Last Scheduler run	07-07-19 at 02:50, Duration 0 min, (started automatically)

Automatic update checking failed, please check for updates manually.

# Backend User Interface

---

## Link Handler

A new link handler has been added that lets backend users set links to phone numbers using the `tel:` protocol.

Link Browser ✕

Page File Folder External URL Email Telephone

**Title**

**Telephone number**

```
<div class="ce-bodytext">
  <p>
    Give the TYPO3 GmbH a
    <a href="tel:+492112054360" title="Call the TYPO3 GmbH">call</a>
    if you have any further questions.
  </p>
</div>
```

# Backend User Interface

---

## SEO Sitemap (1)

EXT: seo now supports change frequencies and priorities for the Sitemap. Page properties (tab "SEO") feature two new fields.

Sitemap

**Change frequency** [sitemap\_changefreq] **Priority** [sitemap\_priority]

None

None

Always [always]

Hourly [hourly]

Daily [daily]

Weekly [weekly]

Monthly [monthly]

Yearly [yearly]

Never [never]

0.5 [0.5]

# Backend User Interface

---

## SEO Sitemap (2)

These settings can also be defined in TypoScript, mapped to fields in the database.

```
plugin.tx_seo {
    config {
        xmlSitemap {
            sitemaps {
                <unique key> {
                    provider = TYPO3\CMS\Seo\XmlSitemap\RecordsXmlSitemapDataProvider
                    config {
                        ...
                        changeFreqField = news_changefreq
                        priorityField = news_priority
                        ...
                    }
                }
            }
        }
    }
}
```

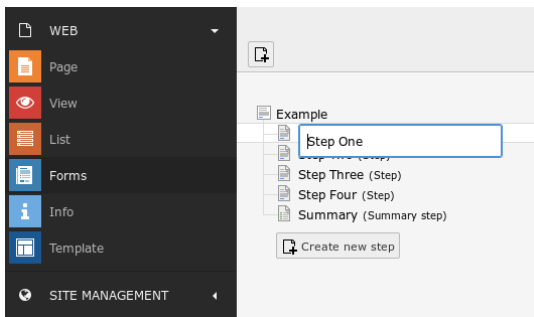


# Backend User Interface

---

## Forms

Labels of form elements can now be edited by double clicking on the title in the structure tree.



## Chapter 2: Changes for Integrators and Developers

# Changes for Integrators and Developers

---

## Page Not Found Handling

- The following global TYPO3 settings have been removed:
  - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling']`
  - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling_statheader']`
  - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling_accessdeniedheader']`
  - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageUnavailable_handling']`
  - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageUnavailable_handling_statheader']`
- The Site Handling introduced in TYPO3 v9 replaces these settings.

# Changes for Integrators and Developers

---

## Page Not Found Handling

- The message "*Page is being generated*" and the corresponding temporary HTTP 503 response have been removed.

**Page is being generated.**

If this message does not disappear within 30 seconds, please reload.

- Instead of offloading the work to wait for the final page content, concurrent requests now wait for the real page content to be rendered.

# Changes for Integrators and Developers

---

## Fluid

- The removal of white spaces in the default layout file of `EXT:fluid_styled_content` caused occasional issues and has been removed.

# Changes for Integrators and Developers

---

## Form Framework: Multiple Recipients (1)

- Mails sent by the *EmailFinisher* can now have multiple recipients.
- The following new options have been introduced:
  - `recipients` (To)
  - `replyToRecipients` (Reply-To)
  - `carbonCopyRecipients` (CC)
  - `blindCarbonCopyRecipients` (BCC)

# Changes for Integrators and Developers

---

## Form Framework: Multiple Recipients (2)

- This change requires a manual migration of single value options to their list value successors.

**Old** Finisher configuration:

```
finishers:  
-  
  identifier: EmailToReceiver  
  options:  
    recipientAddress: user@example.com  
    recipientName: 'Firstname Lastname'
```

**New** Finisher configuration:

```
finishers:  
-  
  identifier: EmailToReceiver  
  options:  
    recipients:  
      -  
      -  
      -  
      -
```

# Changes for Integrators and Developers

---

## Form Framework: Plaintext/HTML

- Mails sent by the *EmailFinisher* can now feature both plaintext and/or HTML part.
- At the same time the option `format` has been marked deprecated and will be removed in TYPO3 v11.
- Existing values will be automatically migrated:
  - `format:html` → `addHtmlPart:true`
  - `format:plaintext` → `addHtmlPart:false`
  - a missing "format" → `addHtmlPart:true`
- Developers should be aware of the following two constants which have been marked deprecated:
  - `EmailFinisher::FORMAT_PLAINTEXT`
  - `EmailFinisher::FORMAT_HTML`



# Changes for Integrators and Developers

---

## Frontend Login: Extbase

- The frontend user login (`EXT:felogin`) will be converted to Extbase and Fluid.
- This is a long-term goal, currently work-in-progress.  
See [forge #84262](#) for further details.
- The following changes have been implemented in TYPO3 v10.0:
  - ➊ Prefix "l1\_" has been removed from locallang keys.
    - Update your TypoScript if you have overwritten language labels and remove the prefix "l1\_" from your keys.
  - ➋ Existing FlexForm structure has been reworked.
    - Execute the Upgrade Wizard to migrate the FlexForm values.

# Changes for Integrators and Developers

---

## Languages

- ISO Codes:
  - The unused database field `static_lang_isocode` has been removed.
  - `EXT:static_info_tables` can be installed to reimplement functionality if required.
  - Developers are advised to fetch all metadata for a language using the Site Configuration and the SiteLanguage API instead.
- Language Files:
  - Usage of the global array `$GLOBALS[LOCAL_LANG]` has been deprecated.
  - The 2nd and 3rd arguments of `LanguageService->includeLLFile()` have been deprecated.

# Changes for Integrators and Developers

---

## Miscellaneous

- The IP locking functionality has been extended to support IPv6 too (frontend and backend).

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['lockIPv6'] = 2;  
$GLOBALS['TYPO3_CONF_VARS']['BE']['lockIPv6'] = 2;
```

- The public property `lockIP` in following PHP class has been removed: `\TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`.
- Migration options:
  - Set `lockIP` and `lockIPv6` in `$GLOBALS['TYPO3_CONF_VARS'] [...]`.
  - Use the new IP-Locker API: `\TYPO3\CMS\Core\Authentication\IpLocker`.

## Chapter 3: Changes for Integrators

# Changes for Integrators

---

## Breaking Changes

Integrators be advised: In TYPO3 v9, some PHP code, TSconfig, TypoScript options and conditions, as well as scheduler tasks were marked deprecated.

In accordance to TYPO3's **deprecation policy**, these components have changed or have been removed in TYPO3 v10.0.




Enable the deprecation log, carefully test your code and review the log to locate possible issues. Use the built-in [Extension Scanner](#) to get a full report of extension incompatibilities.

# Changes for Integrators

---

## Backend User Switch

- A log message is written if an admin user switches to another backend user:

E	Time	User	Type	Action	Details	Actions
	04:11:52	 fred (via admin ) LIVE	LOGIN	LOGOUT	User fred logged out from TYPO3 Backend (msg#255.2.1)	
	04:11:48	 admin LIVE	LOGIN	LOGOUT	User admin switched to user fred (be_users:2) (msg#255.2.1)	
	04:08:38	 admin [-99]	LOGIN	LOGIN	User admin logged in from 172.72.72.1 (msg#255.1.1)	



# Changes for Integrators

---

## TypoScript Changes

- TypoScript property `config.cache` now supports keyword "current" to refer to the current page. For example:

```
config.cache.all = fe_users:current
```

- The Page/User TSconfig setting `TSFE.constants` has been removed.
  - Include TypoScript conditions in `setup/constants` and use a proper configuration in file `ext_localconf.php`.
- The following two options to configure the size of popup windows have been removed:
  - `options.popupWindowSize`
  - `options.rte.popupWindowSize`

# Changes for Integrators

---

## TypoScript Changes

- The database field `nextLevel` of the database table `sys_template` has been removed.
  - Replace the record (the UID is stored in the field `nextLevel`) with a condition to add TypoScript for subpages. For example: `[tree.level > 1]`
- The following values are **not allowed** anymore:
  - `typolink.addQueryString.method = POST`
  - `typolink.addQueryString.method = GET,POST`
  - `typolink.addQueryString.method = POST,GET`
  - Change the assignments in TypoScript, Fluid and PHP to GET.



# Changes for Integrators

---

## Caches

- Caching framework does not support the ApcBackend anymore
  - Use **APCu** instead - note the "u".

OLD:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['rootline']['backend'] =  
\TYPO3\CMS\Core\Cache\Backend\ApcBackend::class;
```

NOW:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['rootline']['backend'] =  
\TYPO3\CMS\Core\Cache\Backend\ApcuBackend::class;
```

- Extbase caches `extbase_reflection` and `extbase_datamapfactory_datamap` have been consolidated and are now available as a single cache named "extbase".

# Changes for Integrators

---

## Form Framework

- The following option has been renamed:  
translationFile → translationFiles
- The default translation files are now registered at index 10:
  - EXT:form/Resources/Private/Language/locallang.xlf
  - EXT:form/Resources/Private/Language/Database.xlf
- Custom form YAML configuration files need to be updated.

OLD:  
translationFile: path/to/locallang.xlf

NEW:  
translationFiles:  
  20: path/to/locallang.xlf

# Changes for Integrators

---

## Cache Storage Type (1)

- TYPO3 features a flexible caching system with a default configuration that is ideal for most use cases.
- The storage type can now be configured to fine-tune the caches and increase performance depending on the individual environment.
  - Choose the **database** storage for a standard environment or if a network file system (NFS) is used for example.
  - Choose the **file system** if a distributed database setup is used for example.
  - Choose **custom cache settings** to configure the storage type for each cache independently.
- For more complex installations, memory-based caches such as [Redis](#) or [Memcached](#) should be considered.

# Changes for Integrators

## Cache Storage Type (2)

### ■ Backend: MAINTENANCE → Settings → Cache:

Cache settings

TYPO3 features a flexible caching system with a default configuration that is ideal for most use cases. Depending on your individual hosting setup, the performance of your TYPO3 instance can be optimized even further by adjusting the storage type below.

Storing the caches in the database is usually the best choice for standard environments as well as for installations where a network file system (NFS) is used and file system access impacts the performance and should be avoided (high disk I/O). The file-based storage for TYPO3 caches is more efficient in shared hosting environments or if a distributed database is used. For more complex, high-end installations, memory-based caches such as Redis or Memcached should be considered which require additional configuration in "LocalConfiguration.php".

To configure the storage type for each cache independently, custom cache settings can be configured.

Prefer database storage for caching [Active]

Prefer File storage for caching

Custom configuration

Custom cache settings:

<code>SYS/caching/cacheConfigurations/hash/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYS/caching/cacheConfigurations/pages/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYS/caching/cacheConfigurations/pagesection/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYS/caching/cacheConfigurations/imagesizes/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYS/caching/cacheConfigurations/rootline/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>

# Changes for Integrators

---

## Task Center and `EXT:sys_action`

- The system extensions `EXT:taskcenter` and `EXT:sys_action` have been removed from the core.
- They are now available as separate extensions from the [TER](#) and at [GitHub](#).
- Keep an eye on the [Dashboard Initiative](#) for a new and much better approach.

# Changes for Integrators

---

## Miscellaneous (1)

- Type of Twitter Card can be selected/configured now. This option renders the meta tag `twitter:card` in the frontend.

```
page {
  meta {
    twitter:card = summary_large_image
    twitter:card.replace = 1
  }
}
```

- Only parameters that are needed to calculate the cHash are included in canonicalized URLs by default. Additional query parameters can now be configured:

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['additionalCanonicalizedUrlParameters'].
```

Note: only add parameters which change the content of your page. Otherwise search

# Changes for Integrators

---

## Miscellaneous (2)

- When importing XML data using `EXT:impexp`, the File Deny Pattern applies now and rejects embedded PHP files for example.
- RTE image handling functionality has been removed completely. For image support in CKEditor, consider to use `EXT:rte_ckeditor_image` for example.
- A property within workspaces for *unpublishing* records has been removed in v10 (including the database field `sys_workspace.unpublish_time`). This feature was disabled in TYPO3 v4.5 and not used or provided by the TYPO3 core.

# Changes for Integrators

---

## Miscellaneous (3)

- When rendering HTML5 output, `<script>` tags do not include the attribute `type="text/javascript"` anymore.
- This can be re-enabled for the frontend by using TypeScript if required:

```
page {
  includeJS {
    myfile = EXT:example/Resources/Public/JavaScript/myfile.js
    myfile.type = text/javascript
  }
}
```

- The following deprecated system extensions have been removed:
  - EXT:rsaauth
  - EXT:fe\_edit



## Chapter 4: Changes for Developers

# Changes for Developers

---

## Breaking Changes

Developers be advised: In TYPO3 v9, some PHP classes, interfaces, class aliases, properties, methods, constants, global options and variables, etc. were marked deprecated.

In accordance to TYPO3's **deprecation policy**, these components have been removed in TYPO3 v10.0.

This also includes some hooks, PHP annotations (such as `@inject` and `@validate`), as well as some changed visibilities (e.g. from "public" to "protected").

Enable the deprecation log, carefully test your code and review the log to locate possible issues. Use the built-in [Extension Scanner](#) to get a full report of extension incompatibilities.

# Changes for Developers

---

## New Mail API

- SwiftMailer has been superseded by more modern libraries:
  - `symfony/mime` for creating mail messages
  - `symfony/mailer` for sending emails
- PHP function `mail()` is no longer supported.
  - It is recommended to switch to `sendmail` or `smtp` instead.
- Custom SwiftMailer plugins or transports require a migration.
- See the [Symfony Documentation](#) for further details how to leverage the new Mail API capabilities.

# Changes for Developers

---

## Symfony Dependency Management/Injection (1)

- The package `symfony/dependency-injection` has been integrated and is used to manage system-wide dependency management and dependency injection for classes.
- This approach aims to replace the Extbase dependency injection container and object manager.
- Therefore, classes should be adjusted and avoid (whenever possible):
  - `\TYPO3\CMS\Extbase\Object\ObjectManager`
  - `\TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance()`

# Changes for Developers

---

## Symfony Dependency Management/Injection (2)

- Configuration options include:
  - Autowiring (see example below)
  - Manual wiring (see [change log](#))
  - Advanced functionality (see [change log](#))

```
# Configuration/Services.yaml
services:
  _defaults:
    autowire: true
    autoconfigure: true
    public: false

  Your\Namespace\:
    resource: '../Classes/*'
```

- See [Symfony documentation](#) for further details.

# Changes for Developers

---

## Event Dispatching (1)

- A new "EventDispatcher" system has been added which aims to replace the hooks and Signal/Slots concepts.
- It is based on the [PSR-14 standard](#) which allows developers to inject logic into an application easily and consistently.
- PSR-14 consists of the following four components:
  - An **EventDispatcher** object that is used to trigger an event.
  - A **ListenerProvider** object that contains registered listeners for all events.
  - One or multiple **Event** objects which are called from the TYPO3 core or extensions ("Emitter").
  - One or multiple **Listeners** (usually in extensions and PHP packages) that are registered.

# Changes for Developers

---

## Event Dispatching (2)

### Implementation example

- 1 Add `event.listener` tag to the file `Configuration/Services.yaml`:

```
services:
  Vendor\Example\EventListener\NullMailer:
    tags:
      - { name: event.listener, identifier: 'myListener', event: TYPO3\CMS\Core\Mail\Event\
        AfterMailerInitializationEvent, before: 'redirects, anotherIdentifier' }
```

- 2 Implement your event object:

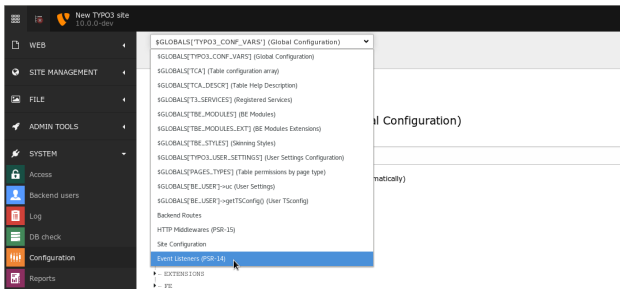
```
namespace Vendor\Example\EventListener;

class NullMailer
{
    public function __invoke(AfterMailerInitializationEvent $event): void
    {
        $event->getMailer()->injectMailSettings(['transport' => 'null']);
    }
}
```

# Changes for Developers

## Event Dispatching (3)

- List of available Event Listeners can be accessed in the backend:  
(requires system extension EXT:lowlevel)





# Changes for Developers

---

## Event Dispatching (4)

- Best practices:
  - Add only one Listener per PHP class and use `__invoke()` as the method name.
  - Add "Event" suffix to the class name when creating a new Event PHP class.
  - Move the Event PHP class file to an appropriate folder e.g. `Classes/Database/Event`.
  - Use dependency injection in form of a constructor argument to receive the `EventDispatcher` object if possible.
- Additional note:

Events provided by the TYPO3 core follow TYPO3's deprecation policy, except for its constructor arguments which may vary.

# Changes for Developers

---

## PSR-3 Logging Interface

- TYPO3's Logging Framework (in particular LogLevel and LogManager) now uses the [PSR-3 Logger Interface](#).
- PSR-3 is a standardized method that allows libraries to receive a `Psr\Log\LoggerInterface` object and to write logs to it in a simple and universal way.
- This lets developers use custom loggers and to interact with other logging systems.

# Changes for Developers

---

## JavaScript Options and Functions (1)

- The following JavaScript files have been removed:
  - `jsfunc.inline.js`
  - `jsfunc.evalfield.js`
    - Use `TYPO3\CMS\Backend\FormEngineValidation` instead.
- Additional submit handlers could previously be added by the option `additionalJavaScriptSubmit`. This option has been removed.
  - Create and register an AMD module instead.
- The global JavaScript function `top.openUrlInWindow()` has been marked deprecated.

# Changes for Developers

---

## JavaScript Options and Functions (2)

- The global object `TBE_EDITOR.typo3form` and its backward layers `typo3FormFieldSet` and `typo3FormFieldGet` have been removed.
- File `md5.js` has been marked deprecated.
  - Load the AMD module `TYPO3\CMS\Backend\Hashing\Md5` via RequireJS instead.
- The following global JavaScript functions have been marked deprecated:
  - `top.rawurlencode()`
  - `top.str_replace()`
- Module `TYPO3\CMS\Backend\SplitButtons` has been deprecated.
  - Use `TYPO3\CMS\Backend\DocumentSaveActions` instead.

# Changes for Developers

---

## UTF-8-based Domains

- PHP has native functions to convert domains from UTF-8 into IDNA ASCII form (“punicode”), for example `idn_to_ascii()`.
- These can be used directly if the PHP extension “`intl`” is installed.
- If the PHP extension is not installed, the package `symfony/polyfill-intl-idn` provides the functions now.
- Previously, the package `algo26-matthias/idna-convert` was used which has been removed now.

# Changes for Developers

---

## BitSet Class

- New class has been introduced to efficiently handle boolean flags:  
TYPO3\CMS\Core\Type\BitSet
- For example:

```
define('PERMISSIONS_NONE', 0b0); // 0
define('PERMISSIONS_PAGE_SHOW', 0b1); // 1
define('PERMISSIONS_PAGE_EDIT', 0b10); // 2
define('PERMISSIONS_PAGE_DELETE', 0b100); // 4
define('PERMISSIONS_PAGE_NEW', 0b1000); // 8
define('PERMISSIONS_CONTENT_EDIT', 0b10000); // 16
define('PERMISSIONS_ALL', 0b11111); // 31

$bitSet = new \TYPO3\CMS\Core\Type\BitSet(PERMISSIONS_PAGE_SHOW | PERMISSIONS_PAGE_NEW);
$bitSet->get(PERMISSIONS_PAGE_SHOW); // true
$bitSet->get(PERMISSIONS_CONTENT_EDIT); // false
```

# Changes for Developers

---

## Request Handler (1)

- The following internal interface has been removed in favor of PSR-15 request handler and middleware interfaces:

`TYPO3\CMS\Core\Http\RequestHandlerInterface`

# Changes for Developers

---

## Request Handler (2)

- The configuration of Extbase request handlers is no longer possible with TYPOScript.

**Old** method in TYPOScript:

```
config.tx_extbase {
    mvc {
        requestHandlers {
            Vendor\Example\Mvc\Web\FrontendRequestHandler = Vendor\Example\Mvc\Web\
                FrontendRequestHandler
        }
    }
}
```

**New** method in file Configuration/Extbase/RequestHandlers.php:

```
<?php
declare(strict_types = 1);

return [
    \Vendor\Example\Mvc\Web\FrontendRequestHandler::class,
];
```



# Changes for Developers

---

## Caching Framework

- The following caches have been renamed:
  - `cache_core` → `core`
  - `cache_hash` → `hash`
  - `cache_pages` → `pages`
  - `cache_pagesection` → `pagesection`
  - `cache_runtime` → `runtime`
  - `cache_rootline` → `rootline`
  - `cache_imagesizes` → `imagesizes`
- New method to access the caches:

OLD:

```
$cacheManager->getCache('cache_core');
```

NEW:

```
$cacheManager->getCache('core');
```

- The prefix `cf_` has been removed from the database tables.

# Changes for Developers

---

## Extbase and Fluid (1)

- Registering plugins/modules require fully-qualified class names now
  - `\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin()`
  - `\TYPO3\CMS\Extbase\Utility\ExtensionUtility::registerModule()`
- Also omit vendor name in the extension name (first argument).
  - Use "ExampleBlog" instead of "Vendor.ExampleBlog".
- For example:

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin(  
    'ExampleBlog', // previously: 'Vendor.ExampleBlog'  
    'pi1',  
    [  
        \Vendor\Example\Controller\BlogController::class => 'list,update,delete'  
    ],  
    [  
        \Vendor\Example\Controller\BlogController::class => 'list,update,delete'  
    ]  
);
```

# Changes for Developers

---

## Extbase and Fluid (2)

- Property `extensionName` of `AbstractController` has been removed.
  - Use `\TYPO3\CMS\Extbase\Mvc\Request::getControllerExtensionName()` instead.

# Changes for Developers

---

## Extbase and Fluid (3)

- Extbase models now support non fully-qualified class names in DocBlocks.

```
use TYPO3\CMS\Extbase\Persistence\ObjectStorage;
use ExtbaseTeam\BlogExample\Domain\Model\Comment;

class Post
{
    /**
     * @var ObjectStorage<Comment>
     */
    public $comments;
}
```

# Changes for Developers

---

## Extbase and Fluid (4)

- Validators are not registered automatically in Extbase anymore.
- For a model named `Vendor\Example\Domain\Model\Blog`, Extbase automatically used the validator `Vendor\Example\Domain\Validator\BlogValidator`
- Validators need to be registered manually now:

```
use Vendor\Example\Domain\Model\Blog;
use TYPO3\CMS\Extbase\Annotation as Extbase;
use TYPO3\CMS\Extbase\Mvc\Controller\ActionController;

class BlogController extends ActionController
{
    /**
     * @Extbase\Validate(param="blog", validator="Vendor\Example\Domain\Validator\BlogValidator")
     */
    public function showAction(Blog $blog)
    {
        // ...
    }
}
```

# Changes for Developers

---

## Extbase and Fluid (5) - Class Mapping

- Persistence related class mapping using TypoScript is no longer supported:

```
config.tx_example_blog {
    persistence {
        classes {
            Vendor\Example\Domain\Model\Author {
                mapping {
                    tableName = fe_users
                    columns.name.mapOnProperty = fullname
                }
            }
        }
    }
}
```

# Changes for Developers

---

## Extbase and Fluid (6) - Class Mapping

- The mapping needs to be implemented in a PHP file  
Configuration/Extbase/Persistence/Classes.php:

```
<?php
declare(strict_types = 1);

return [
    \Vendor\Example\Domain\Model\Author::class => [
        'tableName' => 'fe_users',
        'properties' => [
            'fullname' => [
                'fieldName' => 'name'
            ]
        ]
    ]
];
```

- Note that property name and DB field have swapped!  
Previously: <db-field>.mapOnProperty = <property>  
New: properties.<property>.fieldname = <db-field>

# Changes for Developers

---

## Extbase and Fluid (7)

- Class files now feature the "strict types" mode and type hints for scalars

```
<?php  
declare(strict_types=1);
```

- This results in fatal PHP errors if the method signatures in custom extensions are not compatible with the interfaces and/or parent classes.
- See [forge #87594](#) for a complete list of files and their changes.
- This task is still work in progress and further changes will be made.



# Changes for Developers

---

## TCA Changes

- The following TCA options have been removed:
  - `$TCA[$tableName]['ctrl']['selicon_field_path']`
  - `$TCA[$tableName]['ctrl']['setDefaultOnCopy']`
    - When copying records, a `DataHandler` should be used to reset fields.
- The entire TCA of `sys_history` has been removed and the database field `pid` dropped. Accessing `$GLOBALS['TCA']['sys_history']` now triggers a PHP warning.

# Changes for Developers

---

## User Authentication Classes/Services (1)

- The following abstract class has been restructured:  
`TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`
- This also includes the following two direct sub-classes:
  - `BackendUserAuthentication`
  - `FrontendUserAuthentication`
- This change affects the properties:
  - `sessionTimeout`
  - `gc_time`
  - `sessionDataLifetime`
  - `loginType`

# Changes for Developers

---

## User Authentication Classes/Services (2)

- The following class does not inherit from `AbstractService` anymore:  
`\TYPO3\CMS\Core\Authentication\AbstractAuthenticationService`
- This possibly affects some of the hooks and custom authentication providers available.
- Developers are advised to review their custom authentication services and update their code if required.

# Changes for Developers

---

## Filelist Controllers

- The following controllers have been moved to `EXT:filelist`:
  - `CreateFolderController`
  - `EditFileController`
  - `FileUploadController`
  - `RenameFileController`
  - `ReplaceFileController`
- As a result, their namespace changed to `\TYPO3\CMS\Filelist\Controller\File`

Note: Use the TYPO3 FAL as API and add your own functionality with your own controller instead of reusing the **internal** controllers listed above.

# Changes for Developers

---

## Frontend Preview URL

- The following static method has been marked deprecated:  
`\TYPO3\CMS\Backend\Utility\BackendUtility::getViewDomain()`
- Substitute the method by directly detecting a site based on a given page ID in the TYPO3 backend.
- For example:

```
$pageId = 123;  
$site = GeneralUtility::makeInstance(SiteFinder::class)->getSiteByPageId($pageId);  
$url = $site->getRouter()->generateUri($pageId, ['type' => 13]);
```

# Changes for Developers

---

## cHash in UriBuilder and ViewHelpers

- The following two Extbase UriBuilder methods have been deprecated:
  - `UriBuilder->setUseCacheHash()`
  - `UriBuilder->getUseCacheHash()`
- This also impacts a number of Fluid ViewHelpers:
  - `f:form`
  - `f:uri.page`
  - `f:link.action`
  - `f:uri.typolink`
  - `f:link.page`
  - `f:widget.link`
  - `f:link.typolink`
  - `f:widget.uri`
  - `f:uri.action`
- ...as well as the TypoLink option "useCacheHash".

# Changes for Developers

---

## Frontend Request Workflow (1)

- The Frontend Request Workflow has been reworked significantly.
- Components involved are all built using PSR-15 middlewares, the PSR-15 Request Handler, and the global TypoScriptFrontendController (TSFE) since TYPO3 v9.
- This impacts custom code, if the following hook and a frontend session is used:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tslib/class.tslib_fe.php']['hook_eofe']
```

- Use a PSR-15 middleware instead of a hook, or explicitly call `storeSessionData` within the hook in PHP.

# Changes for Developers

---

## Frontend Request Workflow (2)

- The following global variables have been removed:
  - `$GLOBALS['TYPO3_MISC']['microtime_start']`
  - `$GLOBALS['TYPO3_MISC']['microtime_end']`
  - `$GLOBALS['TYPO3_MISC']['microtime_BE_USER_start']`
  - `$GLOBALS['TYPO3_MISC']['microtime_BE_USER_end']`
- The TYPO3 core used them in the Admin Panel and HTTP header for example.
  - Use `TimeTracker->finish()` instead.



# Changes for Developers

---

## Locales (1)

- Method `Locales::initialize()` has been marked deprecated.
  - Use `GeneralUtility::makeInstance(Locales::class)` or dependency injection to fetch an instance of the class `Locales` instead.
- Functionality of the following method has been marked deprecated:  
`TypoScriptFrontendController->settingLocale()`.
  - Function is now available as `Locales::setSystemLocaleFromSiteLanguage()`.

# Changes for Developers

---

## Locales (2)

- Public property

TypoScriptFrontendController->sys\_language\_isocode has been marked deprecated.

- ▶ Access the property via SiteLanguage->getTwoLetterIsoCode() and `sitelanguage:twoLetterIsoCode` instead.

# Changes for Developers

---

## Frontend Track User

- This following public properties of class `\TYPO3\CMS\Core\Authentication\AbstractUserAuthentication` have been removed:
  - `AbstractUserAuthentication->get_name`
  - `AbstractUserAuthentication->getFallback`
  - `AbstractUserAuthentication->getMethodEnabled`
  - `AbstractUserAuthentication->get_URL_ID`
- Also the property `getMethodUrlIdToken` of class `\TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController`.
- And the TypoScript setting `config.ftu`, as well as the global configuration `$GLOBALS['TYPO3_CONF_VARS']['FE']['get_url_id_token']`.

# Changes for Developers

---

## Constructor Injection in DataMapper

- The following class now uses constructor injection rather than setter injection: `\TYPO3\CMS\Extbase\Persistence\Generic\Mapper\DataMapper`
  - Avoid `GeneralUtility::makeInstance()` and `ObjectManager->get()`.
  - Use dependency injection instead (preferably constructor injection).

# Changes for Developers

---

## Context API (1)

- The Context API features a new aspect "frontend.preview" that can be used to determine if the frontend is in preview mode:

```
GeneralUtility::makeInstance(Context::class)
->getPropertyFromAspect('frontend.preview', 'isPreview');
```

- This aspect replaces the following property which is marked deprecated now: `TypoScriptFrontendController->fePreview`

# Changes for Developers

---

## Context API (2)

- Another new aspect `TypoScriptAspect` can be used to manipulate/check if `TemplateRendering` is forced.
- The setting `forceTemplateParsing` (TSFE and `TemplateService`) has been deprecated. The Context API should be used instead:

```
GeneralUtility::makeInstance(Context::class)
    ->getPropertyFromAspect('typoscript', 'forcedTemplateParsing');


```

# Changes for Developers

---

## Miscellaneous (1)

- Directive `createDirs` in file `ext_emconf.php` not supported anymore.
  - Folders will not be created automatically during extension installation.
- The following two properties in class `TYPO3\CMS\Extbase\Mvc\Controller\ActionController` have been removed:
  - `$namespacesViewObjectNamePattern`
  - `$viewFormatToObjectNameMap`
- The following existing hook has been extended and can now also be used to validate passwords:  
`$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['felogin']['password_changed']`

# Changes for Developers

---

## Miscellaneous (2)

- The following method has been marked deprecated:

`\TYPO3\CMS\Extbase\Utility\TypeHandlingUtility::hex2bin()`

- Use the native PHP function `hex2bin()` instead.

- The following methods of class

`\TYPO3\CMS\Core\Utility\VersionNumberUtility`

have been marked deprecated:

- `convertIntegerToVersionNumber()`
- `splitVersionRange()`
- `raiseVersionNumber()`

- Implement the methods as custom code.



# Changes for Developers

---

## Miscellaneous (3)

- It is now possible to get the default value of a class property when using the ReflectionService.

```
$property = GeneralUtility::makeInstance(ReflectionService::class)
->getClassSchema(MyClass::class)
->getProperty('myProperty');
```

- Backend routes to modules without path configurations are now named `"/module/<main-module>/<sub-module>"` by default (for example: `"/module/web/ts"`).
- Old routes still work (e.g. `"/web/ts/"`) but this syntax will be removed in TYPO3 v11.

# Changes for Developers

---

## Miscellaneous (4)

- The FormEngine DataProvider parentPageTca has been removed.
  - Developers can access `$GLOBALS['TCA']['pages']` directly, instead of `$result['parentPageTca']`.
- The following database fields have been removed:
  - `tt_content.spaceBefore` (replaced by field `space_before_class`)
  - `tt_content.spaceAfter` (replaced by field `space_after_class`)
  - `pages.t3ver_id` (unused since TYPO3 v9)
- The PHP class `\TYPO3\CMS\Frontend\Page\PageRepository` has been moved from the "frontend" system extension into the core.
  - Replace with class: `\TYPO3\CMS\Core\Domain\Repository\PageRepository`

# Changes for Developers

---

## Miscellaneous (5)

- 4th parameter of method `PageRepository->enableFields()` has been removed.
  - If developers use a 4th parameter in this method call, which is set to **"false"**, this can be removed safely.
  - If it is set to **"true"**, the code needs to be replaced with a separate instance of `PageRepository` with a custom `Context`.
- The internal method `File::_getMetaData()`, which is used to fetch meta data of a file, has been marked deprecated.
  - Use `$fileObject->getMetaData()->get()` to fetch the meta data instead.
- The route identifier `"xMOD_tximpexp"` has been marked deprecated.
  - Use `tx_impexp_export` or `tx_impexp_import` depending on the use case.

# Changes for Developers

---

## Miscellaneous (6)

- The following abstract method has been removed:  
`\TYPO3\CMS\Extbase\Configuration\AbstractConfigurationManager::getSwitchableControllerActions()`
  - Use the new method name `getControllerConfiguration()` instead (same PHP class).
- The global variable `$TBE_TEMPLATE` has been removed, including the related PSR-15 middleware (which was marked as internal).
  - Instantiate the `DocumentTemplate` class directly in the controller of the module.
  - Migrate to `ModuleTemplate` which is available since TYPO3 v7.
- The global variable `$GLOBALS['T3_VAR']` has been removed.

## Chapter 5: Sources and Authors

# Sources and Authors

---

## Sources

### TYPO3 News:

- <https://typo3.org/project/news/>

### Release Infos:

- [https://get.typo3.org/release-notes/10.x/TYPO3\CMS\\_10.0.0](https://get.typo3.org/release-notes/10.x/TYPO3\CMS_10.0.0)
- [TYPO3 v10 Changelog](#)
- `typo3/sysexst/core/Documentation/Changelog/10.0/*`

### 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:

Richard Haeser, Jigal van Hemert, Henrietta Kucsovan, Andreas Fießer  
Sinisa Mitrovic, Michael Schams and Roberto Torresani

<https://typo3.org/help/documentation/whats-new/>

Licensed under Creative Commons BY-NC-SA 3.0

