TYPO3 Best Practice Workshop

Extension Key: best_practice_workshop

Language: en

Keywords: keywords comma-separated

Table of Contents

TYPO3 Best Practice Workshop....................1

Introduction...3

TYPO3 Development System................................4

Local Development vs. Remote Development...........4

TYPO3 Provider..5

Directory Structure...6

Install Tool Settings..7

Page and User TSconfig..8

Rich Text Editor (RTE)...9

Paragraph and Character Styles.............................9

Modular TypoScript Code...................................12

Extension Manager...13

Templates..16

Modern Template Building....................................16

Templavoila (Futuristic Template Building)..............18

TypoScript: Standard Configuration...................20

Digital Asset Management (DAM).......................21

Multilanguage Websites......................................23

Multiple-Tree Concept...23

One-Tree Conept...23

Debugging...24

Tools..25

Firefox Add-on: Firebug..25

Firefox Add-on: Web Developer.............................25

Useful Links...26

.. 27

Introduction

There are many different ways to implement a TYPO3 project. For those new to TYPO3 it is often difficult to decide which way to choose and to understand which configurations are necessary and meaningful.

This workshop presents a number of shortcuts and best practices to reduce development time. These have been found very useful in the daily work of the authors, but we understand that some developers may want to choose alternatives to the methods shown here.

TYPO3 Development System

Local Development vs. Remote Development

TYPO3 can be developed on a local PC. In this case, typically one of the following software packages will be installed: XAMPP, LAMP, WAMP, MAMP. (?AMP stands for Apache, MySQL, PHP).

Once the project is completed and ready to go live, it can be transferred to the webserver where it will be accessible for the users.

However it is recommended to do the development on the target system, rather than on a local PC. Otherwise there can be compatibility issues arising from different versions or configuration of the Apache, MySQL and PHP software.

Development on the target system also allows the client to review the stages of development and there is no effort to transfer the project from the local PC to the remote system (which can be several hours of work).

In many cases the development on a remote system is also faster, even though all data has to be transferred over the Internet. The reason is, that a computer in a data center is optimized as a webserver, where a local PC is optimized for desktop applications

Extension Development

TYPO3 extensions are programmed in PHP language. The recommended development platform is Eclipse (Java based), either in form of PDT (PHP Development Tools, free) or ZendStudio (commercial). Most of the TYPO3 developers are using Eclipse, so there is broad support and know-how for it.

TYPO3 Provider

TYPO3 can be installed either in a low-cost webspace (shared server, 5 to 50 Euro) or on an exclusive server (50 to 500 Euro).

Most shared hosting packages are NOT suited for TYPO3 because of the system requirements. The default memory limit for PHP scripts is 16 MB (for PHP5), while TYPO3 typically needs 20 to 50 MB (and more depending on complexity of the project and installed extensions).

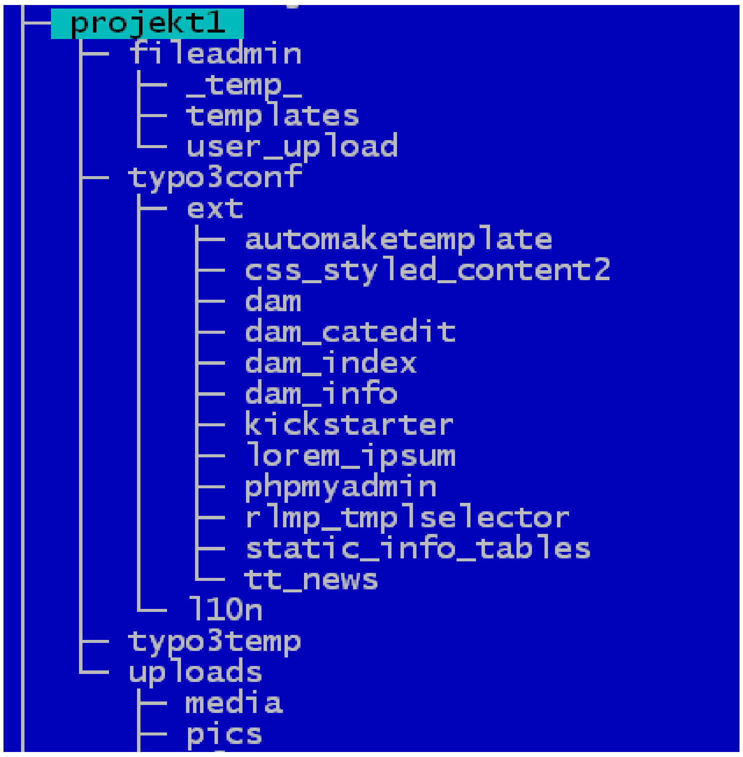
TYPO3 also requires additional software to be installed on the server: ImageMagick or GraphicsMagick for all image processing functions and tools for indexing external documents as well as additional functionality for the Digital Asset Management (DAM) feature.

There are a number of hosting companies that offer specialized TYPO3 hosting (both webspace and servers). Here you can expect that the system requirements are met and that TYPO3 along with the necessary tools are already installed.

Depending on the complexity of the web pages, server performance and number of customers hosted on the server, a webspace package can typically handle between 100.000 and 1 million page request per month. Websites that expect more traffic should be hosted on an exclusive server. Websites with several million page impressions per month will require special clustering configurations.

For maximum flexibility it is strongly recommended that a Secure Shell (SSH) access is available.

Directory Structure



```
├─ projekt1
│   ├─ fileadmin
│   │   ├─ _temp_
│   │   ├─ templates
│   │   └─ user_upload
│   ├─ typo3conf
│   │   ├─ ext
│   │   │   ├─ automaketemplate
│   │   │   ├─ css_styled_content2
│   │   │   ├─ dam
│   │   │   ├─ dam_catedit
│   │   │   ├─ dam_index
│   │   │   ├─ dam_info
│   │   │   ├─ kickstarter
│   │   │   ├─ lorem_ipsum
│   │   │   ├─ phpmyadmin
│   │   │   ├─ rlmp_tmplselector
│   │   │   ├─ static_info_tables
│   │   │   └─ tt_news
│   │   └─ l10n
│   ├─ typo3temp
│   └─ uploads
│       ├─ media
│       ├─ pics
```

Install Tool Settings

After installing TYPO3 and BEFORE starting with the project, the following changes should be made to the default settings in the Install Tool:

```
["GFX"]["TTFdpi"] = '96'

["BE"]["compressionLevel"] = '3';

["FE"]["compressionLevel"] = '3';

["GFX"]["gdlib_2"] = '1';

["GFX"]["noIconProc"] = '0';

['SYS']['forceReturnPath'] = '1'

['BE']['forceCharset'] = 'utf-8'['SYS']

['setDBinit'] = 'SET NAMES utf8;'.chr(10).'SET CHARACTER SET utf8;'.chr(10).'SET SESSION
character_set_server=utf8;'.chr(10).''

['SYS']['UTF8filesystem'] = '1'
```

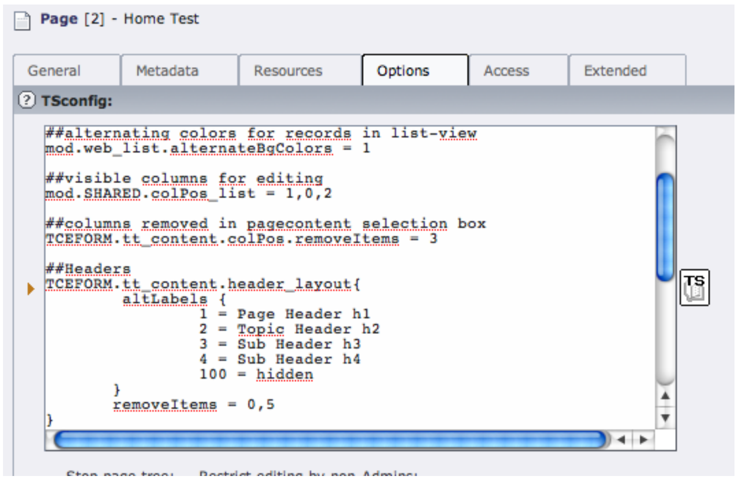
Page and User TSconfig

Page TSconfig concerns the configuration of the modules in the TYPO3 backend. The Page TSconfig is entered into the root page and extends to the pages within this branch.

see: http://typo3.org/documentation/document-library/core-documentation/doc_core_tsconfig/current/

Page Tsconfig Top Level Objects:

mod, RTE, TCMAIN, TCEFORM, TSFE, tx_[extension key with no underscore]

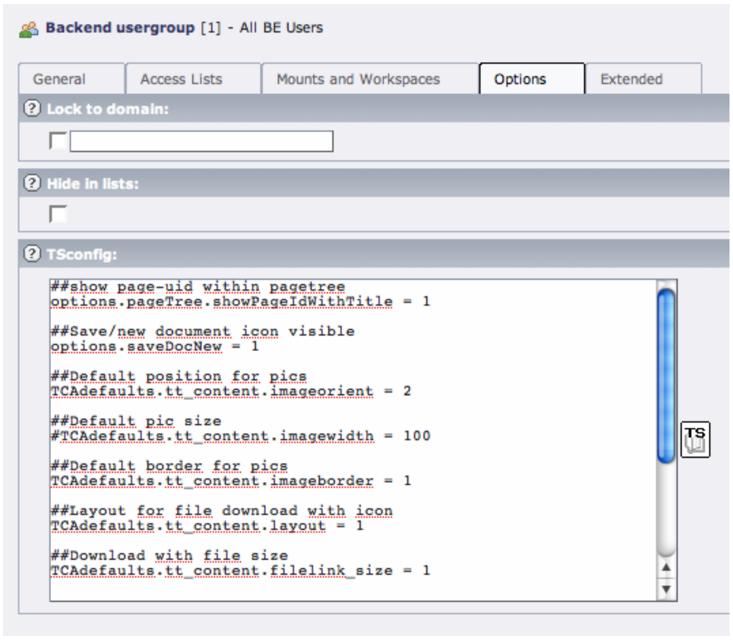


```
##alternating colors for records in list-view
mod.web_list.alternateBgColors = 1

##visible columns for editing
mod.SHARED.colPos_list = 1,0,2

##columns removed in pagecontent selection box
TCEFORM.tt_content.colPos.removeItems = 3

##Headers
TCEFORM.tt_content.header_layout{
        altLabels {
                1 = Page Header h1
                2 = Topic Header h2
                3 = Sub Header h3
                4 = Sub Header h4
                100 = hidden
       }
       removeItems = 0,5
}
```

User TSconfig is designed for users or groups of users. User TSconfig can be entered for both BE users and BE groups.

User TSconfig Top Level Objects:

admPanel, options, setup.defaults, setup.override, TCAdefaults.[tablename].[field], auth, page, tx_[extension key with no underscore]

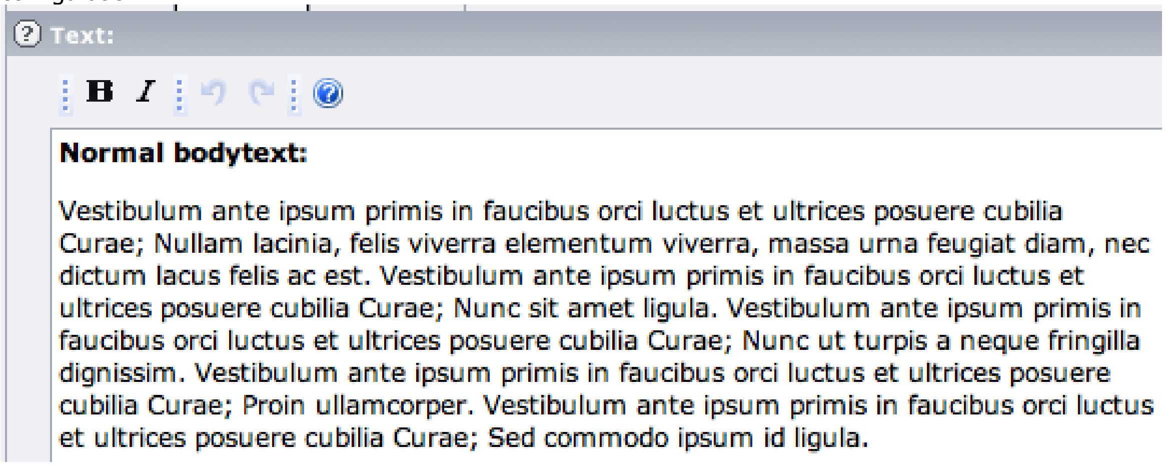


```
##show page-uid within pagetree
options.pageTree.showPageIdWithTitle = 1

##Save/new document icon visible
options.saveDocNew = 1

##Default position for pics
TCAdefaults.tt_content.imageorient = 2

##Default pic size
#TCAdefaults.tt_content.imagewidth = 100

##Default border for pics
TCAdefaults.tt_content.imageborder = 1

##Layout for file download with icon
TCAdefaults.tt_content.layout = 1

##Download with file size
TCAdefaults.tt_content.filelink_size = 1
```

Rich Text Editor (RTE)

There are several text editors available for TYPO3, this chapter discusses the extension rtehtmlarea, which is part of the TYPO3 default software package.

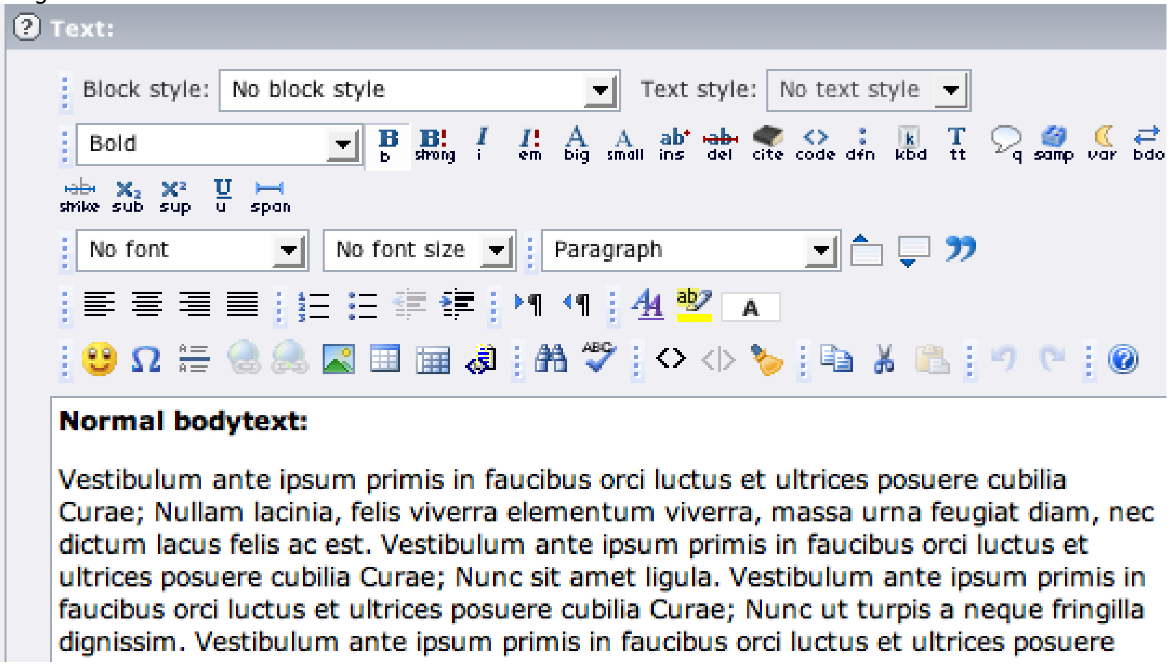
Minimal configuration:



Normal bodytext:

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nullam lacinia, felis viverra elementum viverra, massa urna feugiat diam, nec dictum lacus felis ac est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc sit amet ligula. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc ut turpis a neque fringilla dignissim. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ullamcorper. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed commodo ipsum id ligula.

Demo configuation:



Normal bodytext:

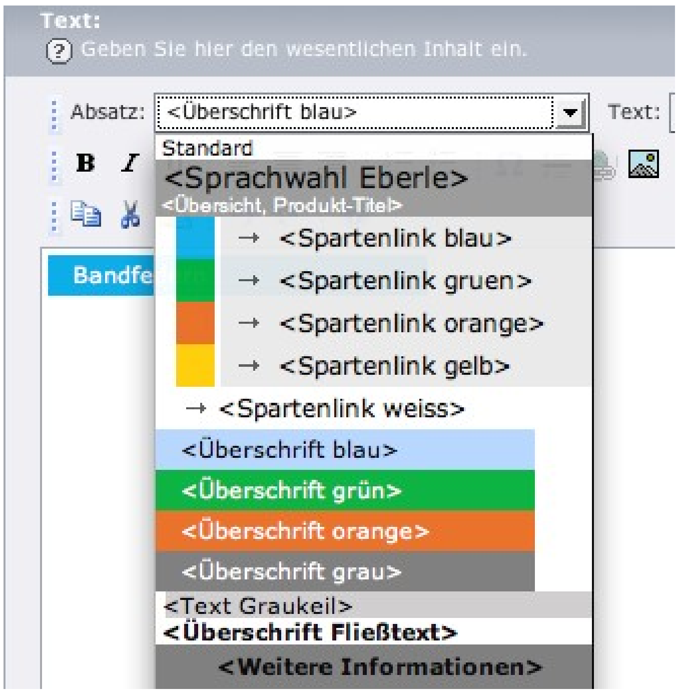
Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nullam lacinia, felis viverra elementum viverra, massa urna feugiat diam, nec dictum lacus felis ac est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc sit amet ligula. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc ut turpis a neque fringilla dignissim. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere

The RTE is configured using TSconfig, there are several sample configurations available in forums. With TSconfig the appearance and features of the editor can be configured on the page level and/or user level. This means that different parts of the website can use different RTE configurations, as well as there can be configurations for different users.

It is recommended to disable the insertion of images inside the RTE, rather the 'text with image' content element should be used. When images are uploaded in the RTE, they will all be stored in the same directory and some of the advanced options for images are not available.

Paragraph and Character Styles

In order to maintain a uniform look of the content elements on the website and keeping the elements in line with CI rules, a number of paragraph styles that the editor is able to choose from can be preconfigured. Along with the configuration of the RTE toolbar, the editor can then only choose from a predefined set of style elements.



Using CSS and TSconfig it is also possible to configure the RTE for a WYSIWYG presentation. Currently a two-step configuration is necessary: the styles need to be set both in an external CSS file as well as in TSconfig.

The externals CSS file is referenced in TSconfig:

```
RTE.default.contentCSS = fileadmin/templates/css/rte.css
```

Styles that are not defined in the CSS file will not be available in the selection boxes of the RTE. This is how the configuration looks in the CSS file:

```
p.moreinfo {
      background: #7a7a7a url(../img/moreinfo.gif) no-repeat;
      font-weight: bold;
      color: #ffffff;
      margin: 3px 0 0 0;
      padding: 4px 0 3px 30px;

}
```

Paragraph styles are defined as p-classes, character styles as span-classes:

```
span.important { color: #8A0020; }
span.name-of-person { color: #10007B; }
span.detail { color: #186900; }
```

Link styles are defined as a-classes:

```
a.external-link {font-weight: bold;}
a.external-link-new-window {}
a.internal-link {}
a.internal-link-new-window {}
a.download {}
a.mail {}
```

Classes defined in CSS can be added or removed from the RTE configuration:

```
RTE.default.classesParagraph := removeFromList(csc-frame-frame1, csc-frame-frame2)
RTE.default.proc.allowedClasses .= removeFromList(csc-frame-frame1, csc-frame-frame2)

RTE.default.classesParagraph := addToList(moreinfo, ....)
RTE.default.proc.allowedClasses := addToList(moreinfo, ....)
```

WYSIWYG styling can be configured with the following TSconfig code:

```
RTE.classes {
    moreinfo {
    name = Weitere Informationen
    value = font-size: 11px; font-weight: bold; padding: 4px 0 5px 30px; background: #7a7a7a
url(fileadmin/templates/main/img/moreinfo.gif) no-repeat; text-decoration:none);
}
```

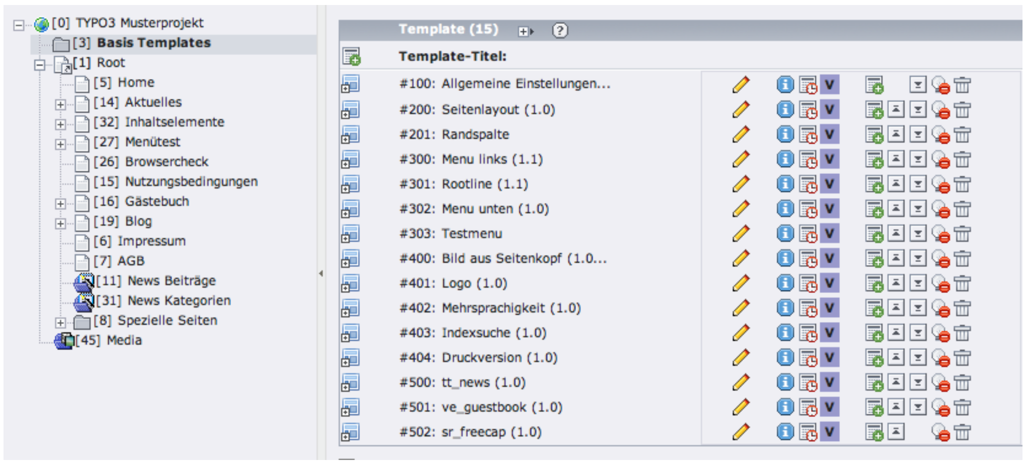
In some cases it will be necessary to use slightly different font-sizes in the CSS for the RTE than on the website to achive the same visual representation, this is because the general font size setting for the RTE is less than 100%.

Finally, use the code below to add the paragraph and character styles to the dropdown boxes (comma separated list):

```
RTE.default.classesParagraph = moreinfo, ...
RTE.default.classesCharacter = important, ...
```

Modular TypoScript Code

In practice it is very handy to divide the TypoScript code into snippets according to its function. These snippets are saved each as extension templates in a system folder.

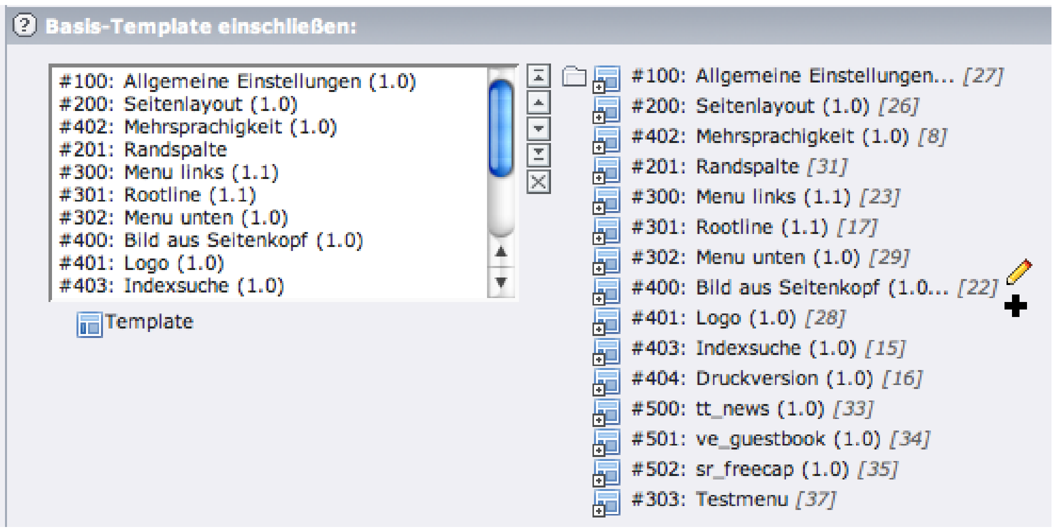


[0] TYPO3 Musterprojekt
- [3] **Basis Templates**
- [1] Root
  - [5] Home
  - [14] Aktuelles
  - [32] Inhaltselemente
  - [27] Menütest
  - [26] Browsercheck
  - [15] Nutzungsbedingungen
  - [16] Gästebuch
  - [19] Blog
  - [6] Impressum
  - [7] AGB
  - [11] News Beiträge
  - [31] News Kategorien
  - [8] Spezielle Seiten
- [45] Media

Template (15)

Template-Titel:
#100: Allgemeine Einstellungen...
#200: Seitenlayout (1.0)
#201: Randspalte
#300: Menu links (1.1)
#301: Rootline (1.1)
#302: Menu unten (1.0)
#303: Testmenu
#400: Bild aus Seitenkopf (1.0...
#401: Logo (1.0)
#402: Mehrsprachigkeit (1.0)
#403: Indexsuche (1.0)
#404: Druckversion (1.0)
#500: tt_news (1.0)
#501: ve_guestbook (1.0)
#502: sr_freecap (1.0)

On the root page of the site with one main template these extensions are the inserted in “Include basis templates”.



Basis-Template einschließen:

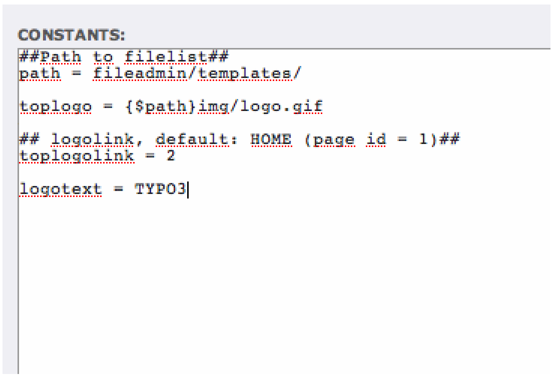
```
#100: Allgemeine Einstellungen (1.0)
#200: Seitenlayout (1.0)
#402: Mehrsprachigkeit (1.0)
#201: Randspalte
#300: Menu links (1.1)
#301: Rootline (1.1)
#302: Menu unten (1.0)
#400: Bild aus Seitenkopf (1.0)
#401: Logo (1.0)
#403: Indexsuche (1.0)
```

Template

```
#100: Allgemeine Einstellungen... [27]
#200: Seitenlayout (1.0) [26]
#402: Mehrsprachigkeit (1.0) [8]
#201: Randspalte [31]
#300: Menu links (1.1) [23]
#301: Rootline (1.1) [17]
#302: Menu unten (1.0) [29]
#400: Bild aus Seitenkopf (1.0... [22]
#401: Logo (1.0) [28]
#403: Indexsuche (1.0) [15]
#404: Druckversion (1.0) [16]
#500: tt_news (1.0) [33]
#501: ve_guestbook (1.0) [34]
#502: sr_freecap (1.0) [35]
#303: Testmenu [37]
```

Using this method you can edit the code more easily and reuse the snippets for other projects.

Defining constants in the root template for the sitename, filepath etc. is also very helpful. Hence these constants can be inserted multiple times into the setup snippets as variables.



CONSTANTS:

```
##Path to filelist##
path = fileadmin/templates/

toplogo = {$path}img/logo.gif

## logolink, default: HOME (page id = 1)##
toplogolink = 2

logotext = TYPO3
```

Extension Manager

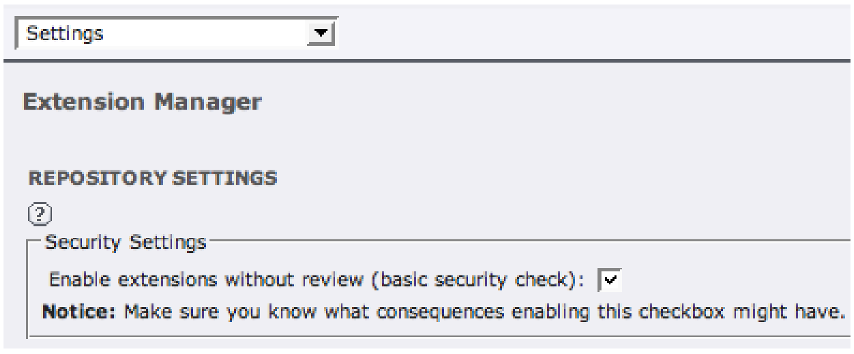
There are 3 categories of extensions:

- **System extensions** are part of the TYPO3 package and supplied with the source code (example: RTE, indexed_search, css_styled_content). They are located in the directory typo3/sysext
- **Global extensions** are available to all TYPO3 projects that share the same source code installation, they are located in the directory typo3/ext (typically empty)
- **Local extensions** are available only to the TYPO3 project in which they are installed. They are located in the directory typo3conf/ext

It is possible to have one extension installed for example as global AND als local extension, in this case the local extension takes precedence.

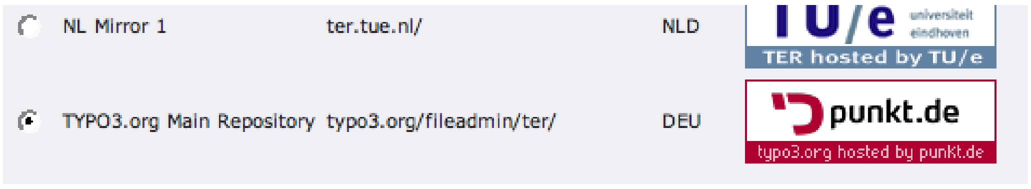
It is recommended to install additional extensions always as local. Only in rare circumstances it will be necessary to update system extensions, updates of these are made available as part of new TYPO3 versions.

Extensions that have been released to the public are available for download from the TYPO3 Extension Repository (TER). Those extensions have been reviewed by the TYPO3 Security Team are always visible in the TER, in order to see extensions without review a checkbox has to be set in the extension manager settings:



Enable extensions without review (basic security check): ✓
Notice: Make sure you know what consequences enabling this checkbox might have.

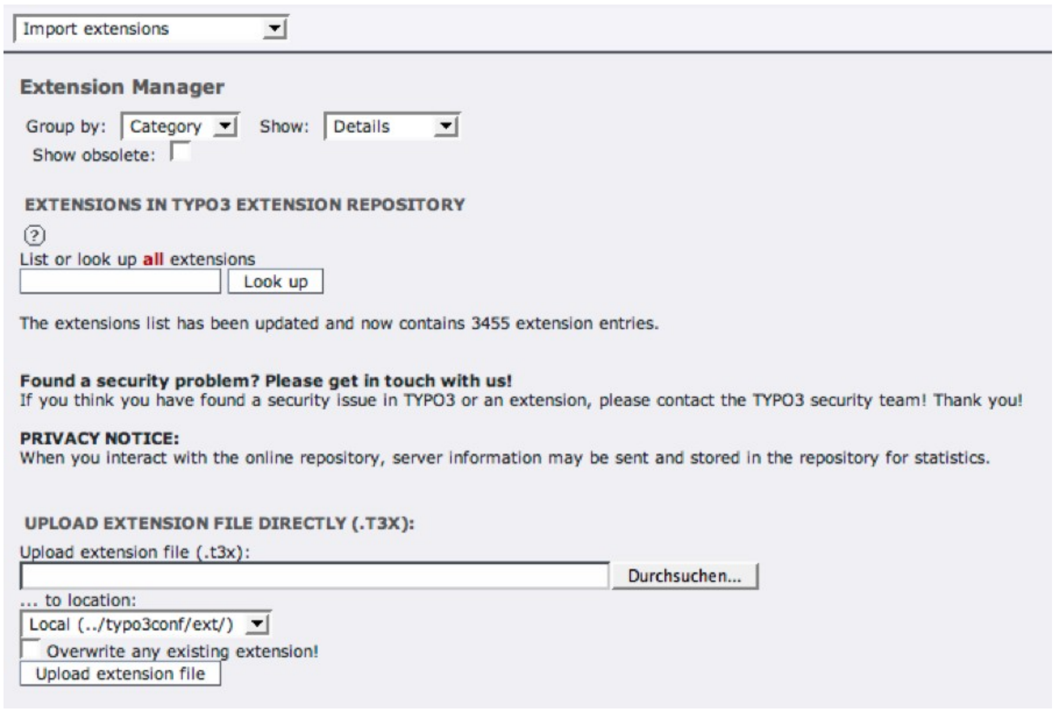
Also you may want to download extensions from the official TER and not from one of the mirror servers (since sometimes there is a delay for updates to find their way to the mirrors):



○	NL Mirror 1	ter.tue.nl/	NLD	TU/e TER hosted by TU/e
●	TYPO3.org Main Repository	typo3.org/fileadmin/ter/	DEU	punkt.de typo3.org hosted by punkt.de

Before installing a new extension, please follow these guidelines:

1. Only install extensions from which you know what they are doing – read the manual! You may also want to take a look at the extensions' source code. You can search for extensions on www.typo3.org/extensions/repository/.
2. Extensions on typo3.org are listed by number of downloads – new (and perhaps very interesting) extensions may be found near the bottom of the list...
3. Check TYPO3 and PHP version requirements as well as dependancies of other extensions
4. Extensions may have a status of Experimental – Alpha – Beta – Stable. If in doubt, prefer a stable version and don't expect that those marked as experimental or alpha behave as expected in all circumstances. Extensions marked as obsolete should not be installed – the functionality ist not required anymore (in most cases the function has been integrated in the TYPO3 source code)
5. Update the TER list to make sure to download the latest version
6. Make sure to have a recent backup if it is necessary to undo the installation
7. Uninstall extensions that are no longer needed (or just have been installed for testing).



Extension Manager

Group by: Category Show: Details
Show obsolete: ☐

EXTENSIONS IN TYPO3 EXTENSION REPOSITORY

List or look up **all** extensions
Look up

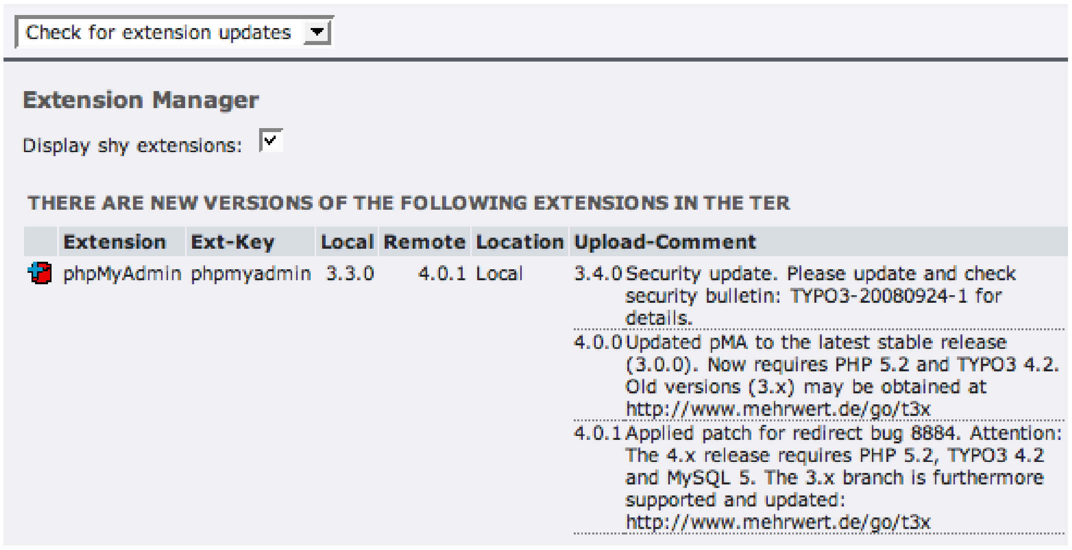
The extensions list has been updated and now contains 3455 extension entries.

Found a security problem? Please get in touch with us!
If you think you have found a security issue in TYPO3 or an extension, please contact the TYPO3 security team! Thank you!

PRIVACY NOTICE:
When you interact with the online repository, server information may be sent and stored in the repository for statistics.

UPLOAD EXTENSION FILE DIRECTLY (.T3X):
Upload extension file (.t3x):
Durchsuchen...
... to location:
Local (../typo3conf/ext/)
☐ Overwrite any existing extension!
Upload extension file

TYPO3 projects should be regularly checked for available extension updates. This is easy with the menu "Check for extension updates". Just click on the extension name to install the update – but be sure to read the comments and update information in the manual. Did we mention that it is a good idea to have a recent backup?



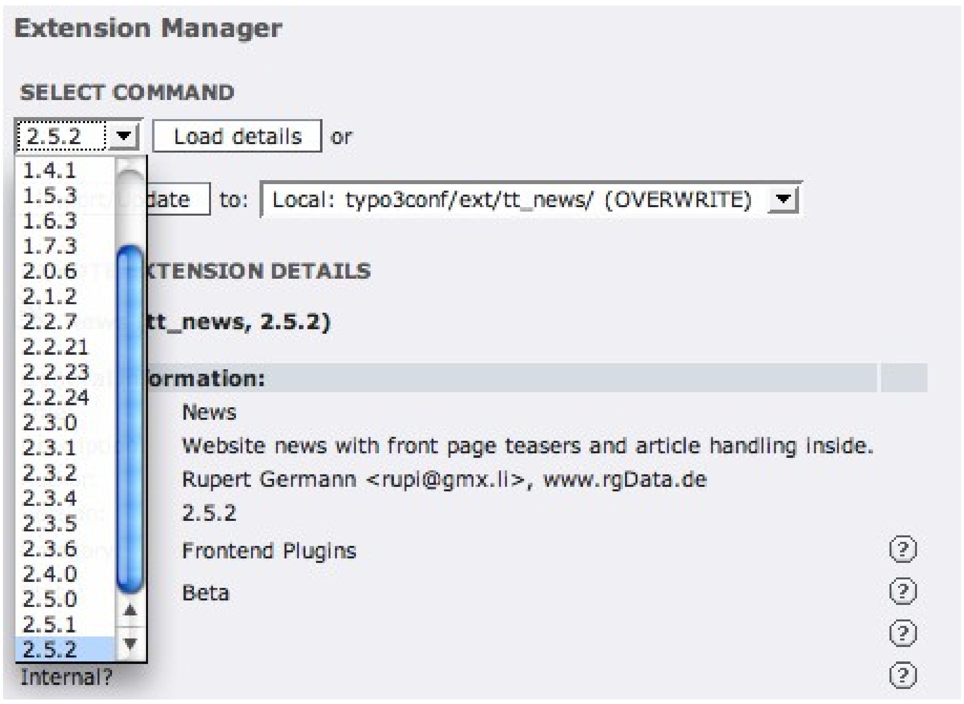
Extension Manager

Display shy extensions: ✓

THERE ARE NEW VERSIONS OF THE FOLLOWING EXTENSIONS IN THE TER

	Extension	Ext-Key	Local	Remote	Location	Upload-Comment	
	phpMyAdmin	phpmyadmin	3.3.0	4.0.1	Local	3.4.0	Security update. Please update and check security bulletin: TYPO3-20080924-1 for details.
						4.0.0	Updated pMA to the latest stable release (3.0.0). Now requires PHP 5.2 and TYPO3 4.2. Old versions (3.x) may be obtained at http://www.mehrwert.de/go/t3x
						4.0.1	Applied patch for redirect bug 8884. Attention: The 4.x release requires PHP 5.2, TYPO3 4.2 and MySQL 5. The 3.x branch is furthermore supported and updated: http://www.mehrwert.de/go/t3x

Sometimes it may be necessary to obtain an older version of an extension. In the default view, the extension manager only offers the newest version for download. But clicking on the name of the extension rather than the download/install icon on the left offers the older versions:



News
Website news with front page teasers and article handling inside.
Rupert Germann <rupi@gmx.li>, www.rgData.de
2.5.2
Frontend Plugins
Beta

Templates

HTML templates are used as the basis for any TYPO3 project. The template defines the overall structure of the website and consists of regular HTML code. Depending on the project, one or more templates may be used for the different sections of a website.

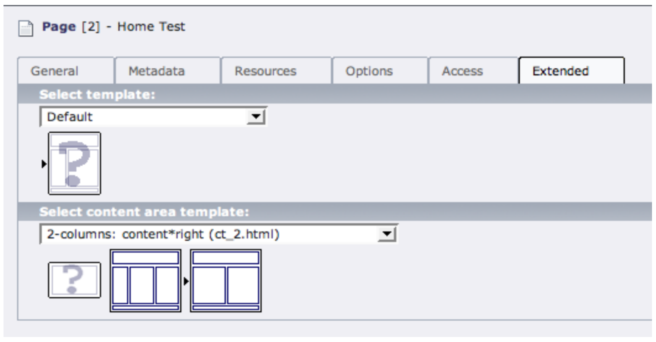
The template contains placeholders which are used to mark the insertion points for the dynamic content. For each dynamic element of the website (menus, content, language selection, ...) a placeholder is required. Before delivering a page to the browser, TYPO3 reads and analyzes the template file, reads the dynamic content from the database and inserts it into the placeholders (or they are replaced with the dynamic elements).

There are two general methods to create and use the templates in TYPO3, which are discussed below.

Note: it is also technically possible to create a project in TYPO3 with TypoScript alone – not using any template file. However this method is obsolete and not covered in this document.

Modern Template Building

With the extensions "rlmp_tmplselector" and "automaketemplate" a quite large range of layout varieties for your website can be realized. The BE user can choose between the different layouts in the page properties.



Select template: Default

Select content area template: 2-columns: content*right (ct_2.html)

The extension automaketemplate converts according to the setup below all div tags into subparts.

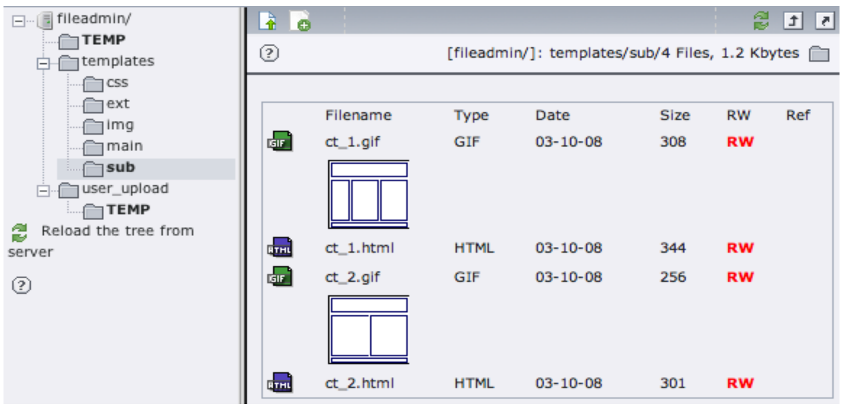
Example:

```
<div id="contentMain"> </div>
```

is converted into:

```
<div id="contentMain"><!-- ###contentMain###--> <!-- ###contentMain###--></div>
```

First you must create a folder structure like this in the folder fileadmin:



```
fileadmin/
  TEMP
  templates
    css
    ext
    img
    main
    sub
  user_upload
    TEMP
Reload the tree from server
```

[fileadmin/]: templates/sub/4 Files, 1.2 Kbytes

Filename	Type	Date	Size	RW	Ref
ct_1.gif	GIF	03-10-08	308	RW	
ct_1.html	HTML	03-10-08	344	RW	
ct_2.gif	GIF	03-10-08	256	RW	
ct_2.html	HTML	03-10-08	301	RW	

The main templates are stored in the folder "fileadmin/template/main/", all sub templates are stored in the folder sub. Save the following templates in the folders.

Main template: template_01.html (wrapper for the sub templates) in folder main

```
<!-- ###DOCUMENT_HEADER### -->
<!-- ###DOCUMENT_HEADER### -->
<!-- ###DOCUMENT_BODY### begin -->
```

```
      <div id="nonFooter">
            <div id="page">
                  <div id="content"> </div>
                  <div id="header"> </div>
                  <div id="topmenu"> </div>
            </div>
      </div>
      <div id="footer"> </div>
<!-- ###DOCUMENT_BODY### end -->
```

Default sub template: ct_1.html in folder sub

```
<title>Standard 3-columns: left*content*right</title>

<div id="ct1">
      <div id="contentWrap">
            <div id="contentMain"> </div>
            <div id="contentRight"> </div>
            <div id="contentBorder"> </div>
      </div>
      <div id="leftmenu"> </div>
</div>
```

TypoScript module for the page layout:

SETUP:

```
plugin.tx_rlmptmplselector_pi1 {
   // Path to the main and sub templates
   templatePathMain = fileadmin/templates/main/
   templatePathSub = fileadmin/templates/sub/

    // default Templates, if none has been chosen
   defaultTemplateFileNameMain = template_01.html
   defaultTemplateFileNameSub = ct_1.html

    // inherit chosen main and sub templates to subpages
    inheritMainTemplates = 1
    inheritSubTemplates = 1
}


## Extension automaketemplate configuration
plugin.tx_automaketemplate_pi1 {
   content < plugin.tx_rlmptmplselector_pi1
   elements {
      BODY.all = 1
      BODY.all.subpartMarker = DOCUMENT_BODY
      HEAD.all = 1
      HEAD.all.subpartMarker = DOCUMENT_HEADER
      HEAD.rmTagSections = title
## enable only for table-based templates (depreciated)
#     TD.all = 1
      DIV.all = 1
   }
   relPathPrefix = fileadmin/templates/main/
}

temp.contentAreaTemplate = TEMPLATE
temp.contentAreaTemplate {
   template =< plugin.tx_automaketemplate_pi1
   template.content.templateType = sub

   workOnSubpart = DOCUMENT_BODY
   subparts.contentMain < styles.content.get
   subparts.contentRight < styles.content.getRight
   subparts.leftmenu < styles.content.getLeft
   subparts.BORDER < styles.content.getBorder
}

temp.mainTemplate = TEMPLATE
temp.mainTemplate {
   template =< plugin.tx_automaketemplate_pi1
   template.content.templateType = main
   workOnSubpart = DOCUMENT_BODY
   subparts.content < temp.contentAreaTemplate
}

temp.headTemplate = TEMPLATE
temp.headTemplate {
   template =< plugin.tx_automaketemplate_pi1
   workOnSubpart = DOCUMENT_HEADER
}

temp.contentAreaHeaderTemplate = TEMPLATE
temp.contentAreaHeaderTemplate {
   template =< plugin.tx_automaketemplate_pi1
   template.content.templateType = sub
   workOnSubpart = DOCUMENT_HEADER
}

page = PAGE
page {
   typeNum = 0
   bodyTag = <body>

   10 < temp.mainTemplate
   headerData.10 < temp.headTemplate
   headerData.20 < temp.contentAreaHeaderTemplate
}
```

That's it!

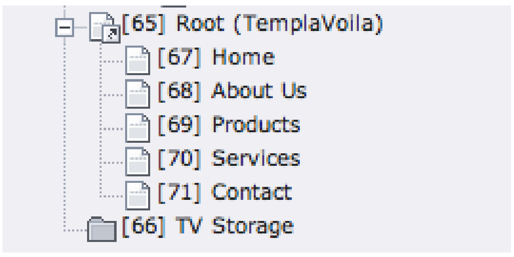
Templavoila (Futuristic Template Building)

TemplaVoila (TV) is a different approach to create a TYPO3 project from an existing HTML template. It is covered in detail in the tutorial 'Futuristic Template Building', however that documentation is somewhat outdated and does not cover the latest version of TV.

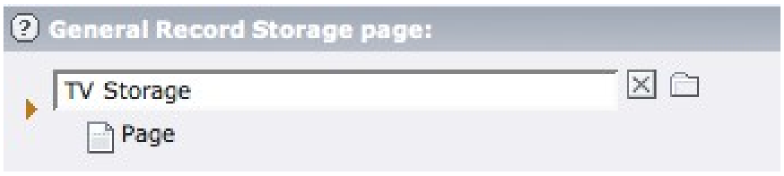
To use TemplaVoila the following requirements must be met:

- Extensions static_info_tables, div, lib and templavoila must be installed
- There must be a system folder in the page tree to hold the TV data structures
- HTML template that passes W3C validation (many problems that occur with TV are caused by templates that do not validate correctly)

In the page tree create a root page that will be the parent of all the pages in the website. Create subpages that reflect the structure of the website:



In the page properties of the root page set the TV Storage folder in the 'Gerneral Record Storage Page':



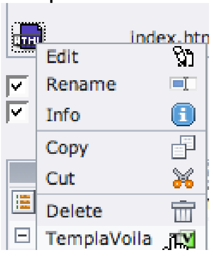
This is the initial Typoscript code needed:

```
page= PAGE
```

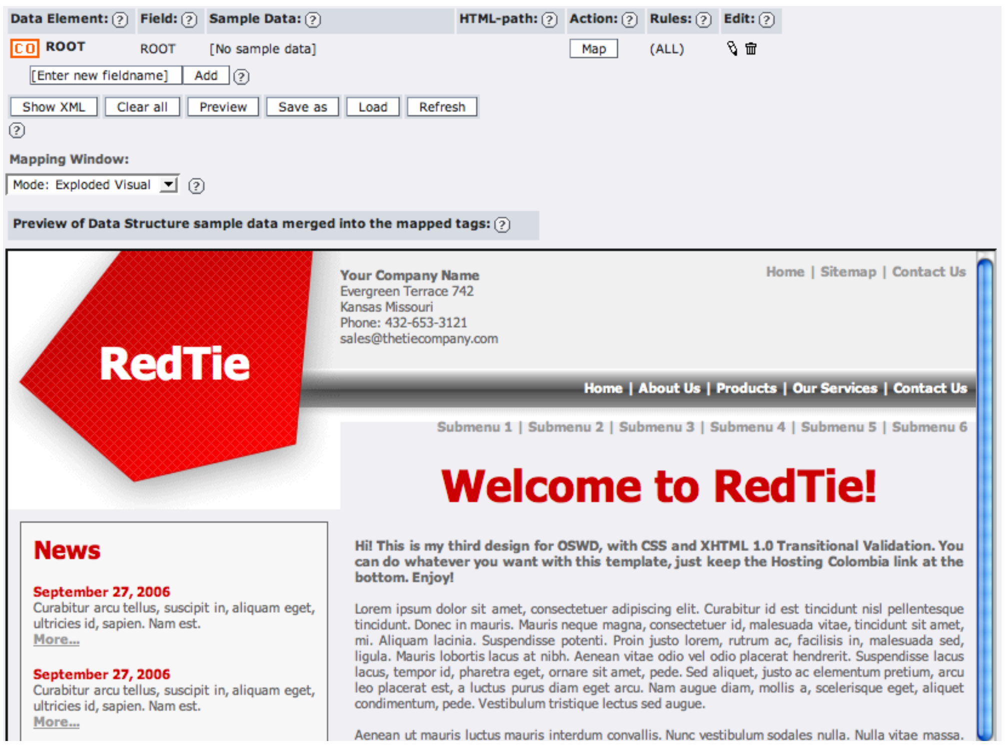
```
page.typeNum = 0
page.10 = USER
page.10.userFunc = tx_templavoila_pi1->main_page
```

Mapping

The key concept of TemplaVoila is the mapping of the template to the elements in TYPO3. Mapping is started through the File->Filelist module. A right-click on the name of the template file offers the option TemplaVoila:



When clicking on the preview button, the template of the website is shown in visual mode:



Your Company Name
Evergreen Terrace 742
Kansas Missouri
Phone: 432-653-3121
sales@thetiecompany.com

Welcome to RedTie!

Hi! This is my third design for OSWD, with CSS and XHTML 1.0 Transitional Validation. You can do whatever you want with this template, just keep the Hosting Colombia link at the bottom. Enjoy!

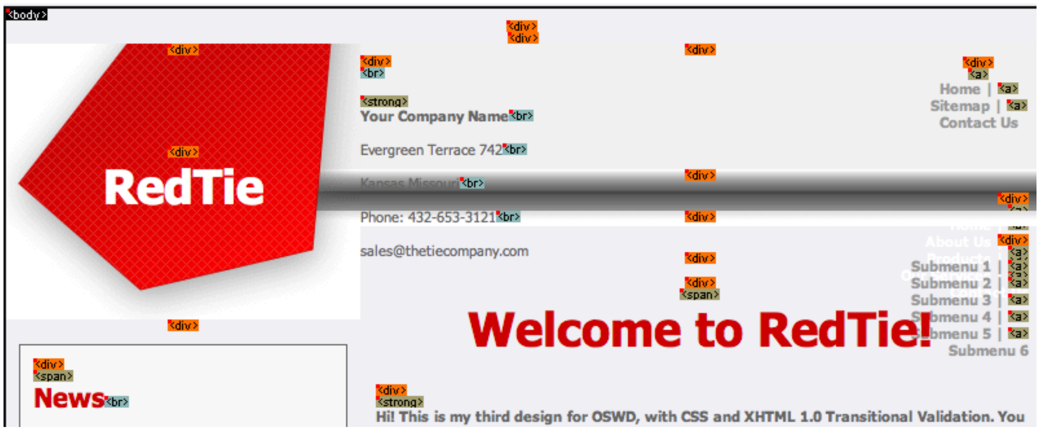
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Curabitur id est tincidunt nisl pellentesque tincidunt. Donec in mauris. Mauris neque magna, consectetuer id, malesuada vitae, tincidunt sit amet, mi. Aliquam lacinia. Suspendisse potenti. Proin justo lorem, rutrum ac, facilisis in, malesuada sed, ligula. Mauris lobortis lacus at nibh. Aenean vitae odio vel odio placerat hendrerit. Suspendisse lacus lacus, tempor id, pharetra eget, ornare sit amet, pede. Sed aliquet, justo ac elementum pretium, arcu leo placerat est, a luctus purus diam eget arcu. Nam augue diam, mollis a, scelerisque eget, aliquet condimentum, pede. Vestibulum tristique lectus sed augue.

Aenean ut mauris luctus mauris interdum convallis. Nunc vestibulum sodales nulla. Nulla vitae massa.

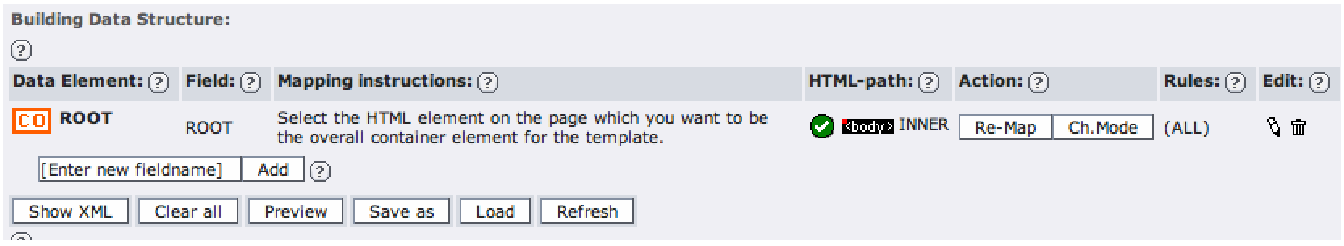
September 27, 2006
Curabitur arcu tellus, suscipit in, aliquam eget, ultricies id, sapien. Nam est.

September 27, 2006
Curabitur arcu tellus, suscipit in, aliquam eget, ultricies id, sapien. Nam est.

In the first step, a root container must be mapped to the body-tag of the template. By clicking on the map button, the HTML tags in the template are displayed in the preview:



The root container is mapped by clicking on the <body> tag. In the Action column we can select between INNER and OUTER mapping. Choosing INNER, the elements inside the tag are mapped, leaving the HTML tag intact. With OUTER mapping, the content between the HTML tag and the tag itself are replaced. Outer mapping is necessary if not only the content but also the tags are generated dynamically. For the root container INNER mapping must be selected, since the <body> tag needs to remain:



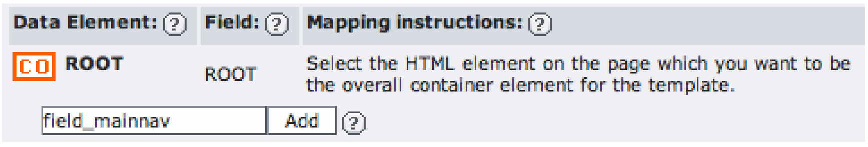
Building Data Structure:

Data Element:	Field:	Mapping instructions:	HTML-path:	Action:	Rules:	Edit:
CO ROOT	ROOT	Select the HTML element on the page which you want to be the overall container element for the template.	<body> INNER	Re-Map Ch.Mode	(ALL)	

[Enter new fieldname] Add

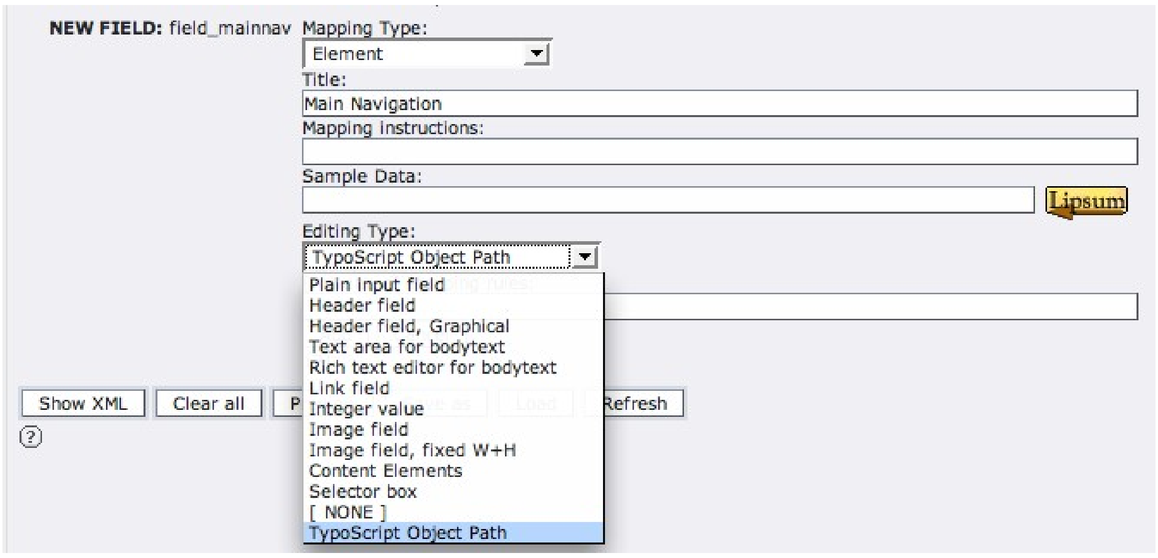
Show XML | Clear all | Preview | Save as | Load | Refresh

By mapping the <body> tag, a container is created that will contain the elements of the webpage. To create the elements, a fieldname must be added. The name always starts with field_ and must be unique. Only letters a-z and underscore are allowed. The first element will be the main navigation of the website:



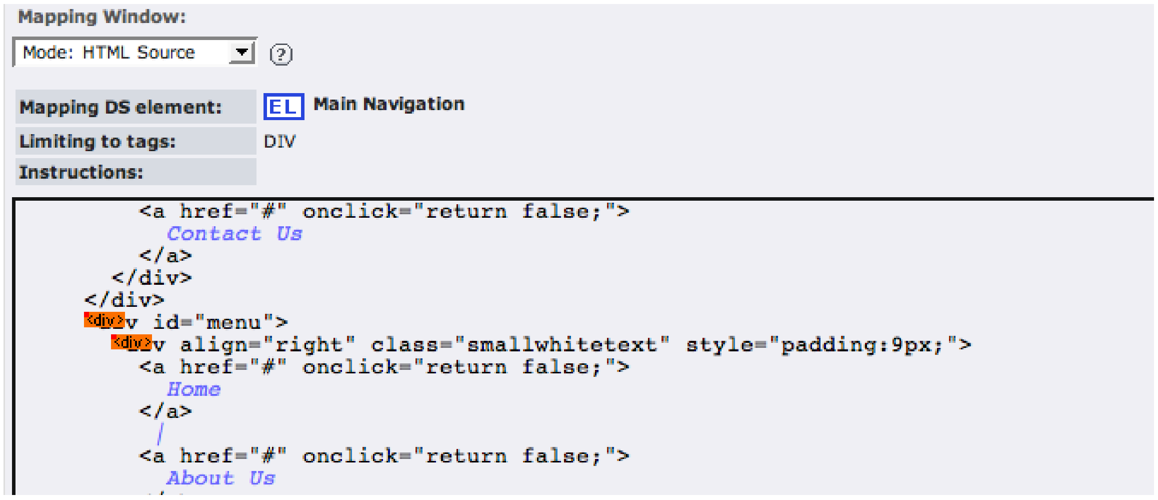
Data Element:	Field:	Mapping instructions:
CO ROOT	ROOT	Select the HTML element on the page which you want to be the overall container element for the template.

field_mainnav Add



NEW FIELD: field_mainnav
Mapping Type: Element
Title: Main Navigation
Mapping instructions:
Sample Data:
Editing Type: TypoScript Object Path

Depending on the layout and implementation of the template, it may sometime be easier to map in the HTML source view:

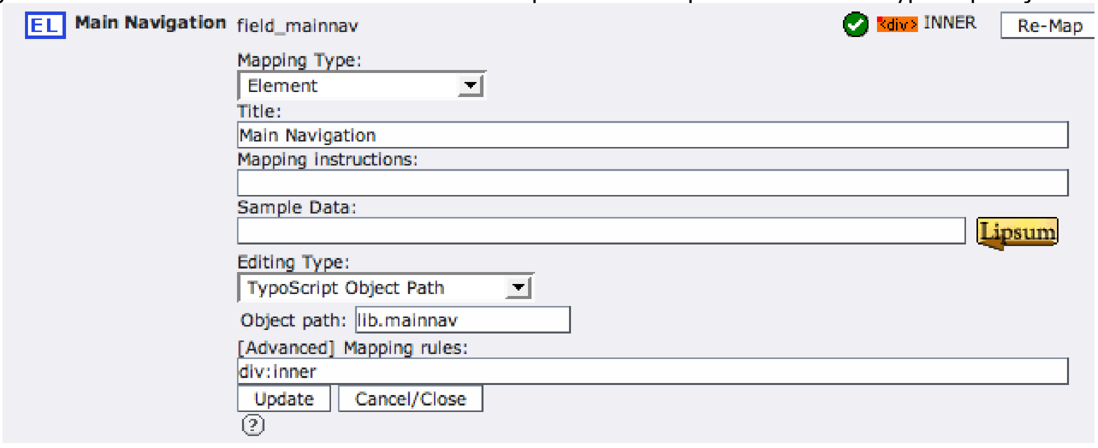


Mapping Window:
Mode: HTML Source

Mapping DS element: EL Main Navigation
Limiting to tags: DIV
Instructions:

```
          <a href="#" onclick="return false;">
            Contact Us
          </a>
        </div>
      </div>
      <div>v id="menu">
        <div>v align="right" class="smallwhitetext" style="padding:9px;">
          <a href="#" onclick="return false;">
            Home
          </a>
            |
          <a href="#" onclick="return false;">
            About Us
```

After saving we can click on the edit icon and now have the option to add a path to our future Typoscript object:



EL **Main Navigation** field_mainnav <div> INNER Re-Map

Mapping Type: Element

Title: Main Navigation

Mapping instructions:

Sample Data:

Editing Type: TypoScript Object Path

Object path: lib.mainnav

[Advanced] Mapping rules: div:inner

TypoScript: Standard Configuration

The first module you should insert beside the page layout settings is the TypoScript module with the standard configuration.

```
## Forbid IE to work in Quirks-Mode
[browser = msie]
config.doctypeSwitch = 1
[GLOBAL]

config {

## XHTML Settings ##
## Set DOCtype
      doctype = xhtml_trans
## Clean XHTML Code
      xhtml_cleaning = all
## Removes comments around content elements
      disablePrefixComment = 1
## Hide <xml...> tag instead of doctypeSwitch
      #config.xmlprologue = none

## Language configuration ##
      lang = de
      language = de
      locale_all = de_DE
      htmlTag_langKey = de

#### IMPORTANT: baseURL must be updated for RealUrl: ####
      baseURL = http://test.riona.de/
      tx_realurl_enable = 1
      prefixLocalAnchors = all

## Activate Admin-Panel
      admPanel = 0

## Extra Debug-Info as comment in HTML code.
## Should be deactivated after going online!
      debug = 1

## Span protection, encode email-address, exchanging @ to '(at)':
      spamProtectEmailAddresses = 1
      spamProtectEmailAddresses_atSubst = &nbsp;(at)&nbsp;

## Configuration for simulateStaticDocuments, deactivate it when using RealURL
      simulateStaticDocuments = 1
      simulateStaticDocuments_noTypeIfNoTitle = 1
      simulateStaticDocuments_pEnc = md5
      simulateStaticDocuments_pEnc_onlyP = L

## RealUrl configuration, deactivate simulateStaticDocuments
      simulateStaticDocuments = 0

## AWSTAT ##
## Create logfile for AWSTATS, count website access
      stat_apache = 1
      stat_apache_logfile = logfile.log
      stat_excludeBEuserHits = 1
## Save statistics in database. Attention: database can grow very big when activating!
#     stat_mysql = 1

## Save klicks on external links in table sys_stat
      jumpurl = 1

## Enable indexedsearch also for extern files (pdf, doc, etc.)
      index_enable = 1
      index_externals = 1

## Save frontend user access
      tx_loginusertrack_enable = 1
}


## Allow HTML tags in headers
lib.stdheader.10.setCurrent.htmlSpecialChars = 0
```

Digital Asset Management (DAM)

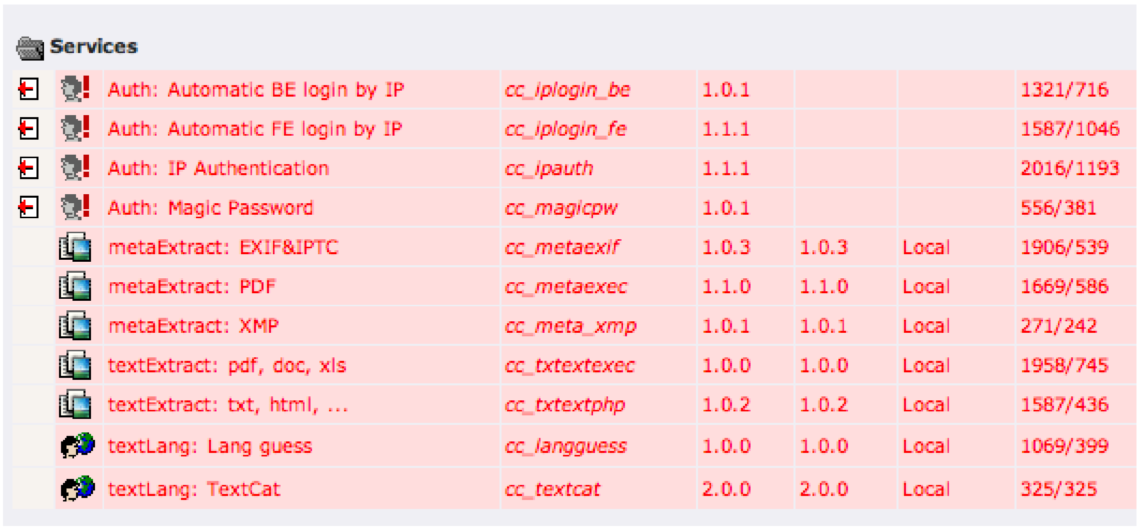
If a project contains 100s or 1,000s of images and other media files (PDF, audio, video, text, ...) it is much better to store information about these files in a media database, rather than keeping these files just in a number of directories.

The Digital Asset Management (DAM) extension for TYPO3 stores information and meta data about these files in the database. Editors can search the database and find related files.

When installing DAM, an option is offered to disable the traditional File->Filelist module. We recommend to not disable this module in general, but disable it for normal editors. Using the File->Filelist module it is possible to edit text files in the backend. This is not possible with the Media->List module, since that can only used to edit the meta data of a file, but not the content.

DAM offers the possibility to automatically extract data from the media files, for example EXIF tags from digital cameras or the text content from Word or PDF files. This data can also be searched when an editor looks for specific media files.

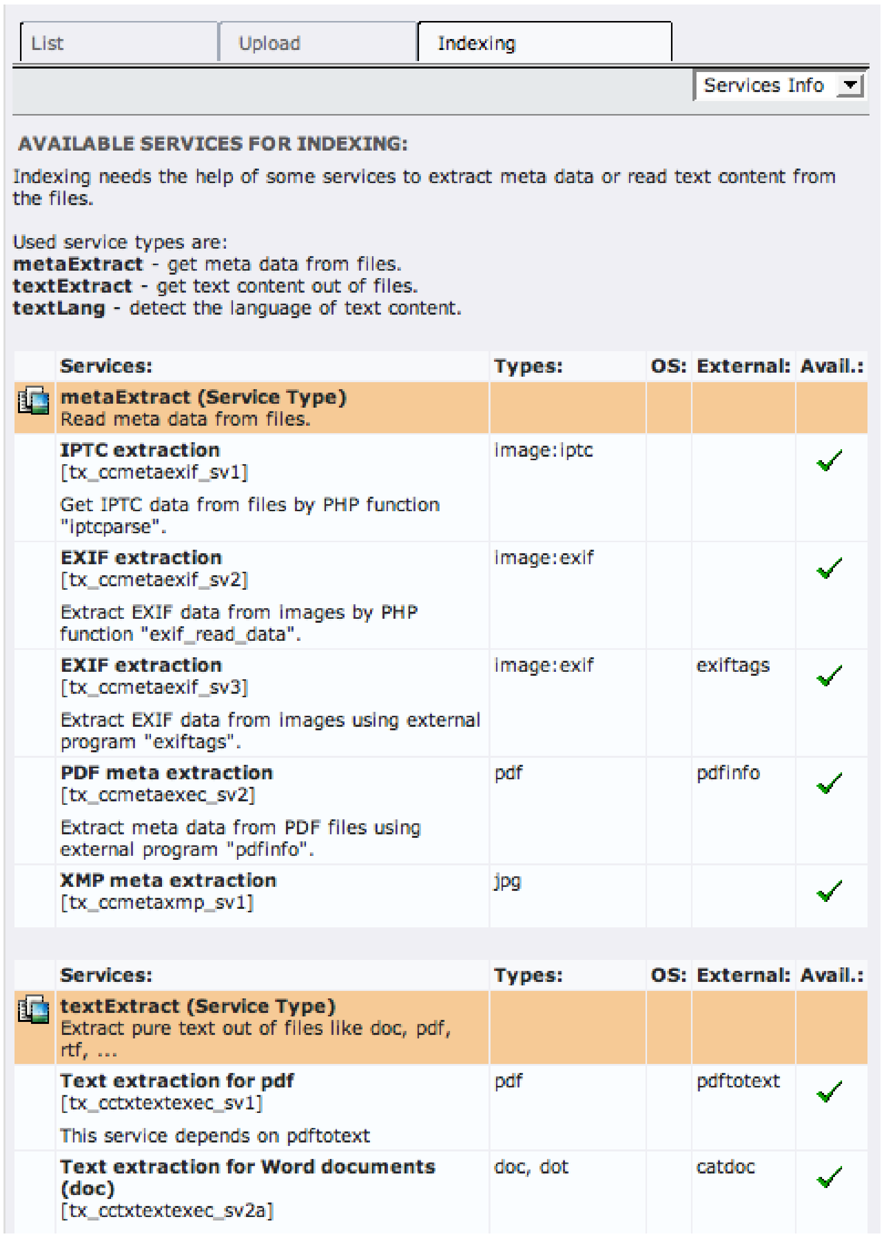
In order that these extract-functions work, a number of additional programs need to be installed on the web server and included in the binPath in the Install Tool. Also a number of additional service-extensions need to be installed:



Services

Auth: Automatic BE login by IP	cc_iplogin_be	1.0.1			1321/716
Auth: Automatic FE login by IP	cc_iplogin_fe	1.1.1			1587/1046
Auth: IP Authentication	cc_ipauth	1.1.1			2016/1193
Auth: Magic Password	cc_magicpw	1.0.1			556/381
metaExtract: EXIF&IPTC	cc_metaexif	1.0.3	1.0.3	Local	1906/539
metaExtract: PDF	cc_metaexec	1.1.0	1.1.0	Local	1669/586
metaExtract: XMP	cc_meta_xmp	1.0.1	1.0.1	Local	271/242
textExtract: pdf, doc, xls	cc_txtextexec	1.0.0	1.0.0	Local	1958/745
textExtract: txt, html, ...	cc_txtextphp	1.0.2	1.0.2	Local	1587/436
textLang: Lang guess	cc_langguess	1.0.0	1.0.0	Local	1069/399
textLang: TextCat	cc_textcat	2.0.0	2.0.0	Local	325/325

When media files are uploaded through the element browser, they are automatically indexed and meta data is extracted. However if files are added for example through FTP or the File->Filelist module, they are not added to the database. Using the DAM indexer these files can be added to the database after uploading.



AVAILABLE SERVICES FOR INDEXING:

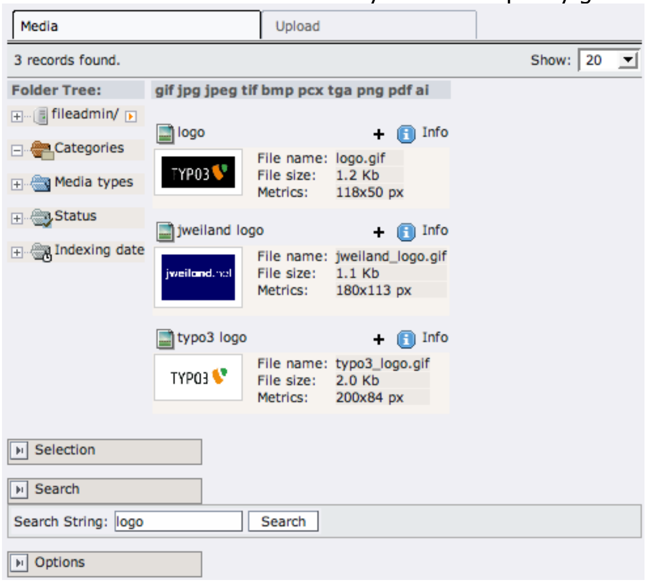
Indexing needs the help of some services to extract meta data or read text content from the files.

Used service types are:
metaExtract - get meta data from files.
textExtract - get text content out of files.
textLang - detect the language of text content.

Services:	Types:	OS:	External:	Avail.:
metaExtract (Service Type) Read meta data from files.				
IPTC extraction [tx_ccmetaexif_sv1] Get IPTC data from files by PHP function "iptcparse".	image:iptc			✓
EXIF extraction [tx_ccmetaexif_sv2] Extract EXIF data from images by PHP function "exif_read_data".	image:exif			✓
EXIF extraction [tx_ccmetaexif_sv3] Extract EXIF data from images using external program "exiftags".	image:exif		exiftags	✓
PDF meta extraction [tx_ccmetaexec_sv2] Extract meta data from PDF files using external program "pdfinfo".	pdf		pdfinfo	✓
XMP meta extraction [tx_ccmetaxmp_sv1]	jpg			✓

Services:	Types:	OS:	External:	Avail.:
textExtract (Service Type) Extract pure text out of files like doc, pdf, rtf, ...				
Text extraction for pdf [tx_cctxtextexec_sv1] This service depends on pdftotext	pdf		pdftotext	✓
Text extraction for Word documents (doc) [tx_cctxtextexec_sv2a]	doc, dot		catdoc	✓

Using the TYPO3 Element Browser the editor can now search for keywords and quickly get a list of all matching files:



3 records found.

logo
File name: logo.gif
File size: 1.2 Kb
Metrics: 118x50 px

jweiland logo
File name: jweiland_logo.gif
File size: 1.1 Kb
Metrics: 180x113 px

typo3 logo
File name: typo3_logo.gif
File size: 2.0 Kb
Metrics: 200x84 px

Search String: logo

Multilanguage Websites

Many websites are multi-lingual: the visitor can select between languages. TYPO3 has extensive support for generating sites in any language.

For optimum results TYPO3 should be configured to use the utf-8 character set. Depending on the language, a single character is represented by 1 to 4 bytes. This not only makes it possible to display character sets like Chinese, but also to mix several different character sets in one page (for example German, Russian and Japanese).

The character set should be set in the install tool before content is entered, since it is quite difficult to convert a project from one encoding to another at a later point in time.

Multiple-Tree Concept

If the page structure and content differs a lot in the various languages, the best way to setup a project will be the multiple-tree concept. Here each language has its own page structure. The language selector will always point to the starting page of the other languages.

Configuration of the multiple-tree concept requires only to set the language-specific Typoscript parameters on the starting page of each page structure.

One-Tree Conept

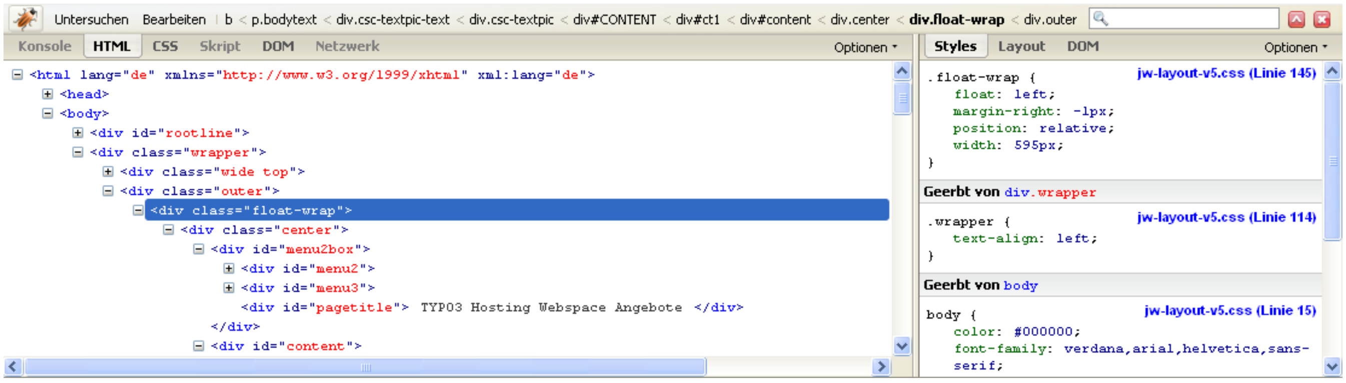
In this configuration it is assumed that the page structure in the various languages is basically identical. For each page there will be a corresponding page in the other language(s). The language selector switches between the different language versions of the same page. If a translation for a specific page is not available either the default language is shown or the language selector will not allow to switch to the other language.

Debugging

Tools

Firefox Add-on: Firebug

Firebug integrates with Firefox. While you browse you can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page.

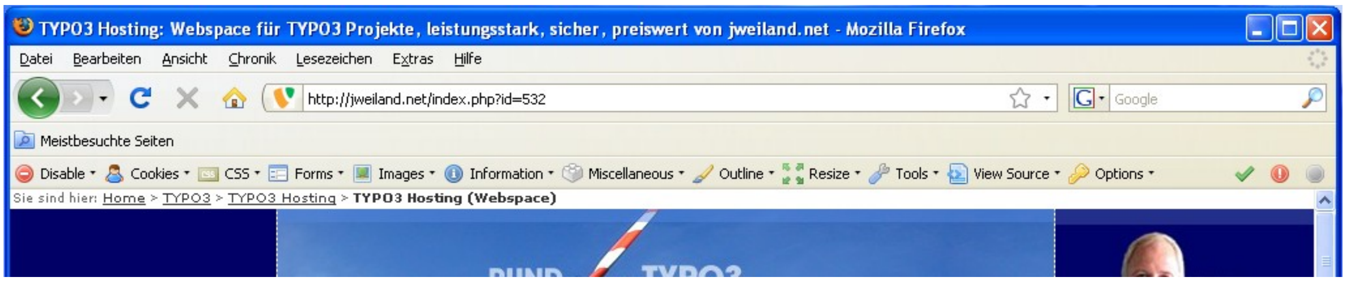

Untersuchen
Bearbeiten
b < p.bodytext < div.csc-textpic-text < div.csc-textpic < div#CONTENT < div#ct1 < div#content < div.center < div.float-wrap < div.outer
Konsole
HTML
CSS
Skript
DOM
Netzwerk
Optionen
Styles
Layout
DOM
Optionen

```
<html lang="de" xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
  <head>
  <body>
    <div id="rootline">
    <div class="wrapper">
      <div class="wide top">
      <div class="outer">
        <div class="float-wrap">
          <div class="center">
            <div id="menu2box">
              <div id="menu2">
              <div id="menu3">
              <div id="pagetitle"> TYPO3 Hosting Webspace Angebote </div>
            </div>
            <div id="content">
```

```
.float-wrap {                                   jw-layout-v5.css (Linie 145)
    float: left;
    margin-right: -1px;
    position: relative;
    width: 595px;
}
Geerbt von div.wrapper
.wrapper {                                      jw-layout-v5.css (Linie 114)
    text-align: left;
}
Geerbt von body
body {                                          jw-layout-v5.css (Linie 15)
    color: #000000;
    font-family: verdana,arial,helvetica,sans-
    serif;
```

Firefox Add-on: Web Developer

Adds a menu and a toolbar with various web developer tools.


TYPO3 Hosting: Webspace für TYPO3 Projekte, leistungsstark, sicher, preiswert von jweiland.net - Mozilla Firefox

http://jweiland.net/index.php?id=532

Sie sind hier: Home > TYPO3 > TYPO3 Hosting > TYPO3 Hosting (Webspace)

Useful Links

Reference

– Possible subsections: Reference (TypoScript)

Property:	**Data type:**	**Description:**	**Default:**
allWrap /+stdWrap	wrap	Wraps the whole item	
wrapItemAndSub	wrap	Wraps the whole item and any submenu concatenated to it.	
subst_elementUid	boolean	If set, all appearances of the string '{elementUid}' in the total element html-code (after wrapped in .allWrap} is substituted with the uid number of the menu item. This is useful if you want to insert an identification code in the HTML in order to manipulate properties with JavaScript.	

RO_chBgColor	string	If property RO is set (see below) then you can set this property to a certain set of parameters which will allow you to change the background color of eg. the tablecell when the mouse rolls over you text-link.
		
		Syntax:
		`[over-color] | [out-color] | [id-prefix]`
		
		Example:
		`page = PAGE`
		`page.typeNum = 0`
		`page.10 = HMENU`
		`page.10.wrap = <table border=1>|</table>`
		`page.10.1 = TMENU`
		`page.10.1.NO {`
		`  allWrap = <tr><td valign=top id="1tmenu{elementUid}" style="background:#eeeeee;">|</td></tr>`
		`  subst_elementUid = 1`
		`  RO_chBgColor = #cccccc | #eeeeee | 1tmenu`
		`  RO = 1`
		`}`
		
		This example will start out with the table cells in #eeeeee and change them to #cccccc (and back) when rolled over. The "1tmenu" string is a unique id for the menu items. You may not need it (unless the same menu items are more than once on a page), but the important thing is that the id of the table cell has the exact same label before the {elementUid} (red marks). The other important thing is that you DO set a default background color for the cell with the style-attribute (blue marking). If you do not, Mozilla browsers will behave a little strange by not capturing the mouseout event the first time it's triggered.	

[tsref:(cObject).TEST]