TYPO3 CMS 7.4 – What’s New

Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

03/August/2015

TYPO3

TYPO3 CMS 7.4 - What's New

Chapter Overview

Introduction

Backend User Interface

TSconfig & TypoScript

TSconfig & TypoScript: Data Processors

In-Depth Changes

Extbase & Fluid

Deprecated/Removed Functions

Sources and Authors

Introduction

The Facts

Introduction

TYPO3 CMS 7.4 - The Facts

- Release date: 04. August 2015
- Release type: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Primary focus: Backend Overhaul Vol 2

Introduction

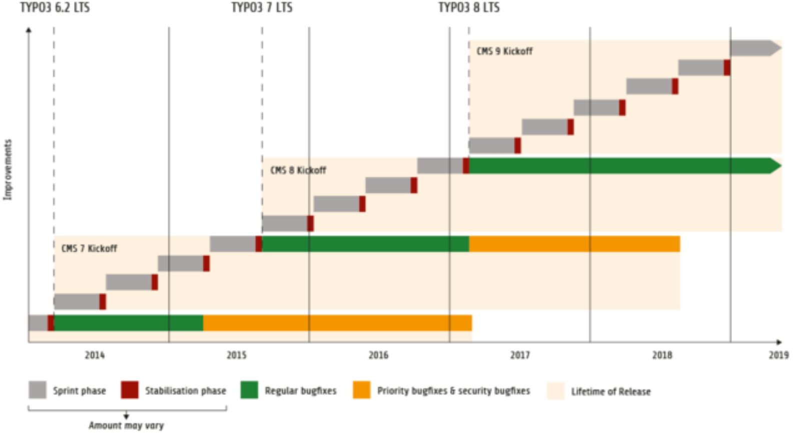
System Requirements

- 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	16/Jun/2015	Package Ecosystem, Composer
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/7.4
$ tar xzf typo3_src-7.4.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.4.0 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_7.4
- Official TYPO3 guide "TYPO3 Installation and Upgrading":

  http://docs.typo3.org/typo3cms/InstallationGuide
- General approach:
  - Check minimum system requirements (PHP, MySQL, etc.)
  - Review **deprecation_*.log** in old TYPO3 instance
  - Update all extensions to the latest version
  - Deploy new sources and run Install Tool -> Upgrade Wizard
  - Review startup module for backend users (optionally)

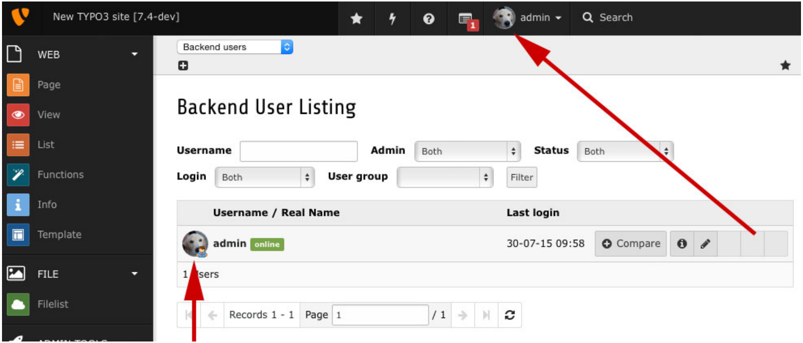
Chapter 1:

Backend User Interface

Backend User Interface

Avatars for Backend Users

To improve the user experience in collaborative content editing, backend users can use avatars now. These small user images are shown in the topbar, users list and other places.



Backend User Listing

Username | Admin: Both | Status: Both

Login: Both | User group | Filter

Username / Real Name	Last login	
admin online	30-07-15 09:58	Compare

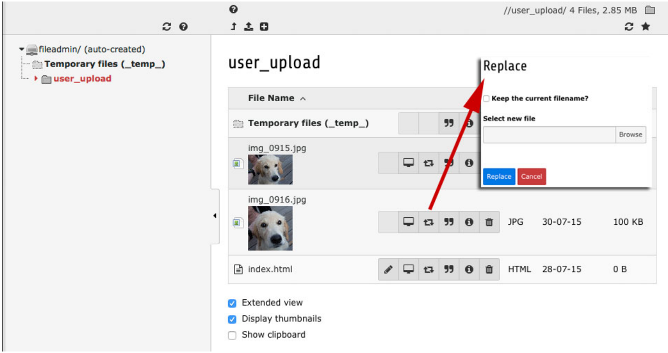
1 Users

Records 1 - 1 | Page 1 / 1

Backend User Interface

Replace Files

Files in the FAL record list can now be **replaced** (requires enabled "extended view"). Filename of the existing file can be retained or updated.



//user_upload/ 4 Files, 2.85 MB

- fileadmin/ (auto-created)
  - Temporary files (_temp_)
  - user_upload

user_upload

File Name			
Temporary files (_temp_)			
img_0915.jpg			
img_0916.jpg	JPG	30-07-15	100 KB
index.html	HTML	28-07-15	0 B

Replace

□ Keep the current filename?

Select new file

☑ Extended view

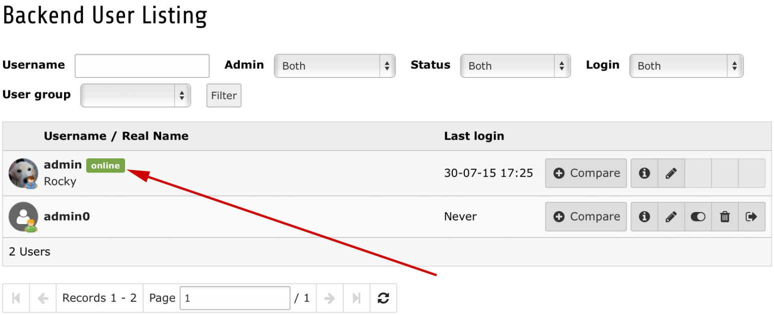
☑ Display thumbnails

□ Show clipboard

Backend User Interface

Online Status of Backend Users

The online status of backend users is shown in module "Backend Users".



Backend User Listing

Username		**Admin**	Both	**Status**	Both	**Login**	Both
User group		Filter					

Username / Real Name	Last login	
admin online Rocky	30-07-15 17:25	Compare
admin0	Never	Compare

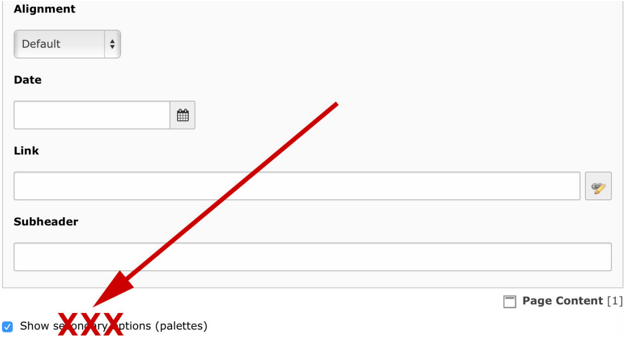
2 Users

Records 1 - 2 Page 1 / 1

Backend User Interface

Secondary Options Removed

The checkbox "Show secondary options (palettes)", the page TSconfig option `options.enableShowPalettes` and the TCA setting have been removed. The palettes are always visible and can not be hidden anymore.



Alignment

Default

Date

Link

Subheader

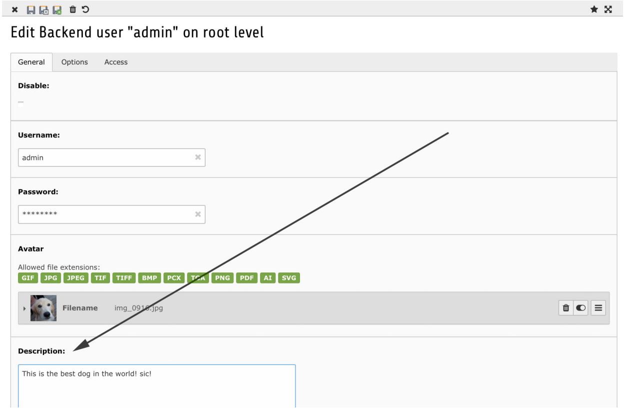
Page Content [1]

☑ Show secondary options (palettes)

Backend User Interface

Description for Backend Users

A new field "Description" has been added to backend user records.



Edit Backend user "admin" on root level

Disable:

Username: admin

Password: ********

Avatar

Allowed file extensions: GIF JPG JPEG TIF TIFF BMP PCX TGA PNG PDF AI SVG

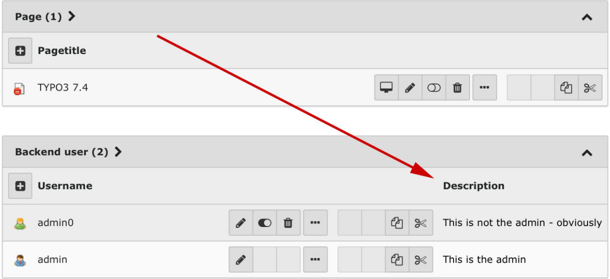
Filename img_0916.jpg

Description: This is the best dog in the world! sic!

Backend User Interface

Description for Table Columns

By configuring a column (usually `description`) in TCA setting `['TCA']['ctrl']['descriptionColumn']`, a description can be shown (improves usability for editors and administrators).



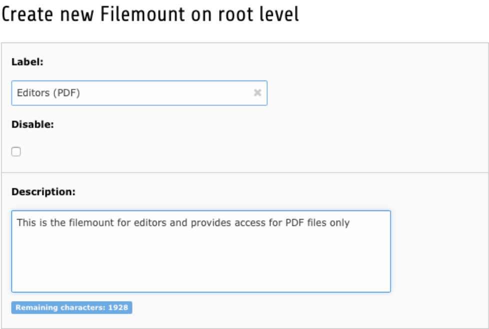
Pagetitle
TYPO3 7.4

Username	Description
admin0	This is not the admin - obviously
admin	This is the admin

Backend User Interface

Description for Filemounts

A new field "Description" has been added to filemount records. The field allows administrators to add a short description what a certain filemount should be used for, which documents it may contain, etc.



Create new Filemount on root level

Label:

Editors (PDF)

Disable:

□

Description:

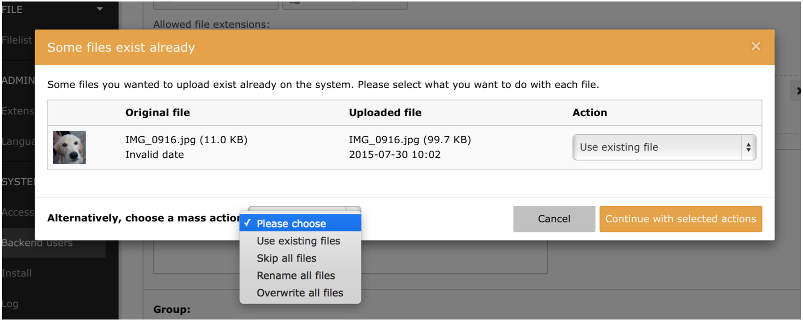
This is the filemount for editors and provides access for PDF files only

Remaining characters: 1928

Backend User Interface

Dialog for Existing Files on Upload

If a file upload would overwrite an existing file, a dialog is shown, asking the user to choose an action (e.g. replace, rename or skip).



Some files exist already

Some files you wanted to upload exist already on the system. Please select what you want to do with each file.

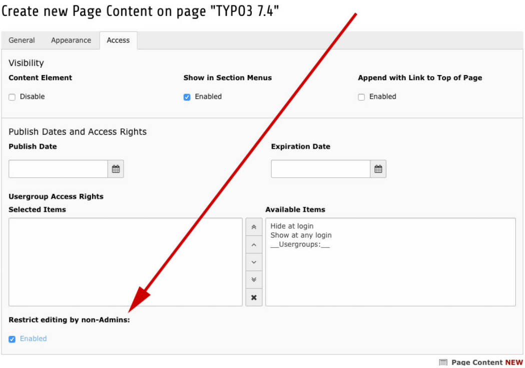
	Original file	Uploaded file	Action
	IMG_0916.jpg (11.0 KB) Invalid date	IMG_0916.jpg (99.7 KB) 2015-07-30 10:02	Use existing file

Alternatively, choose a mass action

Backend User Interface

Editing Content Elements Restriction

Content elements can now be restricted to be editable by admins only (similar to function "Restrict editing by non-Admins" for pages).



Create new Page Content on page "TYPO3 7.4"

Visibility

Content Element
□ Disable

Show in Section Menus
☑ Enabled

Append with Link to Top of Page
□ Enabled

Publish Dates and Access Rights

Publish Date

Expiration Date

Usergroup Access Rights

Selected Items

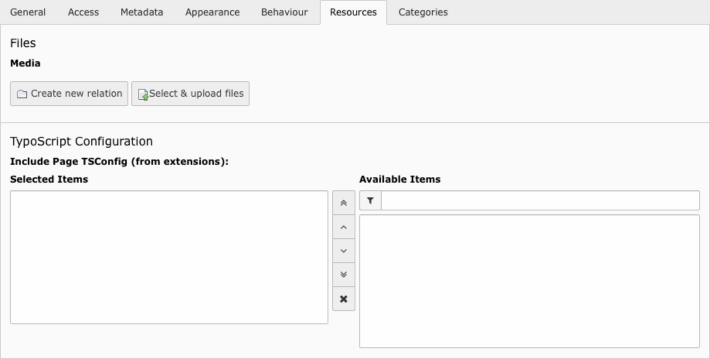
Available Items
Hide at login
Show at any login
__Usergroups:__

Restrict editing by non-Admins:
☑ Enabled

Backend User Interface

Include Static TSconfig Files (1)

In page properties an option allows to include a page TSconfig file (the same way as TypoScript static templates are included).



Files

Media

TypoScript Configuration

Include Page TSConfig (from extensions):

Selected Items

Available Items

Backend User Interface

Include Static TSconfig Files (2)

The following method registers a page TSconfig file:

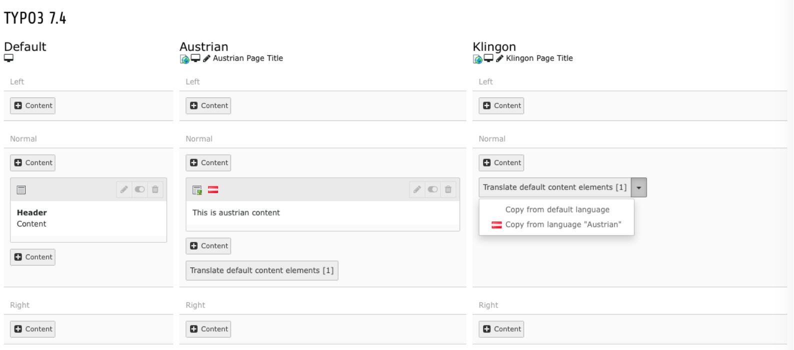
```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::registerPageTSConfigFile(
  'extension_name',
  'Configuration/PageTS/myPageTSconfigFile.txt',
  'My special configuration'
);
```

Backend User Interface

Real Copies of Content Elements

A new button has been added to each column in the "Page" module which allows *real* copies of content element into a language (not only references).

TYPO3 7.4



Default

Austrian
Austrian Page Title

Klingon
Klingon Page Title

Left

Content

Normal

Content

Header
Content

This is austrian content

Translate default content elements [1]

Copy from default language

Copy from language "Austrian"

Right

Chapter 2:

TSconfig & TypoScript

TSconfig & TypoScript

Data-Provider for Backend Layouts (1)

- It is now possible to define backend layouts via page TSconfig and also store them in files. For example:

```
mod {
  web_layout {
    BackendLayouts {
      exampleKey {
        title = Example
        config {
          backend_layout {
            colCount = 1
            rowCount = 2
            rows {
              1 {
                columns {
                  1 {
                    name = LLL:EXT:frontend/ ... /locallang_ttc.xlf:colPos.I.3
                    colPos = 3
                    colspan = 1
                  }
                }
              }
[...]
```

TSconfig & TypoScript

Data-Provider for Backend Layouts (2)

- (continued)

```
[...]
              2 {
                columns {
                  1 {
                    name = Main
                    colPos = 0
                    colspan = 1
                  }
                }
              }
            }
          }
        }
        icon = EXT:example_extension/Resources/Public/Images/BackendLayouts/default.gif
      }
    }
  }
}
```

TSconfig & TypoScript

Meta Tags Extended

- Option `page.meta` supports Open Graph attribute names now

```
page {
  meta {
    X-UA-Compatible = IE=edge,chrome=1
    X-UA-Compatible.attribute = http-equiv
    keywords = TYPO3
    # <meta property="og:site_name" content="TYPO3" />
    og:site_name = TYPO3
    og:site_name.attribute = property
    description = Inspiring people to share
    og:description = Inspiring people to share
    og:description.attribute = property
    og:locale = en_GB
    og:locale.attribute = property
    og:locale:alternate {
      attribute = property
      value.1 = fr_FR
      value.2 = de_DE
    }
    refresh = 5; url=http://example.com/
    refresh.attribute = http-equiv
  }
}
```

TSconfig & TypoScript

`languageField` Set by Default

- TypoScript option `select` (used in cObject CONTENT for example) required to set `languageField` explicitly
- This is not required anymore, as the setting is now fetched from the TCA information structure automatically

```
config.sys_language_uid = 2
page.10 = CONTENT
page.10 {
  table = tt_content
  select.where = colPos=0

  # the following line is not required anymore:
  #select.languageField = sys_language_uid

  renderObj = TEXT
  renderObj.field = header
  renderObj.htmlSpecialChars = 1
}
```

TSconfig & TypoScript

Individual Content Caching

- Since TYPO3 CMS 7.4 an individual content caching exists which – compare to `stdWrap.cache` – also works with COA objects (similar to "Magento Block Caching")

```
page = PAGE
page.10 = COA
page.10 {
  cache.key = coaout
  cache.lifetime = 60
  #stdWrap.cache.key = coastdWrap
  #stdWrap.cache.lifetime = 60
  10 = TEXT
  10 {
    cache.key = mycurrenttimestamp
    cache.lifetime = 60
    data = date : U
    strftime = %H:%M:%S
    noTrimWrap = |10: | |
  }
[...]
```

```
[...]
  20 = TEXT
  20 {
    data = date : U
    strftime = %H:%M:%S
    noTrimWrap = |20: | |
  }
}
```

Count Elements in a List

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

```
1 = TEXT
1 {
  value = x,y,z,1,2,3,a,b,c
  split.token = ,
  split.returnCount = 1
}
```

TSconfig & TypoScript

Sort Order of Tables in List View

- New TSconfig option `mod.web_list.tableDisplayOrder` has been added to the "List" module
- With this option, the order in which tables are displayed is configurable
- Keywords `before` and `after` can be used to specify an order relative to other table names

Syntax:

```
mod.web_list.tableDisplayOrder {
  <tableName> {
    before = <tableA>, <tableB>, ...
    after = <tableA>, <tableB>, ...
  }
}
```

For example:

```
mod.web_list.tableDisplayOrder {
  be_users.after = be_groups
  sys_filemounts.after = be_users
  pages_language_overlay.before = pages
  fe_users.after = fe_groups
  fe_users.before = pages
}
```

TSconfig & TypoScript

Content-Language in HTTP Header

- HTTP header `Content-language: XX` is sent by default, where "XX" is the ISO code of the `sys_language_content` configuration
- By using `config.disableLanguageHeader = 1`, this feature can be disabled (do not send the `Content-language` header at all)

TSconfig & TypoScript

Recursive Option for File Collections

- Folder-based file collections have an option to fetch all files recursively in the given folder now
- The option is also available in the TypoScript Object `FILES`

```
filecollection = FILES
filecollection {
  folders = 1:images/
  folders.recursive = 1
  renderObj = IMAGE
  renderObj {
    file.import.data = file:current:uid
  }
}
```

TSconfig & TypoScript

Extension `.ts` for Static Templates

- In TYPO3 CMS < 7.4, only the following file names are allowed as static TypoScript templates:
  - `constants.txt`
  - `setup.txt`
  - `include_static.txt`
  - `include_static_files.txt`
- For `constants` and `setup`, file extension `.ts` is also allowed now
- In this context, `.ts` is prioritised over `.txt`

TSconfig & TypoScript

Save & view Button

- Button "save & view" is configurable via TSconfig now
- TSconfig `TCEMAIN.preview.disableButtonForDokType` accepts a comma-separated list of "doktypes"
- Default value is "254, 255, 199" (which is: Storage Folder, Recycler and Menu Separator)
- As a consequence, the "save & view" button is not shown in folders and recycler pages by default anymore

TSconfig & TypoScript

stdWrap for `treatIdAsReference`

- For object `getImgResource` the option `treatIdAsReference` exists, which can be used to define that UIDs are treated as UIDs of `sys_file_reference` rather than `sys_file`.
- Option `treatIdAsReference` received stdWrap functionality now

Chapter 3:

TSconfig & TypoScript: Data Processors

TSconfig & TypoScript: Data Processors

Introducing Data Processors

- The following Data Processors have been introduced, which allow for a flexible processing of comma-separated lists, arrays, files, etc.:
  - `SplitProcessor`
  - `CommaSeparatedValueProcessor`
  - `FilesProcessor`
  - `GalleryProcessor`
  - `DatabaseQueryProcessor`
- See: `TYPO3\CMS\Frontend\DataProcessing`

TSconfig & TypoScript: Data Processors

The SplitProcessor

- The "SplitProcessor" allows to split values separated by a delimiter into an array

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.2 = TYPO3\CMS\Frontend\DataProcessing\SplitProcessor
page.10.dataProcessing.2 {
  if.isTrue.field = bodytext
  delimiter = ,
  fieldName = bodytext
  removeEmptyEntries = 1
  filterIntegers = 1
  filterUnique = 1
  as = keywords
}
```

- Possible usage in Fluid:

```
<f:for each="{keywords}" as="keyword">
  <li>Keyword: {keyword}</li>
</f:for>
```

TSconfig & TypoScript: Data Processors

The `CommaSeparatedValueProcessor` (1)

- The "CommaSeparatedValueProcessor" splits values separated by a delimiter in a two-dimensional array:

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.4 = TYPO3\CMS\Frontend\DataProcessing\CommaSeparatedValueProcessor
page.10.dataProcessing.4 {
  if.isTrue.field = bodytext
  fieldName = bodytext
  fieldDelimiter = |
  fieldEnclosure =
  maximumColumns = 2
  as = table
}
```

- Useful to process CSV files for example or `tt_content` data sets of CType "table"

  See an example usage in Fluid on the following slide

TSconfig & TypoScript: Data Processors

The CommaSeparatedValueProcessor (2)

- Possible usage in Fluid:

```
<table>
  <f:for each="{table}" as="columns">
    <tr>
      <f:for each="{columns}" as="column">
        <td>
          {column}
        </td>
      </f:for>
    <tr>
  </f:for>
</table>
```

TSconfig & TypoScript: Data Processors

The FilesProcessor (1)

- The "FilesProcessor" resolves file references, files, or files inside a folder or collection to be used for output in the frontend

```
tt_content.image.20 = FLUIDTEMPLATE
tt_content.image.20 {
  file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
  dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
  dataProcessing.10 {
    references.fieldName = image
    references.table = tt_content
    files = 21,42
    collections = 13,14
    folders = 1:introduction/images/,1:introduction/posters/
    folders.recursive = 1
    sorting = description
    sorting.direction = descending
    as = myfiles
  }
}
```

See an example usage in Fluid on the following slide

TSconfig & TypoScript: Data Processors

The `FilesProcessor` (2)

- Possible usage in Fluid:

```
<ul>
  <f:for each="{myfiles}" as="file">
    <li>
      <a href="{file.publicUrl}">{file.name}</a>
    </li>
  </f:for>
</ul>
```

TSconfig & TypoScript: Data Processors

The GalleryProcessor

- The "GalleryProcessor" calculates the max asset size of file sets

```
tt_content.text_media.20 = FLUIDTEMPLATE
tt_content.image.20 {
  file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
  dataProcessing {
    10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
    20 = TYPO3\CMS\Frontend\DataProcessing\GalleryProcessor
    20 {
      filesProcessedDataKey = files
      mediaOrientation.field = imageorient
      numberOfColumns.field = imagecols
      equalMediaHeight.field = imageheight
      equalMediaWidth.field = imagewidth
      maxGalleryWidth = 1000
      maxGalleryWidthInText = 1000
      columnSpacing = 0
      borderEnabled.field = imageborder
      borderWidth = 0
      borderPadding = 10
      as = gallery
    }
  }
}
```

TSconfig & TypoScript: Data Processors

The DatabaseQueryProcessor (1)

- The "DatabaseQueryProcessor" can be used to fetch data from the database

```
tt_content.mycontent.20 = FLUIDTEMPLATE
tt_content.mycontent.20 {
  file = EXT:myextension/Resources/Private/Templates/ContentObjects/MyContent.html
  dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\DatabaseQueryProcessor
  dataProcessing.10 {
    if.isTrue.field = records
    table = tt_address
    colPos = 1
    pidInList = 13,14
    as = myrecords
    dataProcessing {
      10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
      10 {
        references.fieldName = image
      }
    }
  }
}
```

See an example usage in Fluid on the following slide

TSconfig & TypoScript: Data Processors

The DatabaseQueryProcessor (2)

- Possible usage in Fluid:

```
<ul>
  <f:for each="{myrecords}" as="record">
    <li>
      <f:image image="{record.files.0}" ></f:image>
      <a href="{record.data.www}">{record.data.first_name} {record.data.last_name}</a>
    </li>
  </f:for>
</ul>
```

Chapter 4:

In-Depth Changes

In-Depth Changes

Driver Interface

- The following methods have been added to `DriverInterface`:
  - `getFolderInFolder`
  - `getFileInFolder`
- Every FAL driver should implement these new methods:

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

```
public function getFileInFolder(
  $fileName,
  $folderIdentifier
);
```

BREAKING CHANGE!

In-Depth Changes

IEC/SI Support in File Size Formatting

- File size formatting supports two keywords additionally to the list of labels now:
  - `iec` (default)
    (power of 2, labels: `| Ki| Mi| Gi| Ti| Pi| Ei| Zi| Yi`)
  - `si`
    (power of 10, labels: `| k| M| G| T| P| E| Z| Y`)
- Set formating in TypoScript for example:

```
bytes.labels = iec
```

```
echo GeneralUtility::formatSize(85123);
// => before "83.1 K"
// => now "83.13 Ki"
```

In-Depth Changes

Dependency Ordering Service (1)

- In many cases it is necessary to create a sorted list of items from a set of "dependencies". The ordered list is then used to execute actions in the given order.
- Some examples where the TYPO3 core uses this are:
  - hook execution order,
  - extension loading order,
  - listing of menu items,
  - etc.
- The `DependencyResolver` has been re-worked and provides a `DependencyOrderingService` now

In-Depth Changes

Dependency Ordering Service (2)

- Usage:

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['someExt']['someHook'][<some id>] = [
  'handler' => someClass::class,
  'runBefore' => [ <some other ID> ],
  'runAfter' => [ ... ],
  ...
];
```

- Example:

```
$hooks = $GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['someExt']['someHook'];
$sorted = GeneralUtility:makeInstance(DependencyOrderingService::class)->orderByDependencies(
  $hooks, 'runBefore', 'runAfter'
);
```

In-Depth Changes

Hooks and Signals (1)

- Hook to post-process `InlineRecordContainer::checkAccess` results has been added
- `InlineRecordContainer::checkAccess` can be used to check the access to related inline records
- The following code registers the hook:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_tceforms_inline.php']
  ['checkAccess'][] = 'My\\Package\\HookClass->hookMethod';
```

In-Depth Changes

Hooks and Signals (2)

- Hook to post-process login failures in `AbstractUserAuthentication::checkAuthentication` has been added
- Process stalls for 5 seconds in case of a failed login by default
- Using this new hook, alternative solutions could be implemented (e.g. to prevent brute force attacks)
- The following code registers the hook:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_userauth.php']
  ['postLoginFailureProcessing'][] = 'My\\Package\\HookClass->hookMethod';
```

In-Depth Changes

Hooks and Signals (3)

- New signal `recordMarkedAsMissing` is emitted when the FAL indexer encounters a `sys_file` record which does not have a corresponding filesystem entry and marks it as missing. The signal passes the `sys_file` record UID.
- This is useful in extensions which provide or extend file management capabilities such as versioning, synchronizations, recovery, etc.
- Signal `afterMappingSingleRow` is emitted whenever the DataMapper creates an object

In-Depth Changes

HTML in TypoLink Titles

- Quotes in TypoLink titles are *escaped* automatically now
- This means, instances where HTML codes are already escaped manually, will break the frontend output in TYPO CMS 7.4

  Before: `’Some &quot;special&quot; title’`

  Becomes: `’Some &amp;quot;special&amp;quot; title’`
- It is recommended to avoid escaping, due to the fact that TYPO3 takes care of escaping HTML in TypoLink titles now

BREAKING CHANGE!

In-Depth Changes

Miscellaneous (1)

- By configuring the backend user permission `Files->replace`, users can be allowed to or prevented from replacing Files in module Filelist
- A hash is used in the filename of files, generated by the FileWriter, if no other log file has been configured
  - before: `typo3temp/logs/typo3.log`
  - now: `typo3temp/logs/typo3_<hash>.log`

  (value `<hash>` is calculated based on the encryption key)

In-Depth Changes

Miscellaneous (2)

- Classes used in hooks have to follow the autoloading mechanism
- Therefore the hook definition can be shortened now:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tce']['formevals']
  [\TYPO3\CMS\Saltedpasswords\Evaluation\FrontendEvaluator::class] = '';
```

BREAKING CHANGE!

Chapter 5:

Extbase & Fluid

Extbase & Fluid

Anchor for Pagination Widget

- This new feature allows to add a key `section` to the configuration of a Fluid pagination widget
- The anchor gets appended to every link of the pagination widget
- The following code adds an anchor `#archive`:

```
<f:widget.paginate objects="{plantpestWarnings}" as="paginatedWarnings"
  configuration="{section: 'archive', itemsPerPage: 10, insertAbove: 0, insertBelow: 1,
  maximumNumberOfLinks: 10}">

   [...]

</f:widget.paginate>
```

Extbase & Fluid

Attribute `base` for DateViewHelper

- DateViewHelper has been extended by an optional attribute named `base`
- The attribute can be used to calculate relative time specification for dates
- If the date is a DateTime object, `base` is ignored
- The following example returns "2016", if `dateObject` is a date in 2017:

```
<f:format.date format="Y" base="{dateObject}">-1 year</f:format.date>
```

(see PHP documentation for a list of valid values)

Extbase & Fluid

Option dataProcessing for FLUIDTEMPLATE

- In TYPO3 CMS 7.3, option dataProcessing for cObject FLUIDTEMPLATE has been introduced
- Its FluidTemplateDataProcessorInterface has been refactored to DataProcessorInterface, which also affects method process()

```
public function process(
  ContentObjectRenderer $cObj,
  array $contentObjectConfiguration,
  array $processorConfiguration,
  array $processedData
);
```

BREAKING CHANGE!

Chapter 6:

Deprecated/Removed Functions

Deprecated/Removed Functions

System Extension cms Removed (1)

- System extension cms has been removed
- Extension developers should review dependency settings in file ext_emconf.php

```
[...]
'constraints' => array(
  'depends' => array(
    // 'cms' => ' ... ',           <= WRONG!
    'typo3' => '7.0.0-7.99.99',
  ),
),
[...]
```

- Most of the functionality has been migrated to system extension frontend (this might require an update of language references, see next slide)

Deprecated/Removed Functions

System Extension cms Removed (2)

- Required updates of references to language files:

```
OLD: typo3/sysext/cms/web_info/locallang.xlf
NEW: typo3/sysext/frontend/Resources/Private/Language/locallang_webinfo.xlf

OLD: typo3/sysext/cms/locallang_ttc.xlf
NEW: typo3/sysext/frontend/Resources/Private/Language/locallang_ttc.xlf

OLD: typo3/sysext/cms/locallang_tca.xlf
NEW: typo3/sysext/frontend/Resources/Private/Language/locallang_tca.xlf

OLD: typo3/sysext/cms/layout/locallang_db_new_content_el.xlf
NEW: typo3/sysext/backend/Resources/Private/Language/locallang_db_new_content_el.xlf

OLD: typo3/sysext/cms/layout/locallang.xlf
NEW: typo3/sysext/backend/Resources/Private/Language/locallang_layout.xlf

OLD: typo3/sysext/cms/layout/locallang_mod.xlf
NEW: typo3/sysext/backend/Resources/Private/Language/locallang_mod.xlf

OLD: typo3/sysext/cms/locallang_csh_webinfo.xlf
NEW: typo3/sysext/frontend/Resources/Private/Language/locallang_csh_webinfo.xlf

OLD: typo3/sysext/cms/locallang_csh_weblayout.xlf
NEW: typo3/sysext/frontend/Resources/Private/Language/locallang_csh_weblayout.xlf
```

Deprecated/Removed Functions

PageRenderer Methods Deprecated

- The following `PageRenderer` methods have been classified as **deprecated**:

```
TYPO3\CMS\Backend\Controller\BackendController::getPageRenderer()
TYPO3\CMS\Backend\Template\DocumentTemplate::getPageRenderer()
TYPO3\CMS\Backend\Template\FrontendDocumentTemplate::getPageRenderer()
TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController::getPageRenderer()
```

- The following code should be used to get an instance of the PageRenderer instead:

```
\TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(\TYPO3\CMS\Core\Page\PageRenderer::class)
```

Deprecated/Removed Functions

Deprecated GeneralUtility Methods

- The following `GeneralUtility` methods have been classified as **deprecated** and will be removed in TYPO3 CMS version 8:

```
GeneralUtility::modifyHTMLColor()
GeneralUtility::modifyHTMLColorAll()
GeneralUtility::isBrokenEmailEnvironment()
GeneralUtility::normalizeMailAddress()
GeneralUtility::formatForTextarea()
GeneralUtility::getThisUrl()
GeneralUtility::cleanOutputBuffers()
GeneralUtility::readLLfile()
```

- Method `readLLfile()` can be replaced by the following code:

```
/** @var $languageFactory \TYPO3\CMS\Core\Localization\LocalizationFactory */
$languageFactory = GeneralUtility::makeInstance(
  \TYPO3\CMS\Core\Localization\LocalizationFactory::class
);
$languageFactory->getParsedData($fileToParse, $language, $renderCharset, $errorMode);
```

Deprecated/Removed Functions

JavaScript Libraries Removed

- JavaScript libraries `prototype.js` and `scriptaculous` have been removed. As a consequence, the following TypoScript properties do not have any functionality anymore:
  - `page.javascriptLibs.Prototype`
  - `page.javascriptLibs.Scriptaculous.*`
- Using the following attributes in ViewHelper `be.container` results in an error:

```
<f:be.container loadPrototype="false" loadScriptaculous="false"
     scriptaculousModule="someModule,someOtherModule">
```

- As a replacement, `jQuery` and `RequireJS` should be used (which are already loaded in the backend by default)

Deprecated/Removed Functions

Deprecated: `init.php`, `mod.php` and `ajax.php`

- In order to clean up the content of folder `typo3`, the following files have been marked as **deprecated**: `init.php`, `mod.php` and `ajax.php`
- The following code can be used for Init Entry Points:

```
call_user_func(function() {
  $classLoader = require __DIR__ . '/vendor/autoload.php';
  (new \TYPO3\CMS\Backend\Http\Application($classLoader))->run();
});
```

- The following method call can be used to access `mod.php`:

```
BackendUtility::getModuleUrl()
```

Deprecated/Removed Functions

TCA: Additional Palette Removed

- String showitem of TCA key types allowed developers to define an additional palette
- This has been removed and migrated to the normal palette now
- Before:

```
'types' => array(
  'aType' => array(
    'showitem' => 'aField;aLabel;anAdditionalPaletteName',
  ),
),
```

- Now:

```
'types' => array(
  'aType' => array(
    'showitem' => 'aField;aLabel, --palette--;;anAdditionalPaletteName',
  ),
),
```

Deprecated/Removed Functions

Miscellaneous (1)

- cObjects "Text" and "Text with Images" had a checkbox "RTE enabled" in the past. This has been removed, including the appropriate TCA option `flag`.
- The following TypoScript options to include PHP files have been removed:
  - `config.includeLibrary`
  - `config.includeLibs`
- Directory `typo3/ext` has been removed (but not the option to use global extensions: the directory can be created manually)

Deprecated/Removed Functions

Miscellaneous (2)

- ExtCore (a standalone, slim ExtJS adapter) has been removed, including the following TypoScript options:
  - `page.javascriptLibs.ExtCore.*`
  - `page.javascriptLibs.ExtJs.*`

  This also includes the option in the `<f:be.container>` ViewHelper
- So called "BigButtons" ("Edit Page Properties", "Move Page",...) have been removed, including their TSconfig settings `mod.we_layout.disableBigButtons`

Deprecated/Removed Functions

Miscellaneous (3)

- Error- and exception handling can not longer be configured in extensions (e.g. overwritten in `ext_localconf.php`), but in files `LocalConfiguration.php` or `AdditionalConfiguration.php` only
- Field "General Record Storage Page", which contained the storage PID for a page, has been removed. The storage PID must be configured using TypoScript or FlexForms now.
- Function `IconUtility::getIcon()` has been classified as **deprecated** (use method `IconUtility::getSpriteIconForRecord()` instead)

Chapter 7:

Sources and Authors

Sources and Authors

Sources

TYPO3 News:

- http://typo3.org/news

Release Infos:

- http://wiki.typo3.org/TYPO3_CMS_7.4.0
- INSTALL.md and ChangeLog
- typo3/sysext/core/Documentation/Changelog/7.4/*

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, Pierrick Caillon, Sergio Catala, Jigal van Hemert, Michel Mix, Sinisa Mitrovic, Angeliki Plati, Nena Jelena Radovic, Roberto Torresani

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