TYPO3

TSconfig Reference

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Official documentation

This document is included as part of the official TYPO3 documentation. It has been approved by the TYPO3 Documentation Team following a peer-review process. The reader should expect the information in this document to be accurate - please report discrepancies to the Documentation Team (documentation@typo3.org). Official documents are kept up-to-date to the best of the Documentation Team's abilities.

Core Manual

This document is a Core Manual. Core Manuals address the built in functionality of TYPO3 and are designed to provide the reader with in-depth information. Each Core Manual addresses a particular process or function and how it is implemented within the TYPO3 source code. These may include information on available APIs, specific configuration options, etc.

Core Manuals are written as reference manuals. The reader should rely on the Table of Contents to identify what particular section will best address the task at hand.

Table of Contents

Introduction

About this document

TypoScript can be used to create templates for the TYPO3 frontend, but also to configure the TYPO3 backend. In this case, it is called "TSconfig". TSconfig is divided into configuration for pages ("Page TSconfig") and configuration for users and groups ("User TSconfig"). Each variant is further detailed in its own chapter. TSconfig offers vast possibilities of customizing the TYPO3 backend.

For details about the nature of TypoScript as a syntax, please read the document "TypoScript Syntax & In-depth Study".

What's new

This version of the TSconfig Reference was updated for TYPO3 version 4.4.

This document now uses the new template for Core Documentation. Additionally the document was slightly restructured so that it is in line with the other Core Documents. This guarantees one continuous look and feel through the different official papers.

The description of some properties has been added or updated.

More information about changed properties

You can find a list of changes for more recent TYPO3 versions in the wiki:

http://wiki.typo3.org/Documentation_changes_in_4.2

http://wiki.typo3.org/Documentation_changes_in_4.3

http://wiki.typo3.org/Documentation_changes_in_4.4

Credits

This document was orginially written by Kasper Skårhøj. It has since then been maintained successively by Michæl Stucki, François Suter and Christopher Stelmaszyk.

Feedback

For general questions about the documentation get in touch by writing to documentation@typo3.org.

If you find a bug in this manual, please file an issue in the bug tracker of this manual: http://forge.typo3.org/projects/typo3v4-doc_core_tsconfig/issues

Maintaining quality documentation is hard work and the Documentation Team is always looking for volunteers. If you feel like helping please join the documentation mailing list (typo3.projects.documentation on lists.typo3.org).

Conditions

As of TYPO3 4.3 it is possible to use TypoScript conditions in both User TSconfig and Page TSconfig, just as it is done in TypoScript for templates.

Example:

```
[treeLevel = 1]
TCEFORM.tt_content.section_frame.disabled = 1
[GLOBAL]
```

The above TSconfig will hide the "section_frame" field of content elements only on the first level of the page tree.

Differences to conditions in TypoScript templates

There are some slight differences between conditions in TSconfig and conditions in template TypoScript, which must be taken into account:

– Conditions "usergroup" and "loginUser" apply to BE groups and BE users – respectively – and not to FE groups and FE users, quite obviously.

– In the "globalString" condition, key "TSFE:" will not work because the TSFE global object only exists in the FE context. The "LIT:" key will not work either as it is used to compare TypoScript constants, which are not available in the BE context.

– Note that conditions such as "PIDupinRootline" or "treeLevel" will apply correctly to pages that are being created but are not yet saved.

Furthermore the following condition is available **only** in TSconfig:

adminUser

Syntax:

```
[adminUser = 0/1]
```

Comparison

Checks whether the current BE user has admin rights or not. Value is 1 if the user is an admin, 0 if he is not.

Example:

The following condition will apply only if the BE user is an admin.

```
[adminUser = 1]
```

References

For a general discussion about TypoScript conditions, please refer to "TypoScript Syntax and In-depth Study".

For a list of available conditions, please refer to the "TypoScript Reference".

User TSconfig

You can enter TSconfig for both backend users and groups. The resulting TSconfig for a certain backend user is the accumulated TSconfig for all the backend groups he is a member of. Thus you can override formerly set options from the user groups by entering another configuration in the TSconfig field of the user himself.

The "TSconfig" field

This is how the TSconfig for users is entered in the backend user records:

Verifying the final configuration

It's vital to check the resulting configuration of the users. You can do that in the Tools > User Admin module (which is provided by the extension "beuser") by clicking a username. Then you'll see the TSconfig tree among other information. Here's an example:

Overriding previously set options

Now, lets say the user is a member of a *usergroup* with this configuration:

```
TCAdefaults.tt_content {
  hidden = 1
  header = Hello!
}
```

Then setting these values in the TSconfig field of the user himself, would override the default value of the header (marked red) and add the clear cache option (marked blue). The default value of the hidden field is not changed and simply inherited directly from the group:

```
TCAdefaults.tt_content.header = 234
options.clearCache.all = 1
```

Setting default User TSconfig

User TSconfig is designed to be individual for users or groups of users. However it can be very handy to set global values that will be initialized for all users.

In extensions this is easily done by the extension API function, `t3lib_extMgm::addUserTSConfig()`. In the `(ext_)localconf.php` file you can call it like this to set default configuration:

```
/**
 * Adding the admin panel to users by default and forcing the display of the edit-icons
 */
t3lib_extMgm::addUserTSConfig('
admPanel {
  enable.edit = 1
  module.edit.forceNoPopup = 1
  module.edit.forceDisplayFieldIcons = 1
  module.edit.forceDisplayIcons = 0
  hide = 1
}
options.enableShortcuts = 1
');
```

This API function simply adds the content to `$TYPO3_CONF_VARS['BE']['defaultUserTSconfig']`.

Top Level Objects

These are the User TSconfig Top Level Objects (TLOs):

Property:	Data type:	Description:	Default:
admPanel	->ADMPANEL	Options regarding the front-end admin panel	
options	->OPTIONS	Options for the user, various	
mod	*(see ->MOD of Page TSconfig)*	Overriding values for the backend modules **Deprecated**. Use page.mod instead!	
setup.default **setup.override**	->SETUP	Default values and override values for the user settings known from the setup module. **Notice:** There is a tricky aspect to these settings; If first you have set a value by setup.override and then removes it again you will experience that the value persists to exist. This is because it is saved in the backend users profile. Therefore, if you have once set a value, do *not* remove it again by rather set it blank if you want to disable the effect again!	

Property:	Data type:	Description:	Default:
TCAdefaults. [tablename]. [field]	string	Sets default values for records. The order of default values when creating new records in the backend is this: 1. Value from $TCA 2. Value from User TSconfig (these settings) 3. Value from Page TSconfig 4. Value from "defVals" GET vars (see alt_doc.php) 5. Value from previous record based on 'useColumnsForDefaultValues' However the order for default values used by tcemain.php if a certain field is not granted access to for user will be: 1. Value from $TCA 2. Value from User TSconfig (these settings) So these values will be authoritative if the user has no access to the field anyway. **Example**: This sets the default hidden flag for pages to "clear" `TCAdefaults.pages.hidden = 0`	
user		This is for custom purposes. Deprecated, use "tx_*" below from extensions	
auth		Configuration for authentication services. Currently these are the options: **auth.BE.redirectToURL** Specifies a URL to redirect to after login is performed in the backend login form.	
page	all page TSconfig properties	You can override all page TSconfig properties by putting them into user TSconfig and prefixing them with page. **Example**: `page.TCEMAIN.table.pages.disablePrependAtCopy = 1`	
tx_[extension key with no underscore]		This is reserved space for extensions.	

[beuser]

->ADMPANEL

Configuration of the Admin Panel in the Frontend for the user.

Property:	Data type:	Description:	Default:
enable	[object]	Used to enable the various parts of the panel for users. All values are 0/1 booleans. **General**: .all: enables all modules **Modules**: .preview .cache .publish .edit .tsdebug .info	(For admin-users, all = 1 is default! Hardcoded in class)
hide	boolean	If set, the panel will not display in the bottom of the page. This has only a visual effect.	

Property:	Data type:	Description:	Default:
override	[object]	Override all admin panel settings: .[modulename].[propertyname] **Note**: You have to activate a module first by setting .modulename = 1 **Full reference**: To find out the name of a modulename/property, you can have a look at the HTML code of the admin panel and watch the names of the form elements. In this example, the module name is "tsdebug", and the property is called "displayTimes": *name="TSFE_ADMIN_PANEL[tsdebug_displayTimes]"* **Most common options** .preview.showHiddenPages (boolean) .preview.showHiddenRecords (boolean) .preview.simulateDate (timestamp) .preview.simulateUserGroup (integer) .cache.noCache (boolean) .cache.clearCacheLevels (integer) .edit.displayFieldIcons (boolean) .edit.displayIcons (boolean) .edit.editFormsOnPage (boolean) .edit.editNoPopup (boolean)	

module.edit	[object]	**DEPRECATED**, use override.* instead (see above).
		
		.forceDisplayIcons (boolean):
		Forces edit-panels to appear regardless of the selectorbox.
		.forceDisplayFieldIcons (boolean):
		Forces edit-icons to appear regardless of the selectorbox.
		.forceNoPopup (boolean):
		Forces edit-forms to open in same window - not pop up window.
		
		Example, that forces the display of the edit icons without displaying the admin-panel itself:
		
		`admPanel {`
		`  enable.edit = 1`
		`  module.edit.forceDisplayFieldIcons = 1`
		`  hide = 1`
		`}`	

[beuser:admPanel]

> **Tip**
>
> If you want to link to the login-screen of the backend, but wish that the user should return to the frontend for editing, you do that with this link, sending the "redirect_url" parameter to the login-screen. In that case the backend interface is not started.

```
<a href="typo3/index.php?redirect_url=../">
    Backend login returning to frontend
</a>
```

This is how the admin panel looks like. Notice the fact that the visibility of the admin panel is ultimately depending on being configured in your TypoScript template for the website! This is easily done by inserting this string in the TypoScript Template:

```
config.admPanel = 1
```

Or if you use frames, this is probably better:

```
page.config.admPanel = 1
```

->OPTIONS

Various options for the user affecting the core at various points.

Property:	Data type:	Description:	Default:
dontMountAdminMounts	boolean	This options prevents the root to be mounted for an admin user. **NOTE:** Only for admin-users. For other users it has no effect.	
RTEkeyList	[list of keywords]	This is a list of the Rich Text Editor buttons the user may see displayed. The user will not see any buttons not listed here. Either enter a comma list of button keywords (see "TYPO3 Core API / RTE section") or specify all with a wildcard "*" for everything.	* (If value is not set at all, *, is default)
clearCache.pages	boolean	This will allow a user to clear the whole page cache.	
clearCache.all	boolean	This will allow a user to clear all cache (that is everything including templates)	

Property:	Data type:	Description:	Default:
clearCache.clearRTEC ache	boolean	If set, the option «Clear RTE Cache» is enabled in the Clear Cache menu. Note that the option is always available to admin users.	0
lockToIP	string	List of IP-numbers with wildcards. **Note**: This option is enabled only if the TYPO3_CONF_VARS[BE][enabledBeUserIPLock] configuration is true. **Examples**: 192.168.*.* - will allow all from 192.168-network 192.168.*.*, 212.22.33.44 - will allow all from 192.168-network plus all from REMOTE_ADDR 212.22.33.44 192.168, 212.22.33.44 - the same as the previous. Leaving out parts of the IP address is the same as wild cards...	
saveClipboard	boolean	If set, the clipboard content will be preserved for the next login. Normally the clipboard content lasts only during the session.	
clipboardNumberPads	int (0-20)	This allows you to enter how many pads you want on the clipboard.	3
enableShortcuts	boolean	Enables the usage of shortcuts in the backend.	1
shortcutFrame	boolean	If set, the shortcut frame in the bottom of the window appears. This frame contains the shortcuts, the search field and the workspace selector. **Note**: Only takes effect, if alt_main.php, the old backend from TYPO3 4.1, is used. **Note**: This option has been removed in TYPO3 4.4.	

shortcutGroups	Array of integers/string s	Set groups of shortcuts that can be accessed by the user. By default, 5 default groups will be defined globally (shared, can only be set by admins) and also for each user (personal shortcuts):
		
		  1. Pages
		  2. Records
		  3. Files
		  4. Tools
		  5. Miscellaneous
		
		Set 0 to disable one of these group IDs, 1 to enable it (this is the default) or "string" to change the label accordingly.
		
		Example:
		<pre>shortcutGroups {
		  1=1
		  2=My Group
		  3=0
		  4=
		}</pre>
		Shortcut group 1 is loaded with the default label (Pages), group 2 is loaded and labeled as "My Group" and groups 3 and 4 are disabled. Group 5 has not been set, so it will be displayed by default, just like group 1.	

shortcut_onEditId_do ntSetPageTree	boolean	If set, the page tree is *not* opened to the page being edited when an id number is entered in the "Edit Id" box	

Property:	Data type:	Description:	Default:
shortcut_onEditId_keepExistingExpanded	boolean	If set, the existing expanded pages in the page tree are not collapsed when an id is entered in the "Edit Id" box. (provided .shortcut_onEditId_dontSetPageTree is not set!)	
mayNotCreateEditShortcuts	boolean	If set, the user cannot create or edit shortcuts. **Note**: In TYPO3 4.3 and older depends on .shortcutFrame being set.	
createFoldersInEB	boolean	If set, a createFolders option appears in the element browser (for admin-users this is always enabled).	
noThumbsInEB	boolean	If set, then image thumbnails are not shown in the element browser.	
noThumbsInRTEimageSelect	boolean	As .noThumbsInEB but for the Rich Text Editor image selector.	
uploadFieldsInTopOfEB	boolean	If set, the upload-fields in the element browser are put in the top of the window.	

saveDocNew
saveDocNew.[table]
	boolean / "top"	If set, a button "Save and create new" will appear in TCEFORMS.
		Any value set for a single table will override the default value set to the object "saveDocNew".
		
		Example:
		In this example the button is disabled for all tables, except tt_content where it will appear, and in addition create the records in the top of the page (default is after instead of top).
		
		`    options.saveDocNew = 0`
		`    options.saveDocNew.tt_content = top`	

saveDocView **saveDocView.[table]**	boolean	If set, a button "Save and view" will appear in TCEFORMS. Any value set for a single table will override the default value set to the object "saveDocView".	1
disableDelete **disableDelete.[table]**	boolean	Disables the "Delete" button in TCEFORMs. Overriding for single tables works like "saveDocNew" above.	
showHistory **showHistory.[table]**	boolean	Shows link to the history for the record in TCEFORMs. Overriding for single tables works like "saveDocNew" above.	
pageTree.disableIconLinkToContextmenu **folderTree.disableIconLinkToContextmenu**	boolean / "titlelink"	If set, the page/folder-icons in the page/folder tree will not activate the clickmenu. If the value is set "titlelink" then the icon will instead be wrapped with the same link as the title.	
pageTree.disableTitleHighlight	boolean	If set, the page titles in the page tree will not be highlighted when clicked.	
pageTree.showPageIdWithTitle	boolean	If set, the titles in the page navigation tree will have their ID numbers printed before the clickable title.	
pageTree.showDomainNameWithTitle	boolean	If set, the domain name will be appended to the page title for pages that have "Is root of web site?" checked in the page properties. Useful if there are several domains in one page tree.	
pageTree.showNavTitle	boolean	If set, the navigation title is displayed in the page navigation tree instead of the normal page title. The page title is showed in a tooltip if the mouse hovers the navigation title.	
pageTree.onlineWorkspaceInfo	boolean	If set, the workspace info box will also be shown in the page tree even in online mode. Recommended when working with workspaces a lot.	

Property:	Data type:	Description:	Default:
pageTree.hideFilter	boolean	If set, the filter-box in the top of the page tree will be hidden.	
pageTree.separateNotinmenuPages	boolean	If set, not in menu and special pages are separated in the page tree from standard pages.	
pageTree.alphasortNotinmenuPages	boolean	If set and pageTree.seperateNotinmenuPages=1, separated pages are sorted alphabetically.	
pageTree.altElementBrowserMountPoints	list of integers	Sets alternative webmounts for use in the Element Browser. You separate page ids by a comma. Non-existing page ids are ignored. If you insert a non-integer it will evaluate to "0" (zero) and the root of the page tree is mounted. Effective in workspaces too. These alternative webmounts **replace** configured DB Mountpoints. **Example**: `options.pageTree.altElementBrowserMountPoints = 34,123`	
folderTree.altElementBrowserMountPoints	list of foldernames	Sets alternative filemounts for use in the Element Browser. The folders you specify here must exist within the fileadmin/ folder. You separate folders by a comma. If a folder you specify dœs not exist it will not get mounted. Effective in workspaces too. The alternative filemounts are **added** to the existing Filemounts. **Example**: `options.folderTree.altElementBrowserMountPoints = _temp_/, templates`	
folderTree.uploadFieldsInLinkBrowser	int	This value defines the number of upload fields in the element browser. Default value is 3, if set to 0, no upload form will be shown.	3
folderTree.hideCreateFolder	boolean	If set, the user can't create new folders.	false
contextMenu.[key].disableItems	list of items	List of context menu ("clickmenu") items to disable. "key" points to which kind of icon that brings up the menu, and possible options are "pageTree", "pageList", "folderTree", "folderList". "page" and "folder" obviously point to either the Web or File main module. "Tree" and "List" points to whether the menu is activated from the page/folder tree or the listing of records/files. Items to disable are (for "page" type - that is database records): view,edit,hide,new,info,copy,cut,paste,delete,move_wizard, history,perms,new_wizard,hide,edit_access,edit_pageheader, db_list,versioning,moreoptions Items to disable are (for "folder" type - that is files/folders): edit,upload,rename,new,info,copy,cut,paste,delete	
contextMenu.options.leftIcons	boolean	If set, the icons in the clickmenu appear to the left instead of right	
contextMenu.options.clickMenuTimeOut	int, 1-100	Number of seconds the click menu is visible in the top frame before it disappears by it self.	5
contextMenu.options.alwaysShowClickMenuInTopFrame	boolean	If set, then the clickmenu in the top frame is always shown. Default is that it's shown only if the pop-up menus are disabled by user or by browser.	

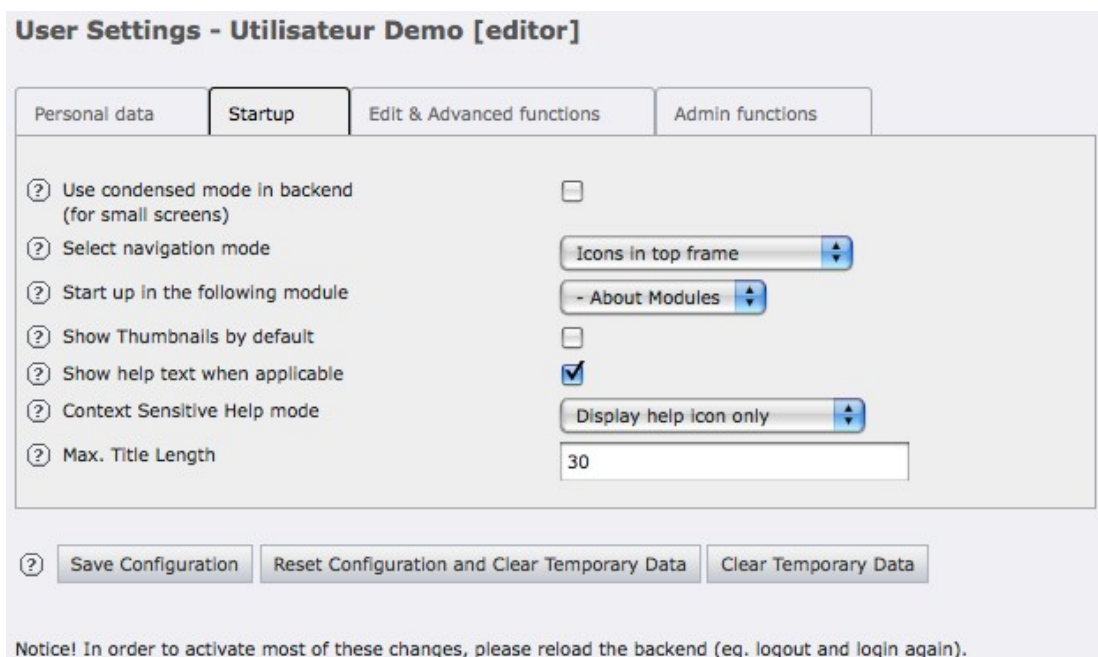| Property:	Data type:	Description:	Default:
overridePageModule	string	By this value you can substitute the default "Web > Page" module key ("web_layout") with another backend module key. **Example**: options.overridePageModule = web_txtemplavoilaM1 This will enable TemplaVoila page module as default page module.	
moduleMenuCollapsable	boolean	If set, the user can collapse main modules in the left menu.	1
alertPopups	bitmask	Configure which Javascript popup alerts have to be displayed and which not: 1 – onTypeChange 2 – copy / move / paste 4 - delete 8 – FE editing 128 - other (not used yet)	255 (show all warnings)
defaultFileUploads	integer	Default number of file upload forms shown in the File->List module	
hideRecords.[table]	list of record ids	This hides records in the backend user interface. It is not an access restriction but makes defined records invisible. That means in principle those records can still be edited if the rights allow. This makes sense if a specialized module should be used only to edit those records. This option is currently implemented for pages only and has an effect in following places: - Page tree navigation frame - Web > List module - New record wizard **Example**: options.hideRecords.pages = 12,45	
workspaces.previewLinkTTLHours	integer	Number of hours for expiry of preview links to workspaces. Default is 48 hours.	
workspaces.swapMode	string	Possible values are: "any" - if page or element (meaning any record on the page) is published, all content elements on the page and page itself will be published regardless of the current editing stage. "page" - if page is published, all content elements on the page will be published as well. If element is published, its publishing dœs not affect other elements or page.	normal behavior (same as in 4.0)
workspaces.changeStageMode	string	Possible values are: "any" - if page or element (meaning any record on the page) stage is changed (for example, from "editing" to "review"), all content elements on the page and page will change to that new stage as well (possibly bypassing intermediate stages). "page" - if page stage is changed (for example, from "editing" to "review"), all content elements on the page will change stage as well (possibly bypassing intermediate stages). If stage is changed for element, all other elements on the page and page itself remain in the previous stage.	normal behavior (same as in 4.0)

Property:	Data type:	Description:	Default:
workspaces.allowed_languages.[workspaceId]	list of sys_language ids	This is a list of sys_language uids which will be allowed in a workspace. This list - if set - will override the allowed languages list in the BE usergroup configuration.	
additionalPreviewLanguages	list of sys_language ids	The user will see these additional languages when localizing stuff in TCEforms. The list are uid numbers of sys_language records.	
checkPageLanguageOverlay	boolean	If set, localized fields in flexforms are shown only for languages which the current page is translated to.	
view.languageOrder	list of sys_language ids	When a backend user clicks a view-page link in the backend (magnifying glass) the first language uid in this list which the user has access to edit will be added as the parameter "&L=[UID]" to the view-link. This is a useful setting for translators which primarily wish to see their translation when they click the view-links. **Example**: `options.view.languageOrder = 2,1`	

[beuser:options]

->SETUP

Default values and overriding values for the "User tools > User settings" module.

Notice: The "User tools > User settings" module only represents a subset of the options from the table below.

Default values are set by 'setup.default' while overriding values are set by 'setup.override'. Overriding values will be impossible for the user to change himself and no matter the current value the overriding value will overrule. The default values are used for new users or if the setup is re-initialized.

NOTICE: If you have first set a value (by override e.g.) and then REMOVE that value from being set, the value is NOT restored to the original default but is kept at the current value! Therefore setting a value and later removing that value would require the users preferences to be reset (you can do that from the Install Tool > Database Analyser > Reset user preferences) OR better, don't remove the value, just change the value of it! (e.g. to a blank string if you wish to "reset" the value).

This table shows the keys for both defaults and override values:

Property:	Data type:	Description:	Default:
thumbnailsByDefault	boolean	Show Thumbnails by default	
emailMeAtLogin	boolean	Notify me by email, when somebody logs in from my account	
startInTaskCenter	boolean	If set, then the backend will start up in the task center (task center should be enabled for the user). This is an old property. Rather look at startModule below instead, as it provides more flexibility.	
helpText	boolean	Show help text when applicable	
titleLen	int+	Max. Title Length	
edit_wideDocument	boolean	Wide document background	
edit_RTE	boolean	Enable Rich Text Editor	
edit_docModuleUpload	boolean	File upload directly in Doc. module	
edit_showFieldHelp	string	Keywords: "" (empty string), "icon" or "text" Determines the type of help text mode for TCA form fields.	
navFrameWidth	int+	The width in pixels of the navigation frame in the Page and File main modules	245 pixels
navFrameResizable	boolean	If set, the frameset modules will have the border between the navigation and list frame resizable.	
lang	language-key	One of the language-keys. See t3lib/config_default.php for current options. E.g. "dk", "de", "es" etc.	
copyLevels	int+	Recursive Copy: Enter the number of page sublevels to include, when a page is copied	
recursiveDelete	boolean	Recursive Delete(!): Allow ALL subpages to be deleted when deleting a page	
allSaveFunctions	boolean	Display all save functions in Doc-module menu	
neverHideAtCopy	boolean	If set, then the hideAtCopy feature for records in TCE will not be used.	
condensedMode	boolean	If set, the backend will not load the Web-submodules and File-submodules in a frameset but allow the page and folder trees to load the submodule in its own frame. This allows for a better display on small screens.	
startModule	string	Name of the module that is called when the user logs into the Backend	
noMenuMode	boolean / string	If set, the backend will not load the left menu frame but rather put a selector-box menu in the topframe. This saves a lot of space on small screens. Also icons will not be displayed in the clickmenu panel in the top. **Value "icons":** Setting noMenuMode to "icons" will still remove the menu, but instead of the selectorbox menu you will have the whole clickmenu panel as a menu with the icons only as the hidden state of the clickmenu panel. This is extremely nice (in my opinion) for experienced users who know the icons of the modules.	
classicPageEditMode	boolean	Setting this option will not open the Web>Page module but rather load the content elements (normal column/default language) together with the page header in one big form when a page is edited (clicking a page icon in the page tree). This simulates the old behaviour in Classic Backend	
hideSubmoduleIcons	boolean	If set then submodule icons will not be shown in the left menu of the backend.	
dontShowPalettesOnFocusInAB	boolean	If set, palettes are not activated in the TCEFORMs when focus is moved to a field.	

Property:	Data type:	Description:	Default:
disableCMlayers	boolean	Disable the context menu layers in the backend.	
disableTabInTextarea	boolean	If you are using IE or Mozilla, TYPO3 will load a little JavaScript file that makes it possible to use the <tab> key in textareas. If you don't like the feature for some reason, you can disable it here.	
resizeTextareas	boolean	This option makes textareas resizable. When moving towards the right or bottom border of the textarea, the mouse cursor changes to a resize cursor. This is active by default.	1
resizeTextareas_Max Height	int+	Defines the maximal height of textarea (in pixels).	600
resizeTextareas_Flexible	boolean	This option makes textareas flexible, which means that their height grows automatically while typing. Limit is the maximal height set. This is active by default.	1
enableFlashUploader	boolean	This option enables the Flash-based uploader in the File module, which allows to select multiple files at once when uploading files. It requires to have the Flash plugin installed (Flash 9 or higher).	

[beuser:setup.default/setup.override]

Do not use any other properties than the ones listed in the table above.
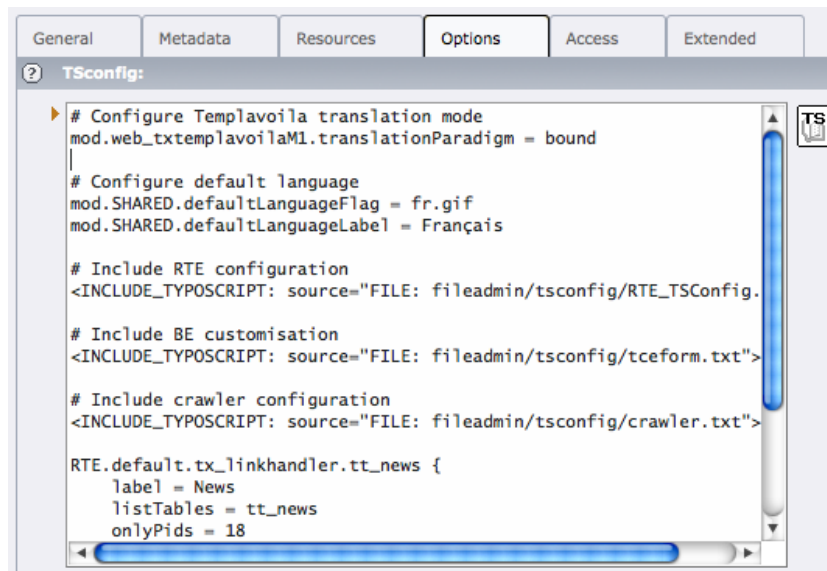
Page TSconfig

The Page TSconfig primarily concerns configuration of the modules in the TYPO3 backend. It may also relate to values in the frontend display occasionally.

The Page TSconfig for a page is accumulated from the root and extends to cover the whole branch of subpages as well (unless values are overridden further out).

The "TSconfig" field

This is an example of the TSconfig field with a snippet of configuration for the Rich Text Editor. Precisely the Rich Text Editor is quite a good example of the usefulness of 'Page TSconfig'. The reason is that you may need the RTE to work differently in different parts of the website. For instance you might need to offer other style-classes in certain parts of the website. Or some options might need to be removed in other parts. The 'Page TSconfig' is used to configure this.

The "TSconfig" field here is available in the page headers:

Verifying the final configuration

If you need to check out the actual configuration for a certain branch in the website, use the 'Web > Info' module (provided by the extension "info_pagetsconfig"):

17
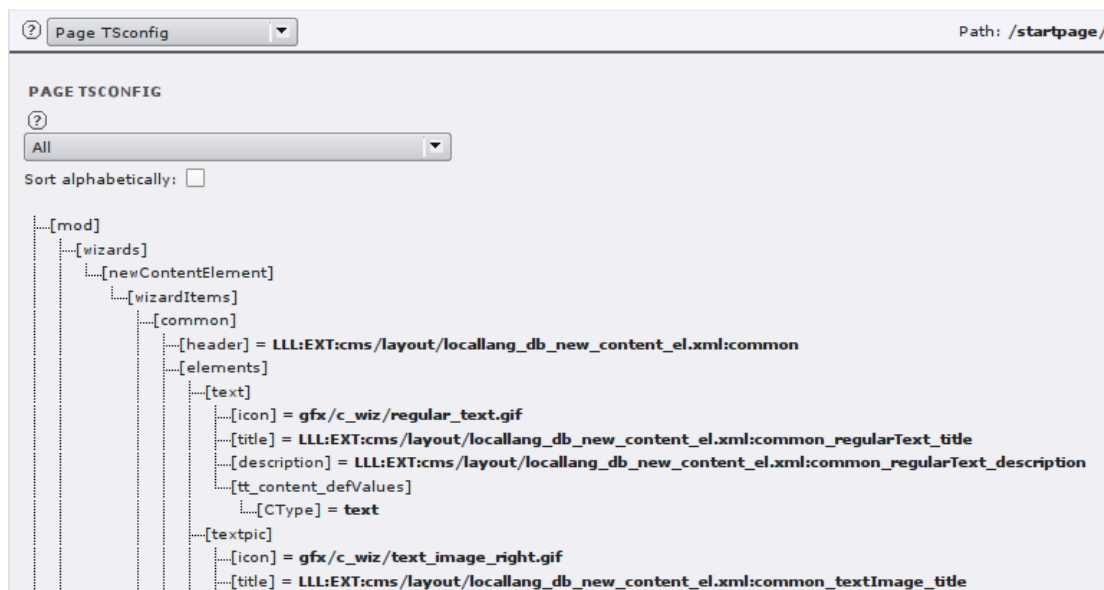
Setting default Page TSconfig

Page TSconfig is designed to be individual for branches of the page tree. However it can be very handy to set global values that will be initialized from the root of the tree.

In extensions this is easily done by the extension API function, `t3lib_extMgm::addPageTSConfig()`. In the `(ext_)localconf.php` file you can call it like this to set default configuration:

```
t3lib_extMgm::addPageTSConfig('
    RTE.default {
        proc.preserveTables = 1
        showButtons = cut,copy,paste,fontstyle,fontsize,textcolor
        hideButtons = class,user,chMode
    }
');
```

This API function simply adds the content to `$TYPO3_CONF_VARS['BE']['defaultPageTSconfig']`.

Top Level Objects

These are the Page TSconfig Top Level Objects (TLOs):

Property:	Data type:	Description:	Default:
mod	->MOD	Options for the backend modules. *Notice that these options are merged with settings from User TSconfig (TLO: mod) which takes precedence.*	
RTE	->RTE	This defines configuration for the Rich Text Editor. Please refer to the document "TYPO3 Core API" which you find in the section Core Documentation on typo3.org. There the chapter "RTE API" contains more information.	
TCEMAIN	->TCEMAIN	Configuration for the TYPO3 Core Engine (TCEmain)	
TCEFORM	->TCEFORM	Extra configuration for the form fields rendered by the TCEforms-class in general.	
TSFE	->TSFE	Options for the TSFE front end object.	
user		This is for custom purposes. Deprecated, use "tx_*" below from extensions	
tx_[extension key with no underscore]		This is reserved space for extensions.	

[page]

->MOD

Configuration for backend modules. Generally the syntax is *[module_name].[property]*. The module name is defined in the `conf.php` files for the module.

Blinding Function Menu options in Backend Modules

Most of the modules in TYPO3 have a "function menu" selector box and this menu is usually configurable so you are able to remove menu items in specific sections of the page tree (or by overriding via User TSconfig, you could disable an option totally for a specific user/group).
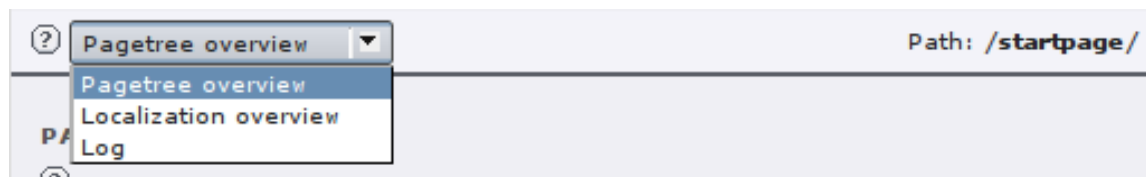
In this case the main menu of the Web > Info module looks like this:

By adding this Page TSconfig we can remove the "Page TSconfig" item:
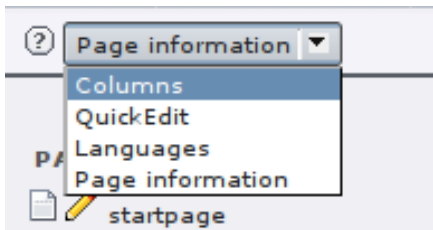
```
mod.web_info.menu.function {
   tx_infopagetsconfig_webinfo = 0
}
```

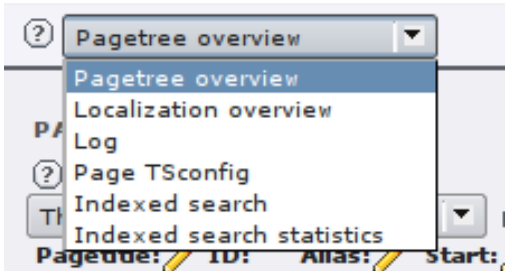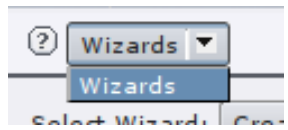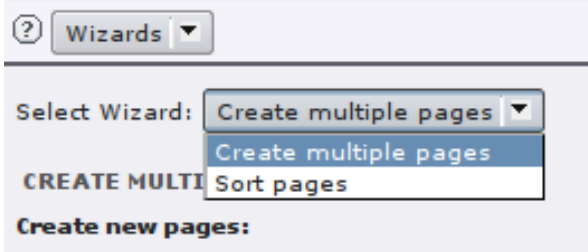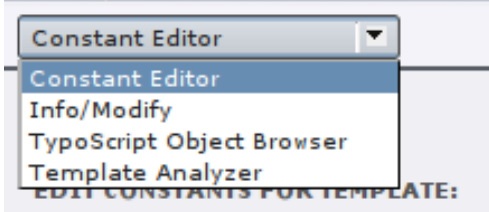The function menu will now look like this:

The 'Page TSconfig' option is simply disabled by setting this Page TSconfig!

All you need to know in order to disable function menu items in the backend modules is, *which* modules support it and what the *key* of the menu item is (in the above example it was 'tx_infopagetsconfig_webinfo'). Modules extending the class "t3lib_SCbase" will most likely provide this out-of-the-box since it's a part of the base class in `t3lib_SCbase::menuConfig()`.

Examples from the TYPO3 core are listed in the table below (Testsite package):

Property:	Description:
web_layout.menu.function	**Web > Page module** **Option tags**: <pre><select name="SET[function]">\n <option value="1">Columns</option>\n <option value="0">QuickEdit</option>\n <option value="2">Languages</option>\n <option value="3">Page information</option>\n</select></pre> **Example**: <pre> # Disables all items except the "QuickEdit" item:\nmod.web_layout.menu.function {\n 1 = 0\n 2 = 0\n 3 = 0\n}</pre>

Property:	Description:

web_info.menu.function	**Web > Info module**
	
	Option tags:
	
	``
	<select name="SET[function]">
	    <option value="tx_cms_webinfo_page">Pagetree
	overview</option>
	    <option value="tx_belog_webinfo">Log</option>
	    <option value="tx_infopagetsconfig_webinfo">Page
	TSconfig</option>
	    <option value="tx_cms_webinfo_hits">Hit Statistics</option>
	    <option value="tx_indexedsearch_modfunc1">Indexed
	search</option>
	</select>
	``
	
	Example:
	
	``
	  # Disables the item "Hit Statistics":
	mod.web_info.menu.function {
	  tx_cms_webinfo_hits = 0
	}
	``

web_func.menu.function	**Web > Functions module** **Option tags:** ``` <select name="SET[function]"> <option value="tx_funcwizards_webfunc">Wizards</option> </select> ```

Property:	Description:

web_func.menu.wiz	**Web > Functions module, Wizards sub module**
	This is the 2nd-level Function Menu in the Web > Functions module. Instead of the "function" key of the main menu it just uses the key "wiz" instead.
	
	Option tags:
	
	``
	<select name="SET[wiz]">
	    <option value="tx_wizardcrpages_webfunc_2">Create multiple pages</option>
	    <option value="tx_wizardsortpages_webfunc_2">Sort pages</option>
	</select>
	``
	Example:
	
	``
	    # Disables the sub-item "Create multiple pages":
	mod.web_func.menu.wiz {
	    tx_wizardcrpages_webfunc_2 = 0
	}
	``

web_ts.menu.function	**Web > Template module** **Option tags**: ``` <select name="SET[function]"> <option value="tx_tstemplateceditor">Constant Editor</option> <option value="tx_tstemplateinfo">Info/Modify</option> <option value="tx_tstemplateobjbrowser">TypoScript Object Browser</option> <option value="tx_tstemplateanalyzer">Template Analyzer</option> <option value="tx_tstemplatestyler_modfunc1">CSS Styler</option> </select> ```

Property:	Description:

user_task.menu.function	**User > Task Center**
	This is slightly different because the Task Center does not provide a selector box function menu. But behind the scene it uses the same functionality of saving "states" and therefore you can also blind items in the Task Center.
	There is one tricky thing though: The Task Center is not depending on a page in the page tree! So you either have to set default Page TSconfig or User TSconfig to blind options here!
	
	TYPO3 taskcenter admin
	
	Export Presets
	
	Recent Pages
	startpage
	List recently edited records
	
	Plugins
	Media
	General User Storage
	News
	
	Quick Note
	Greetings from T3DD09 Coding Night.
	
	Web>List module / root
	
	Tasks
	Inbox: 0 To-Dos
	
	Keys are:
	
	tx_sysnotepad = Quick Note
	tx_taskcenterrecent = Recent Pages
	tx_taskcenterrootlist = Web > List module / root
	tx_taskcentermodules = Plugins
	tx_sysaction = Actions
	tx_systodos = Tasks
	
	Example:
	Set this as *User TSconfig*:
	
	``
	   # Task Center configuration:
	mod.user_task.menu.function {
	   # Disable "Recent Pages" display:
	  tx_taskcenterrecent = 0
	    # Disable "Action" list
	  tx_sysaction = 0
	}
	``

[page:mod; beuser:mod]

Since function menu items can be provided by extensions it is not possible to create a complete list of menu keys. The list above represents a rather typical installation of the TYPO3 Core and a few system extensions.

Therefore, if you want to blind a menu item, the only safe way of doing it, is to look at the HTML

source of the backend module, to find the selector box with the function menu and to extract the key from the <option> tags. This listing is a cleaned-up version of a function menu. The keys are the values of the option tags:

```
<select>
    <option value="tx_cms_webinfo_page">Pagetree overview</option>
    <option value="tx_belog_webinfo">Log</option>
    <option value="tx_infopagetsconfig_webinfo">Page TSconfig</option>
</select>
```

As you can see, this is where the key for the example before was found:

```
mod.web_info.menu.function {
    tx_infopagetsconfig_webinfo = 0
}
```

WARNING:

Blinding Function Menu items is *not* hardcore access control! All it dœs is to hide the possibility of accessing that module functionality from the interface. It might be possible for users to hack their way around it and access the functionality anyways. You should use the option of blinding elements mostly to remove otherwise distracting options.

Overriding Page TSconfig with User TSconfig

In all standard modules the Page TSconfig values of the "mod." branch may be overridden by the same branch of values set for the backend user.
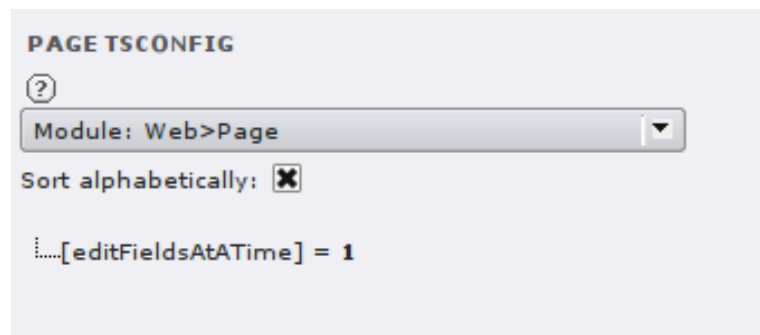
To illustrate this feature let's consider the case from above where a menu item in the Web > Info module was disabled in the Page TSconfig with this value:
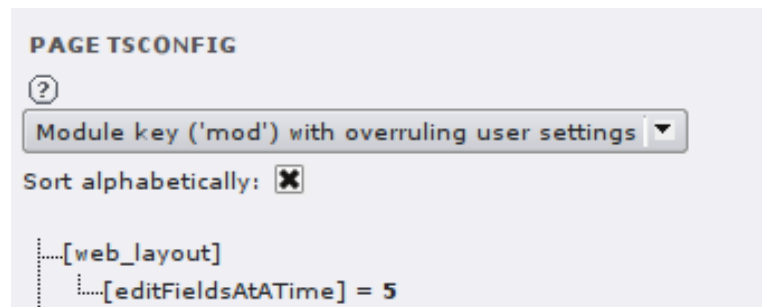
```
mod.web_info.menu.function {
    tsconf = 0
}
```

If however we activate this configuration in the TSconfig of a certain backend user (e.g. the admin user), that user would still be able to select this menu item because the value of his User TSconfig overrides the same value set in the Page TSconfig:

```
mod.web_info.menu.function {
    tsconf = 1
}
```

Here is another example: The value of 'mod.web_layout.editFieldsAtATime' has been set to '1' in Page TSconfig. Additionally it is also set in the User TSconfig of the user, who is currently logged in, but there to the value '5'. The upper image shows you how to check the Page TSconfig. In the lower image you see the result of this user's User TSconfig: It overrides the Page TSconfig and alters the configuration:
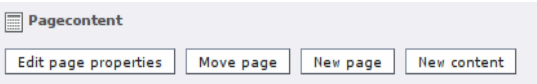
Shared options for modules (mod.SHARED):

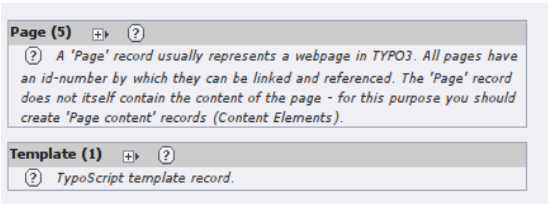| Property:	Data type:	Description:	Default:
colPos_list	(list of integers, blank = don't do anything.)	This option lets you specify which columns of tt_content elements should be displayed in the 'Columns' view of the modules, in particular Web > Page. By default there are four columns, Left, Normal, Right, Border. However most websites use only the Normal column, maybe another also. In that case the remaining columns are not needed. By this option you can specify exactly which of the columns you want to display. Each column has a number which ultimately comes from the configuration of the table tt_content, field 'colPos' found in the tables.php file. This is the values of the four default columns: Left: 1 Normal: 0 Right: 2 Border: 3 **Example**: This results in only the Normal and Border column being displayed: `mod.SHARED.colPos_list = 0,3`	
defaultLanguageLabel	string	Alternative label for "Default" when language labels are shown in the interface. Used on Web > List, Web > Page and TemplaVoilà page module.	
defaultLanguageFlag	string	Flag icon filename from gfx/flags/ for default language Used in Web > List and TemplaVoilà page module.	
disableLanguages	string	Comma-separated list of language UID which will be disabled in the given page tree.	

[page:mod.SHARED; beuser:mod.SHARED]

Web > Page (mod.web_layout):
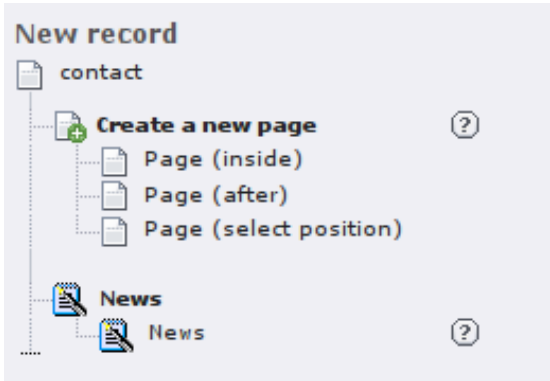
Property:	Data type:	Description:	Default:
tt_content.colPos_list	(list of integers, blank = don't do anything.)	See mod.SHARED.colPos_list for details. If non-blank, this list will override the one set by mod.SHARED.colPos_list. **Example**: This results in only the Normal and Border column being displayed: `mod.web_layout.tt_content.colPos_list = 0,3`	

Property:	Data type:	Description:	Default:

tt_content.fieldOrder	list of field names from tt_content table	This allows you to specify (and thereby overrule) the preferred order of the field names of the "Quick Edit" editing forms of the tt_content table (Content Elements). Just specify the list of fields, separated by comma. Then these fields will be listed first and all remaining fields thereafter in their original order.
		
		Example:
		This results in the 'Text' field and thereafter 'Header' field being display as the very first fields instead of the 'Type' field.
		
		``
		mod.web_layout.tt_content {
		    fieldOrder = bodytext, header
		}
		``	

editFieldsAtATime	int+	Specifies the number of subsequent content elements to load in the edit form when clicking the edit icon of a content element in the 'Columns' view of the module.
		
		Example:
		
		``
		mod.web_layout {
		    editFieldsAtATime = 2
		}
		``
			1

noCreateRecordsLink	boolean	If set, the link in the bottom of the page, "Create new record", is hidden.	
QEisDefault	boolean	If set, then the QuickEditor is the first element in the Function Menu in the top of the menu in Web > Page	
disableSearchBox	boolean	Disables the search box in Columns view.	
disableBigButtons	boolean	Disables the large buttons in top of the Columns view. These are the buttons that are hidden by this option: 	
disableAdvanced	boolean	Disables the clear cache advanced function in the bottom of the page in the module, including the "Create new record" link.	
disableNewContentElementWizard	boolean	Disables the fact that the new-content-element icons links to the content element wizard and not directly to a blank "NEW" form.	
defaultLanguageLabel	string	Alternative label for "Default" when language labels are shown in the interface. Overrides the same property from mod.SHARED if set.	
defLangBinding	boolean	If set, translations of content elements are bound to the default record in the display. This means that within each column with content elements any translation found for exactly the shown default content element will be shown in the language column next to. This display mode should be used depending on how the frontend is configured to display localization. The frontend must display localized pages by selecting the default content elements and for each one overlay with a possible translation if found.	
disableIconToolbar	boolean	Disables the topmost icon toolbar with the "view"-Icon and the icon toolbar below.	

[page:mod.web_layout; beuser:mod.web_layout]

Web > List (mod.web_list):

Property:	Data type:	Description:	Default:

noCreateRecordsLink	boolean	If set, the link in the bottom of the page, "Create new record", is hidden.
		
		Example:
		
		``
		mod.web_list {
		        noCreateRecordsLink = 1
		}
		``	

alternateBgColors	boolean	If set, the background colors of elements will alternate.
		
		Example:
		
		``
		mod.web_list {
		        alternateBgColors = 1
		}
		``
		
		The result is alternating background colors for each element:	

disableSingleTableView	boolean	If set, then the links on the table titles which shows a single table listing only will not be available (including sorting links on columns titles, because these links jumps to the table-only view).	

listOnlyInSingleTableView	boolean	If set, then records will be listed only when the single table view is enabled. This is very practical for pages containing many records from many tables!
		
		Example:
		
		``
		mod.web_list {
		        listOnlyInSingleTableView = 1
		}
		``
		
		The result will be that records from tables are listed only in the single-table mode and the default view will just show number of table records in page and the table description if available:	

itemsLimitSingleTable	int+	Set the default maximum number of items to show in single table view	100
itemsLimitPerTable	int+	Set the default maximum number of items to show per table	20

26

Property:	Data type:	Description:	Default:
noViewWithDokTypes	string	Hide view icon for the defined doktypes (comma-separated)	254,255
hideTables	*list of table names*	Hide these tables in record listings (comma-separated)	

table.[*tablename*].hideTable	boolean	If set to non-zero, the table is hidden. If it is zero, table is shown if table name is not listed in "hideTables" list.
		
		Example:
		
		`mod.web_list.table.tt_content.hideTable`
		`= 1`	

allowedNewTables	*list of tablenames*	If this list is set, then only tables listed here will have a link to "create new" in the page and subpages.
		
		This also affects "db_new.php" (the display of "Create new record")
		Note: Technically records can be created (e.g. by copying/moving), so this is "pseudo security". The point is to reduce the number of options for new records visually.
		
		Example:
		
		`mod.web_list {`
		`    allowedNewTables = pages, tt_news`
		`}`
		
		Only pages and tt_news table elements will be linked to in the New record screen:	

deniedNewTables	*list of tablenames*	If this list is set, then the tables listed here won't have a link to "create news" in the page and subpages. This also affects "db_new.php" (the display of "Create new record").
		
		This is the opposite of the previous property "allowedNewTables".
		If allowedNewTables and deniedNewTables contain a common subset, deniedNewTables takes precedence.
		
		Example:
		
		`mod.web_list {`
		`    deniedNewTables = tt_news,tt_content`
		`}`	

newWizards	boolean	If set, then the new-link over the control panel of the pages and tt_content listings in the List module will link to the wizards and not create a record in the top of the list.	

27

Property:	Data type:	Description:	Default:
showClipControlPanelsDespiteOfCMlayers	boolean	If set, then the control- and clipboard panels of the module is shown even if the context-popups (ClickMenu) are available. Normally the control- and clipboard panels are disabled (unless extended mode is set) in order to save bandwidth.	
newPageWiz.overrideWithExtension newContentWiz.overrideWithExtension	string	If set to an extension key, (eg. "templavoila") then the "mod1/index.php" file of that extension will be used for creating new elements on the page. "newContentWiz" will likewise use the "mod1/db_new_content_el.php" for creating new content elements. Also see "options.overridePageModule"	
clickTitleMode	string	Keyword which defines what happens when a user clicks the title in the list. Default is that pages will go one level down while other records have no link at all. Keywords: **edit** = Edits record **info** = Shows information **show** = Shows page/content element in frontend	

[page:mod.web_list; beuser:mod.web_list]
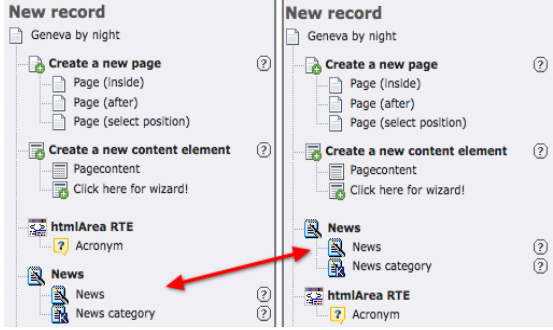
Web > View (mod.web_view)

Property:	Data type:	Description:	Default:

type	int+	Enter the value of the &type parameter passed to the webpage.
		
		Example:
		By this configuration frontend pages will be shown with "index.php?id=123&type=1" from the Web > View module:
		
		``
		mod.web_view {
		        type = 1
		}
		``	

[page:mod.web_view; beuser:mod.web_view]
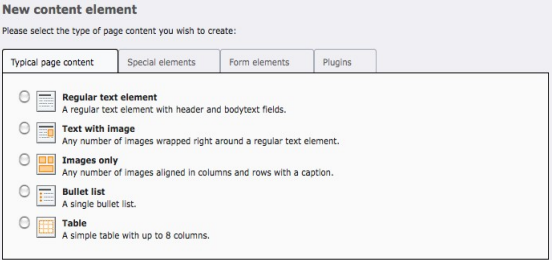
Wizards (mod.wizards)

The configuration for wizards was introduced in TYPO3 4.3. They make it possible to customize the new record wizard or the new content element wizard, for example.

New record wizard (mod.wizards.newRecord)

Property:	Data type:	Description:	Default:
order	*list of values*	Define an alternate order for the groups of records in the new records wizard. Pages and content elements will always be on top, but the order of other record groups can be changed. Records are grouped by extension keys, plus the special key "system" for records provided by the TYPO3 Core. **Example**: `mod.wizards.newRecord.order = tt_news` This places the tt_news group at the top (after pages and content elements). The other groups follow unchanged: 	

[page:mod.wizards.newRecord; beuser:mod.wizards.newRecord]

New content element wizard (mod.wizards.newContentElement)

Property:	Data type:	Description:	Default:
renderMode	string	Alternative rendering mode; set to "tabs", if you want tabs. **Example**: `mod.wizards.newContentElement.renderMode = tabs` 	
wizardItems.[group]	array	In the new content element wizard, content element types are grouped together by type. Each such group can be configured independently. The four default groups are: "common", "special", "forms" and "plugins". The configuration options below apply to any group.	
wizardItems.[group].header	string (localized)	Name of the group	

Property:	Data type:	Description:	Default:

wizardItems.[group].show	string	Comma-separated list of items to show in the group. Use "*" to show all.
		
		Example:
		
		`    # Hide bulletList`
		`mod.wizards.newContentElement.wizardItems`
		`.common.show := removeFromList(bullets)`
		
		`    # Only show text and textpic in common`
		`mod.wizards.newContentElement.wizardItems`
		`.common.show = text,textpic`	

wizardItems. [group].elements	array	List of items in the group	
wizardItems. [group].elements.[name]	array	Configuration for a single item	
wizardItems. [group].elements. [name].icon	resource	Path to the icon	
wizardItems. [group].elements. [name].title	string (localized)	Name of the item	
wizardItems. [group].elements. [name].description	string (localized)	Description text for the item	
wizardItems. [group].elements. [name].tt_content_defValue s	array	Default values for tt_content fields	

[page:mod.wizards.newContentElement; beuser:mod.wizards.newContentElement]

Example 1:

Add a new element (header) to the "common" group:
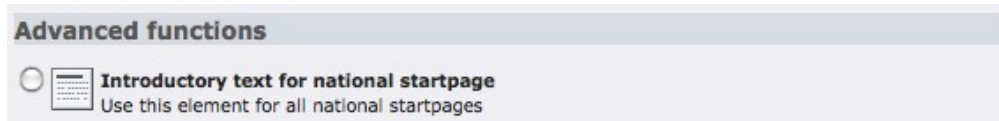
```
mod.wizards.newContentElement.wizardItems.common.elements.header {
    icon = gfx/c_wiz/regular_text.gif
    title = Header
    description = Adds a header element only
    tt_content_defValues {
        CType = header
    }
}
mod.wizards.newContentElement.wizardItems.common.show := addToList(header)
```
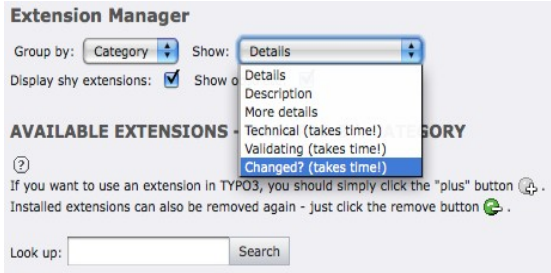
Example 2:

Create a new group and add a (pre-filled) element to it:

```
mod.wizards.newContentElement.wizardItems.myGroup {
    header = LLL:EXT:cms/layout/locallang.xml:advancedFunctions
    elements.customText {
        icon = gfx/c_wiz/regular_text.gif
        title = Introductory text for national startpage
        description = Use this element for all national startpages
        tt_content_defValues {
            CType = text
            bodytext (
<h2>Section Header</h2>
<p class="bodytext">Lorem ipsum dolor sit amet, consectetur, sadipisci velit ...</p>
            )
            header = Section Header
            header_layout = 100
        }
    }
}
mod.wizards.newContentElement.wizardItems.myGroup.show = customText
```

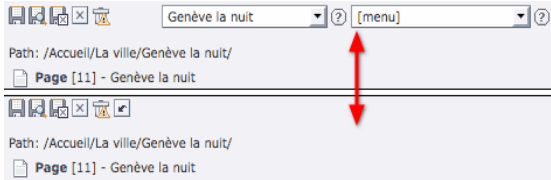This will add the following at the bottom of the new content element wizard:

Tools > Extension Manager (mod.tools_em)

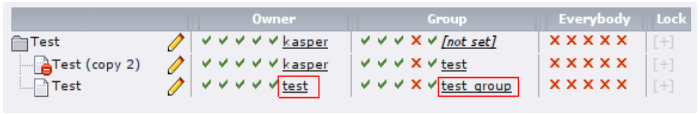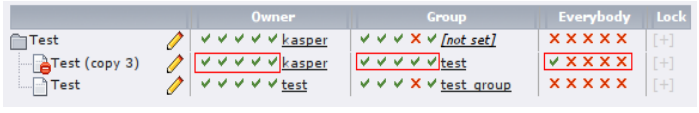| Property:	Data type:	Description:	Default:
allowTVlisting	boolean	If set the Technical and Validation listings are available in the EM. Those will evaluate ALL available extensions and that can take many seconds (up to 30) depending on number of extensions. **Example**: mod.tools_em.allowTVlisting = 1 Enables these options in the Extension Manager: 	

[beuser:mod.tools_em]

Edit document 'module' (mod.xMOD_alt_doc):

Property:	Data type:	Description:	Default:

disableDocSelector	boolean	If set, the document selector is disabled
		
		Example:
		
		    mod.xMOD_alt_doc {
		        disableDocSelector = 1
		        disableCacheSelector = 1
		    }
		
		Note: as of TYPO3 4.2 this has been replaced by the "opendocs" extension, on which this setting has no effect anymore. The above screenshot comes from TYPO3 4.1.	

disableCacheSelector	boolean	If set, the cache/save/close selector is disabled. *See example above.* **Note**: as of TYPO3 4.2 this menu dœs not exist anymore.	

[page:mod.xMOD_alt_doc; beuser:mod.xMOD_alt_doc]

->TCEMAIN

Property:	Data type:	Description:
table.[*tablename*] default	->TCEMAIN_tables	Options for each table.

permissions.userid permissions.groupid	int+	Hardcodes the default owner Backend User / Group UID of new and copied pages.
		
		(The default owner is the backend user that creates / copies the record. The default user group is the "main group" of the backend user - the group in the very top of the users group-list.)
		
		Example:
		
		`TCEMAIN {`
		    `# Owner be_users UID for new pages:`
		  `permissions.userid = 2`
		    `# Owner be_groups UID for new pages:`
		  `permissions.groupid = 3`
		`}`
		
		Backend User with UID 2 is "test" and the Backend Group with UID 3 is "test_group". With the configuration above a new page would be created with this user/group setting instead of the defaults:

permissions.user permissions.group permissions.everybody	list of string/int[0-31]	Default permissions set for owner-user, owner-group and everybody.
		
		Keylist: show,edit,delete,new,editcontent
		Alternatively you can specify an integer from 0 to 31 indicating which bits corresponding to the keylist should be set. (Bits in keylist: show=1,edit=2,delete=4,new=8,editcontent=16)
		
		Defaults from $TYPO3_CONF_VARS:
		    `"user" => "show,edit,delete,new,editcontent",`
		    `"group" => "show,edit,new,editcontent",`
		    `"everybody" => ""`
		
		Example:
		
		`TCEMAIN.permissions {`
		    `# User can do anything (default):`
		  `user = 31`
		    `# Group can do anything`
		    `# (normally "delete" is disabled)`
		  `group = 31`
		    `# Everybody can at least see the page`
		    `# (normally everybody can do nothing)`
		  `everybody = show`
		`}`
		
		The page "Test (copy 3)" was created with the settings above for permissions. Compared to the two other pages created with default permissions you can see the effect: The Backend Group can now also delete the page by default and Everybody has read access:

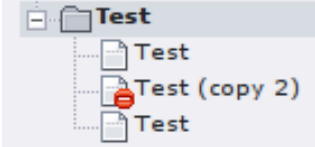| Property:	Data type:	Description:

clearCacheCmd	List of values (integers, "all", "pages")	This can allow you to have the cache for additional pages cleared when saving to some page or branch of the page tree.
		
		Examples:
		
		``
		    # Will clear the cache for page ID 12 and 23
		    # when saving a record in this page:
		TCEMAIN.clearCacheCmd = 12,23
		
		    # Will clear all pages cache:
		TCEMAIN.clearCacheCmd = pages
		
		    # Will clear ALL cache:
		TCEMAIN.clearCacheCmd = all
		``

clearCache_pageSiblingChildren	boolean	If set, then children of all siblings of a page being edited will have the page cache cleared. (Default is that when a page record is edited, the cache for itself and siblings (same level) is cleared.)
clearCache_pageGrandParent	boolean	If set, then the grand parent of a page being edited will have the page cache cleared.
clearCache_disable	boolean	If set, then the automatic clearing of page cache when records are edited etc. is disabled. This also disables the significance of the two "clearCache_page*" options listed above.

translateToMessage	string	Defines the string that will be prepended to every field value if you copy an element to another language version.
		
		The special string "%s" will be replaced with the language title.
		Default is "Translate to [language title]:".
		
		Example:
		
		``
		TCEMAIN {
		    # Set a German label:
		  translateToMessage = Bitte in "%s" übersetzen:
		}
		``

notificationEmail_subject	string	Defines the header of workspace status change notification emails.
		
		The special string "%s" will be replaced with the tablename and record uid of the changed record.
		Default is "TYPO3 Workspace Note: Stage Change for [tablename:uid]".
		
		Example:
		
		``
		TCEMAIN {
		    # Set a German header:
		  notificationEmail_subject = TYPO3-
		Arbeitsumgebungshinweis: Änderung der Stufe für %s
		}
		``

Property:	Data type:	Description:

notificationEmail_body	string	Defines the bodytext of workspace status change notification emails.
		
		There are eleven special strings (like "%s", "%11$s" and "%10$s") present in that text.
		In order of appearance the nine strings "%s" will be replaced as follows:
		1. Sitename coming from $GLOBALS['TYPO3_CONF_VARS']['SYS']['sitename']
		2. Link to the TYPO3_mainDir
		3. Title of the workspace
		4. uid of the workspace
		5. Tablename and uid of the changed record
		6. Name of the new stage
		7. Comment which the user entered when requesting the stage change
		8. Realname of the changing user coming from his user record
		9. Username of the changing user coming from his user record
		Note that you cannot change their order.
		Finally, there are the strings %10$s and %11$s.
		%10$s is replaced with the path of the changed record and
		%11$s is replaced with the title of the record.
		
		Default text is:
		"At the TYPO3 site "%s" (%s)
		in workspace "%s" (#%s)
		the stage has changed for the element(s) "%11$s" (%s) at location "%10$s" in the page tree:
		==> %s
		User Comment:
		"%s"
		
		State was change by %s (username: %s)"
		
		Example:
		
		<pre>TCEMAIN {
		    # Set a German bodytext:
		  notificationEmail_body (
		    Auf der TYPO3-Seite "%s" (%s)
		    wurde in der Arbeitsumgebung "%s" (%s)
		    die Stufe des Elements/der Elemente "%11$s"
		(%s) am Ort "%10$s" im Seitenbaum verändert:
		    ==> %s
		    Kommentar des Benutzers:
		    "%s"
		
		    Die Stufe wurde geändert von %s (Benutzername:
		%s).
		  )
		}</pre>

[page:TCEMAIN]

->TCEMAIN_tables

Processing options for a $TCA configured table.

Property:	Data type:	Description:
history.maxAgeDays	int+	The number of days elements are in the history at most. Takes precedence over keepEntries. Default is 7 days. Range 0-200. Zero turns the magAgeDays off.

Property:	Data type:	Description:
disablePrependAtCopy	boolean	Disables the "prependAtCopy" feature (if configured for table in $TCA) (The word "prepend" is misguiding - the "(copy)" label is *appended* to (put after) the record title! Sorry for that mistake, it isn't the only time I have made that.) **Example**: ```\nTCEMAIN.table.pages {\n # Pages will NOT have \"(copy)\" appended:\n disablePrependAtCopy = 1\n # Pages will NOT be hidden upon copy:\n disableHideAtCopy = 1\n}\n``` These settings means that a page copied (say, the last page in this tree, labeled "Test") will neither have "(copy X)" appended nor be hidden. This effect is shown in the first sub page here, labeled "Test". It appears exactly like the third page that was copied. The page in the middle, labeled "Test (copy 3)" is the default mode where the page is hidden and the "(copy X)" suffix is added if another page with the same named existed already.
disableHideAtCopy	boolean	Disables the "hideAtCopy" feature (if configured for table in $TCA) For example, see "disablePrependAtCopy" above.

[page:TCEMAIN.default/TCEMAIN.table.(tablename)/->TCEMAIN_tables]

->TCEFORM

Allows detailed configuration of how TCEforms are rendered for a page tree branch and for individual tables if you like. You can enable and disable options, blind options in selector boxes etc.

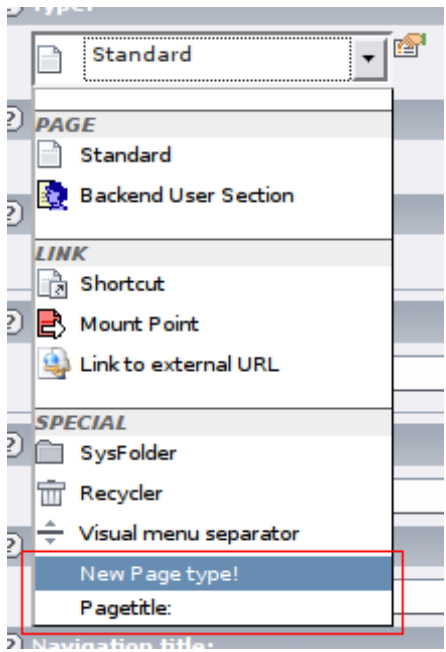| Property:	Data type:	Description:
[tablename].*[field]* *[tablename]*.*[field]*.**types**.*[type]*	->TCEFORM_confObj	These objects contain additional configuration of the TCEFORM interface. For the properties available, refer to the table below. This is a description of how you can customize in general and override for specific types. TCEFORM.[tablename].[field] - configures the field in TCEFORM for all types. TCEFORM.[tablename].[field].types.[type] - configures the field in TCEFORM in case the 'type'-value of the field matches type.

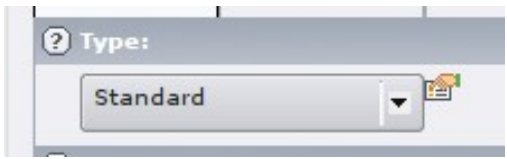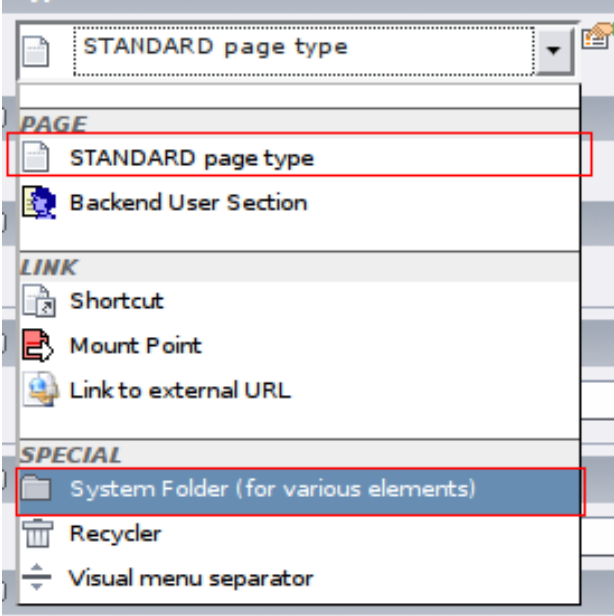| Property:	Data type:	Description:
[*tablename*].[*field*].config.[*key*]	string / array	This setting allows to override TCA field configuration and offers a flexible opportunity to reuse tables and TCA definitions but adapt them to individual demands. So this will influence configuration settings in $TCA[<tablename>]['columns'][<field>]['config'][<key>]. Depending on the $TCA type of the field, the allowed keys are: **input** - size, max **text** - cols, rows, wrap **check** - cols, showIfRTE **select** - size, autoSizeMax, maxitems, minitems **group** - size, autoSizeMax, max_size, show_thumbs, maxitems, minitems, disable_controls **inline** - appearance, foreign_label, foreign_selector, foreign_unique, maxitems, minitems, size, autoSizeMax, symmetric_label
suggest.default **suggest.[queryTable]** **[tablename].** **[field].suggest.default** **[tablename].** **[field].suggest.** **[queryTable]**	->TCEFORM_suggest	Configuration for the "suggest" wizard. Each level of the configuration overwrites the values of the level below it: - "suggest.default" is overwritten by "suggest.[queryTable]". - Both are overwritten by "[tablename].[field].suggest.default" which itself is overwritten by "[tablename].[field].suggest.[queryTable]" suggest.default: Configuration for all suggest wizards in all tables suggest.[queryTable]: Configuration for all suggest wizards from all tables listing records from table [queryTable] [tablename].[field].suggest.default Configuration for the suggest wizard for field [field] in table [tablename] [tablename].[field].suggest.[queryTable] Configuration for the suggest wizard for field [field] in table [tablename] listing records from [queryTable]

[page:TCEFORM]

->TCEFORM_confObj

Properties for the TCEFORM configuration object (see introduction above).

Property:	Data type:	Description:

disabled	boolean	If set, the field is not rendered.
		
		Example:
		
		``
		TCEFORM.pages.title {
		    # You cannot edit the Page title field now:
		    disabled = 1
		}
		``

label	string (localized)	This allows you to enter alternative labels for any field.
		
		Example:
		
		``
		TCEFORM.pages.title {
		  label =
		LLL:EXT:my_ext/locallang_db.xml:table.column
		  label.default = New Label
		  label.de = Neuer Feldname
		}
		``

Property:	Data type:	Description:

keepItems	list of values	*(applies to select-types only)*
		
		This keeps in the list only the items defined here. All others are removed.
		
		Example:
		
		``
		TCEFORM.pages.doktype {
		    # Show only standard and "Spacer" page types
		  keepItems = 1,199
		}
		``

removeItems	list of values	*(applies to select-types only)*
		
		This removes the items from the list which has a value found in this comma list of values.
		
		Example:
		
		``
		TCEFORM.pages.doktype {
		    # Remove the "Recycler" and "Spacer" page type options:
		  removeItems = 199, 255
		}
		``

addItems.[itemValue]	string (localized)	*(applies to select-types only)*
		
		This will add elements to the list. Notice that the added elements might be removed if the selector represents records. In that case only still existing records will be preserved.
		
		Example:
		
		``
		TCEFORM.pages.doktype {
		    # Creates a new page type option:
		  addItems.123 = New Page type!
		
		    # Creates yet a page type with "locallang" title:
		  addItems.124 = LLL:EXT:lang/locallang_tca.php:title
		}
		``
		
		This example extends the options for Page types with two new items:
		
		Warning: This example shows the principle of adding adhoc-items to a selector box in TYPO3 but you *should not* add new *page types* or Content Element types this way!

Property:	Data type:	Description:

disableNoMatchingValueElement	boolean	*(applies to select-types only)*
		
		 If set, the element "INVALID VALUE" will not be added to the list.
		
		 Example:
		
		 If a selector box value is not available among the options in the box the default behavior of TYPO3 is to preserve the value and show a label that warns about this special state:
		
		 Setting this option disables that behavior:
		
		 ``TCEFORM.pages.doktype {
		    # "INVALID VALUE ..." label will never show up:
		  disableNoMatchingValueElement = 1
		}```
		
		 ... and the selector box will default to the first element in the selector box:

noMatchingValue_label	string (localized)	*(applies to select-types only)*
		
		 Allows for an alternative label the "noMatchingValue" element.
		 You can insert the placeholder "%s" to insert the value.
		 If you supply a blank value the label will be blank.
		
		 Example:
		
		 ``TCEFORM.pages.doktype {
		    # Alternative "INVALID VALUE ..." label:
		  noMatchingValue_label = VALUE "%s" was not
		available!
		}```
		
		 The result will be:

Property:	Data type:	Description:
| **altLabels.[item_value]** | string (localized) | *(applies to select-types only)*<br><br>This allows you to enter alternative labels for the items in the list.<br><br>**Example**:<br><br>```
TCEFORM.pages.doktype {
    # Setting alternative labels:
  altLabels.1 = STANDARD page type
  altLabels.2 = ADVANCED page type
  altLabels.254 = System Folder (for various
elements)
    # Sets the default labels for Recycler (via
"locallang"):
  altLabels.255 =
LLL:EXT:lang/locallang_tca.php:doktype.I.2
}
```<br><br>Result will be:<br><br> |
PAGE_TSCONFIG_ID	integer	*(applies to select-types with foreign table)* When the select-types are used with foreign-table the where-query has four markers (see description of $TCA in the "Inside TYPO3" document.) The value of three of these markers may be set from Page TSconfig.
PAGE_TSCONFIG_ID LIST	comma list of integers	*(applies to select-types with foreign table)* See above
PAGE_TSCONFIG_STR	string	*(applies to select-types with foreign table)* See above
itemsProcFunc.[...]	(custom)	*(applies to select-types with itemsProcFunc)* The properties of this key is passed on to the itemsProcFunc in the parameter array by the key "TSconfig".
RTEfullScreenWidth	int+/%	*(applies for RTE text fields only with the RTE wizard configured)* The width of the RTE full screen display. If nothing is set, the whole width is used which means "100%". If you set an integer value, that indicates the pixels width.

Property:	Data type:	Description:

linkTitleToSelf	boolean	*(all fields)*
		
		 If set, then the title of the field in the forms links to alt_doc.php editing ONLY that field.
		 Works for existing records only - not new records.
		
		 Example:
		
		 ``
		TCEFORM.pages.title {
		    # The label for the "title" field will link
		itself
		  linkTitleToSelf = 1
		}
		``
		
		 The result is that the label for the title field will be a link:
		
		 Clicking the link brings you to a form where only this field is shown:

[page:TCEFORM.(tablename).(field)/TCEFORM.(tablename).(field).types.(type)]

->TCEFORM_suggest

Properties for the suggest wizard (see introduction above).

Property:	Data type:	Description:	Default:

pidList	list of values	Limit the search to certain pages (and their subpages). When pidList is empty all pages will be included in the search (as long as the be_user is allowed to see them)
		
		 Example:
		
		 ``
		TCEFORM.suggest.default {
		  # sets the pidList for a suggest fields in
		all tables
		  pidList = 1,2,3,45
		}
		``	

pidDepth	int+	Expand pidList by this number of levels. Has only an effect if pidList has a value.
		
		 Example:
		
		 ``
		TCEFORM.suggest.default {
		  pidList = 6,7
		  pidDepth = 4
		}
		``	

Property:	Data type:	Description:	Default:

minimumCharacters	int+	Minimum number of characters needed to start the search. Works only for single fields.
		
		Example:
		
		``
		TCEFORM.pages.storage_pid.suggest.default {
		    minimumCharacters = 3
		}
		``
			2

maxPathTitleLength	int+	Maximum number of characters to display when a path element is too long.
		
		Example:
		
		``
		TCEFORM.suggest.default {
		    maxPathTitleLength = 30
		}
		``	

searchWholePhrase	boolean	Whether to do a LIKE=%mystring% (searchWholePhrase = 1) or a LIKE=mystring% (to do a real find as you type).
		
		Example:
		
		``
		TCEFORM.pages.storage_pid.suggest.default {
		    # configures the sugesst wizard for the field "storage_pid" in table "pages" to search only for whole phrases
		    searchWholePhrase = 1
		}
		``
			0

searchCondition	string	Additional WHERE clause (no AND needed to prepend).
		
		Example:
		
		``
		TCEFORM.pages.storage_pid.suggest.default {
		    # configures the sugesst wizard for the field "storage_pid" in table "pages" to search only for pages with doktype=1
		    searchCondition = doktype=1
		}
		``	

cssClass	string	Add a CSS class to every list item of the result list.
		
		Example:
		
		``
		TCEFORM.suggest.pages {
		    # configures all suggest wizards which list records from table "pages" to add the css-class "pages" to every list item of the result list.
		    cssClass = pages
		}
		``	

receiverClass	string	PHP class alternative receiver class - the file that holds the class needs to be included manually before calling the suggest feature, should be derived from "t3lib_tceforms_suggest_defaultreceiver"	t3lib_tceforms_ suggest_default receiver
renderFunc	string	User function to manipulate the displayed records in the result.	

hide	boolean	Hide the suggest field. Works only for single fields.
		
		Example:
		
		``
		TCEFORM.pages.storage_pid.suggest.default {
		    hide = 1
		}
		``	

[page:TCEFORM.suggest.default/TCEFORM.suggest.(queryTable)/TCEFORM.(tablename).(field).suggest.default/TCEFORM.(tablename).(field). (queryTable)]

->TSFE

Property:	Data type:	Description:	Default:
jumpUrl_transfer Session	boolean	If set, the jumpUrl redirection to the URL will be prepended with a parameter that transfers the current fe_users session to that URL. This URL should be the TYPO3 frontend in the same database, just at another domain (else it makes no sense). You can implement it in your own links if you like. This is how you do: You must send the parameter 'FE_SESSION_KEY' as GET or POST. The parameter looks like this: [fe_user-session-id]-[a hash made to prevent misuse] The parameter can be calculated like this: ```$param = '&FE_SESSION_KEY=' . rawurlencode(\n $GLOBALS['TSFE']->fe_user->id . '-' .\n md5(\n $GLOBALS['TSFE']->fe_user->id. '/' .\n $GLOBALS['TYPO3_CONF_VARS']['SYS']\n['encryptionKey']\n)\n);```	

Property:	Data type:	Description:	Default:

constants	[TypoScript Frontend Constants defaults]	Defaults for TypoScript Template constants!
		
		This features allows you to pass some amount of information (in the form of TypoScript Template constants) to the frontend. The specific use of this should be information which you want to configure for both frontend and backend. For instance you could have a backend module which should act in a certain way depending on in which branch of the page tree it operates. The change of behavior is set by Page TSconfig as always but since you need the same setting applied somewhere in the frontend you don't want the redundancy of specifying the value twice. In such a case you can use this feature.
		
		Example:
		
		`TSFE.constants.websiteConfig.id = 123`
		
		In the TypoScript templates you can now insert this constant as {$websiteConfig.id}
		
		In the backend module (in the Web main module) you can reach the value by a few lines of code like these:
		
		`$PageTSconfig =`
		`    t3lib_BEfunc::getPagesTSconfig($this->pObj->id);`
		`$websiteID =`
		`    $PageTSconfig['TSFE.']['constants.']['websiteConfig.']['id'];`
		
		Notice: In the frontend the setting of default constants will only apply to a branch of the tree *if* a template record is found on that page (or if a template record is set for "next level"). In other words: If you want the Page TSconfig constant defaults to affect only a certain branch of the page tree, make sure to create a template record (a blank one will do) on the page that carries the Page TSconfig information.	

[page:TSFE]

Next steps

For details about the nature of TypoScript as a syntax, please read the document "TypoScript Syntax &
In-depth Study".