

TSconfig

Extension Key: **doc_core_tsconfig**

Copyright 2000-2006, kasper@typo3.com, <kasper@typo3.com>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

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

Revised for TYPO3 3.7, may 2004

All options tested.

Table of Contents

TSconfig	1	->SETUP.....	8
Introduction	2	Page TSconfig	11
Syntax.....	2	Top Level Objects.....	12
User TSconfig	3	->MOD.....	12
Top Level Objects.....	4	->TCEMAIN.....	20
->ADMPANEL.....	5	->TCEFORM.....	22
->OPTIONS.....	6	->TSFE.....	26

Introduction

TypoScript is mainly known as the means to configure Templates Records for TYPO3 based websites. However TypoScript itself is [basically a way to define values in a hierarchy](#). And so TypoScript is also used to make detailed configuration of the backend, both regarding users and groups and the way modules are working in branches of the Page Tree. This is done through the 'TSconfig' which is found for both the backend and frontend users/groups in addition to the pages. This is often referred to as "User TSconfig" and "Page TSconfig".

Page TSconfig / User TSconfig

This document deals with the options you can set for "Page TSconfig" and "User TSconfig". Notice that because TypoScript is a 'declarational' language (that is, it's mere "configuration") it does not in itself carry out a sequence of actions. The reason why the TypoScript *Templates* used for the frontend seems to work a like this is because those values are interpreted into PHP function calls which constructs a page. In the case where TypoScript is used for configuration like here it's clearly more like the Registration Database in Windows: The values configure how the application works.

For details about the nature of TypoScript as a syntax, please [read the document "TypoScript Syntax"](#).

Syntax

The syntax is as usual TypoScript, only difference is that you cannot use any Constants or Conditions.

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's 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 (extension: beuser) module by clicking a username. Then you'll see the TSconfig tree among other information. Here's an example:

This configuration could be set by entering this value in the TSconfig field of the backend user:

```
options.clearCache.all = 1
TCAdefaults.tt_content {
    hidden = 1
    header = 234
}
```

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 clear cache option (marked blue). The default value of the hidden field is not changed and simply inherited directly from the group:

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

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.shortcutFrame = 1
');
```

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

Top Level Objects

This is 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.defaults 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 <i>not</i> remove it again by rather set it blank if you want to disable the effect again!	
TCAdefaults.[tablename].[field]	string	Sets default values for records. The order of default values when creating new records in the backend is this: <ul style="list-style-type: none">Value from \$TCAValue from User TSconfig (these settings)Value from "defVals" GET vars (see alt_doc.php)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: <ul style="list-style-type: none">Value from \$TCAValue 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" <code>TCAdefaults.pages.hidden = 0</code>	
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: <code>page.TCEMAIN.table.pages.disablePrependAtCopy = 1</code>	

Property:	Data type:	Description:	Default:
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.	
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": <i>name="TSFE_ADMIN_PANEL[tsdebug_displayTimes]"</i> 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): <i>Forces edit-panels to appear regardless of the selectorbox.</i> .forceDisplayFieldIcons (boolean): <i>Forces edit-icons to appear regardless of the selectorbox.</i> .forceNoPopup (boolean): <i>Forces edit-forms to open in same window - not pop up window.</i> <i>Example, that forces the display of the edit icons without displaying the admin-panel itself:</i> <pre>admPanel { enable.edit = 1 module.edit.forceDisplayFieldIcons = 1 hide = 1 }</pre>	

[beuser:admPanel]

Tip:

If you want to link to the login-screen of the backend, but wishes that the user should return to the front-end for editing, you do it by this link, sending the "redirect_url" parameter to the login-screen. In that case the backend interface is not started.

`Backend login returning to front-end`

This is how the admin panel looks. 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)	
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

Property:	Data type:	Description:	Default:
disableDocModuleInAB	boolean	If set, the doc module is not displayed in the backend even if enabled in general.	
shortcutFrame	boolean	If set, the shortcut frame in the bottom of the window appears.	
shortcutGroups	Array of integers/strings	<p>Set shortcutGroups 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):</p> <p>1: Pages 2: Records 3: Files 4: Tools 5: Miscellaneous</p> <p>Set 0 to disable one of these group IDs, 1 to enable it (this is the default) or "string" to change the label accordingly.</p> <p>Example:</p> <pre>shortcutGroups { 1=1 2=My Group 3=0 4= }</pre> <p>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.</p>	
shortcut_onEditId_dontSetPageTree	boolean	If set, the page tree is <i>not</i> opened to the page being edited when an id number is entered in the "Edit Id" box	
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, then the user cannot create or edit shortcuts. 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"	<p>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".</p> <p>Example:</p> <p>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).</p> <pre>options.saveDocNew = 0 options.saveDocNew.tt_content = top</pre>	
disableDelete disableDelete.[table]		Disables the "Delete" button in TCEFORMs. Overriding for single tables works like "saveDocNew" above.	
showHistory showHistory.[table]		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.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.	
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	

Property:	Data type:	Description:	Default:
		<p>possible options are "pageTree", "pageList", "folderTree", "folderList". "page" and "folder" obviously points to either the Web og File main module. "Tree" and "List" points to whether the menu is activated from the page/folder tree or the listing of records/files</p> <p>Items to disable are (for "page" type - that is database records):</p> <p>view,edit,hide,new,info,copy,cut,paste,delete,move_wizard,history,perms,new_wizard,hide,edit_access,edit_pageheader,db_list</p> <p>Items to disable are (for "folder" type - that is files/folders):</p> <p>edit,upload,rename,new,info,copy,cut,paste,delete</p>	
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.	
overridePageModule	string	<p>By this value you can substitute the default "Web > Page" module key ("web_layout") with another backend module key.</p> <p>Example:</p> <pre>options.overridePageModule = web_txttemplavoilaM1</pre> <p>This will enable TemplaVoila page module as default page module.</p>	
moduleMenuCollapsable	boolean	If set, the user can collapse main modules in the left menu.	1
alertPopups	bitmask	<p>Configure which Javascript popup alerts have to be displayed and which not:</p> <pre>1 - onChange 2 - copy / move / paste 4 - delete 8 - FE editing 128 - other (not used yet)</pre> <p>Default: 255 (show all warnings)</p>	
defaultFileUploads	integer	Default number of file upload forms shown in the File->List module	
hideRecords.[table]	[list of record uids]	<p>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.</p> <p>This option is currently implemented for pages only and has an effect in following places:</p> <ul style="list-style-type: none"> - Pagetree navigation frame - Web>List module - New record wizard <p>Example:</p> <pre>options.hideRecords.pages = 12,45</pre>	

[beuser:options]

->SETUP

Default values and overriding values for the "User > Setup" module.

Notice: the User > Setup module only represents a subset of the options from the table below.

User Settings - [admin]

Simulate backend user:

LANGUAGE:

English

STARTUP:

Use condensed mode in backend (for small screens):	<input type="checkbox"/>
Select navigation mode:	<div style="border: 1px solid #ccc; padding: 2px;">Left frame menu (default)</div>
Start up in the module 'Task Center':	<input type="checkbox"/>
Show Thumbnails by default:	<input type="checkbox"/>
Show help text when applicable:	<input checked="" type="checkbox"/>
Max. Title Length:	<div style="border: 1px solid #ccc; padding: 2px;">30</div>

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 eg) 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! (eg. 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)	
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: "", "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. Eg. "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.	
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	

Property:	Data type:	Description:	Default:
		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.	
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.	

[beuser:setup.default/setup.override]

Don't use any other properties than the ones listed in the table above.

Page TSconfig

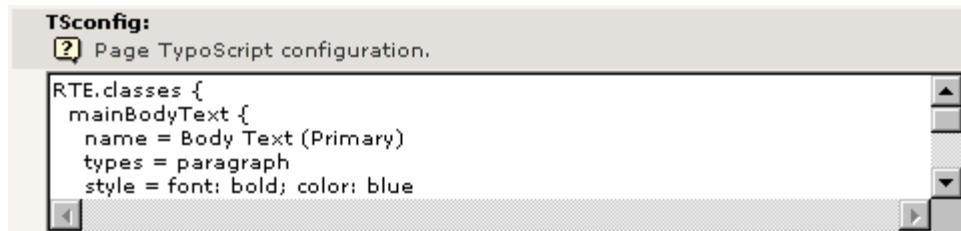
The Page TSconfig primarily concerns configuration of the modules in the TYPO3 backend. It may also relate to values in the frontend display occasionally.

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

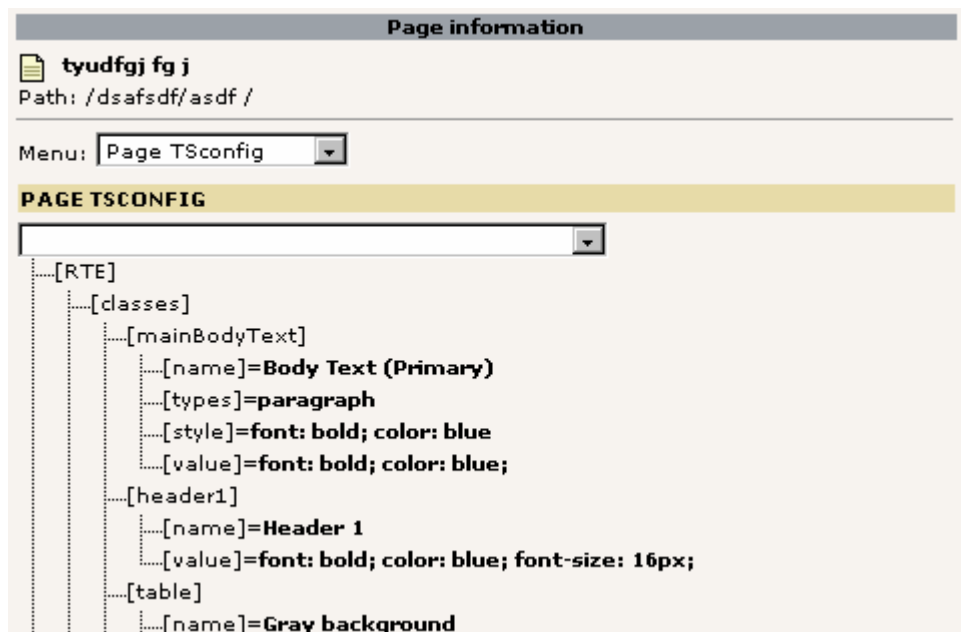
This is an example of the TSconfig field with a snippet of configuration for the Rich Text Editor. Precisely the Rich Text Editor is quite a good example of the usefulness of 'Page TSconfig'. The reason is that you may need the RTE to work differently in different parts of the website. For instance you 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 (extension "info_pagetsconfig"):



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

This is the Page TSconfig Top Level Objects (TLOs):

Property:	Data type:	Description:	Default:
mod	->MOD	Options for the backend modules. <i>Notice that these options are merged with settings from User TSconfig (TLO: mod) which takes precedence.</i>	
RTE	->RTE	This defines configuration for the Rich Text Editor. Please refer to the RTE section in the "TYPO3 Core API" document .	
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]

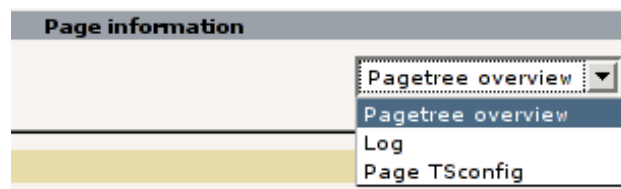
->MOD

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 has 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
}
```

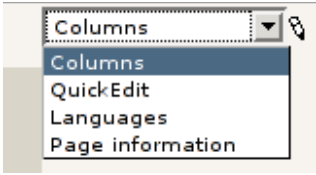
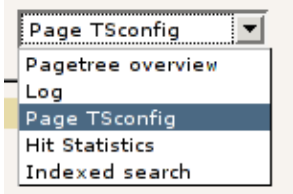
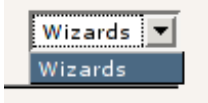
The function menu will now look like this:


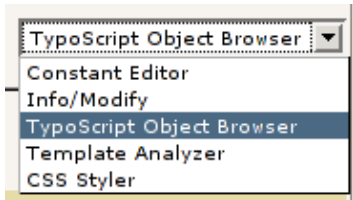
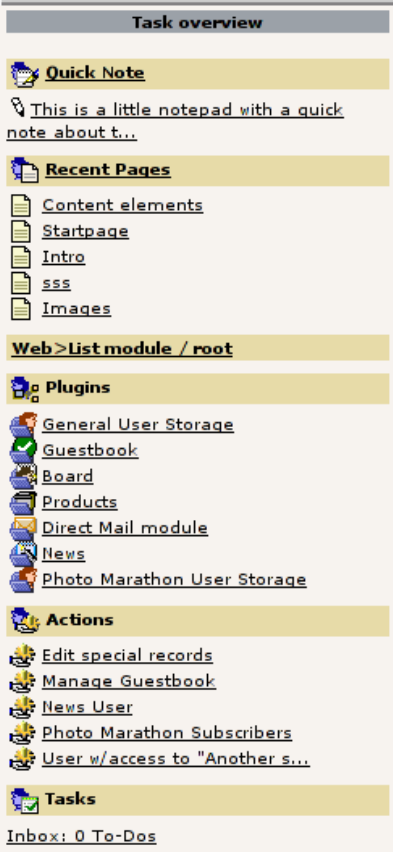


The 'Page TSconfig' option is simply disabled by setting this Page TSconfig!

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 is listed in the table below (Testsite package):

Property:	-	Description:
web_layout.menu.function		<p>Web > Page module</p>  <p>Option tags:</p> <pre><select name="SET[function]"> <option value="1">Columns</option> <option value="0">QuickEdit</option> <option value="2">Languages</option> <option value="3">Page information</option> </select></pre> <p>Example:</p> <pre># Disables all items except the "QuickEdit" item: mod.web_layout.menu.function { 1 = 0 2 = 0 3 = 0 }</pre>
web_info.menu.function		<p>Web > Info module</p>  <p>Option tags:</p> <pre><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></pre> <p>Example:</p> <pre># Disables the item "Hit Statistics": mod.web_info.menu.function { tx_cms_webinfo_hits = 0 }</pre>
web_func.menu.function		<p>Web > Functions module</p>  <p>Option tags:</p> <pre><select name="SET[function]"> <option value="tx_funcwizards_webfunc">Wizards</option> </select></pre>
web_func.menu.wiz		<p>Web > Functions module, Wizards sub module</p> <p>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.</p>

Property:	-	Description:
		 <p>Option tags:</p> <pre><select name="SET[wiz]"> <option value="tx_wizardcrpages_webfunc_2">Create multiple pages</option> <option value="tx_wizardsortpages_webfunc_2">Sort pages</option> </select></pre> <p>Example:</p> <pre># Disables the sub-item "Create multiple pages": mod.web_func.menu.wiz { tx_wizardcrpages_webfunc_2 = 0 }</pre>
web_ts.menu.function		<p>Web > Template module</p>  <p>Option tags:</p> <pre><select name="SET[function]"> <option value="tx_tstemplateeditor">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></pre>
user_task.menu.function		<p>User > Task Center</p> <p>This is slightly different because behind the scene it uses the same items in the Task Center. There is one tricky thing though either have to set default Page</p>  <p>...tor box function menu. But before you can also blind page in the page tree! So you s here!</p>

Property:	-	Description:
		<p>Keys are:</p> <p>tx_sysnotepad = Quick Note tx_taskcenterrecent = Recent Pages tx_taskcenterrootlist = Web>List module / root tx_taskcentermodules = Plugins tx_sysaction = Actions tx_systodos = Tasks</p> <p>Example: Set this as <i>User TSconfig</i>:</p> <pre># Task Center configuration: mod.user_task.menu.function { # Disable "Recent Pages" display: tx_taskcenterrecent = 0 # Disable "Action" list tx_sysaction = 0 }</pre>

[page:mod; beuser:mod]

Since function menu items can be provided from extensions it is not possible to create a complete list of menu keys. (The list in the table above is what is installed by default in the Testsite package.)

Therefore, if you want to blind a menu item the only sure way of doing it is to look in the HTML source of the backend module and find the selector box with the function menu and 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 does 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 lets consider the case from above where a menu item in the Web>Info module was disabled in the Page TSconfig by this value:

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

If however we set this configuration in the TSconfig of a certain backend user (eg. 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's another example where the value 'mod.web_layout.editFieldsAtATime' has been overridden by User TSconfig to the value '5' instead of '1'. The image shows you how to check the Page TSconfig (the upper image) and also see if User TSconfig of the current user (the logged in user) has altered anything from this configuration (lower image):

Page information

3453453453 4534 5

Path: /dsafsdf/

Menu:

Page TSconfig

PAGE TSCONFIG

...[mod]

...[web_layout]

...[editFieldsAtTime]=1

Page information

3453453453 4534 5

Path: /dsafsdf/

Menu:

Page TSconfig

PAGE TSCONFIG

Module key ('mod') with overruling user settings

...[web_layout]

...[editFieldsAtTime]=5

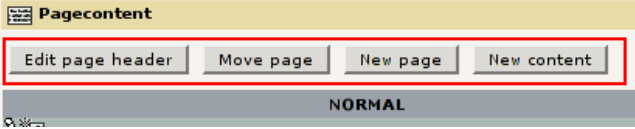
Shared options for modules (mod.SHARED):

Property:	Data type:	Description:	Default:
colPos_list	(list of integers, blank = don't do anything.)	<p>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.</p> <p>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.</p> <p>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:</p> <p>Left: 1 Normal: 0 Right: 2 Border: 3</p> <p>Example: This results in only the Normal and Border column being displayed:</p> <p><code>mod.SHARED.colPos_list = 0,3</code></p>	
defaultLanguageLabel	string	<p>Alternative label for "Default" when language labels are shown in the interface.</p> <p>Used on Web>List, Web>Page and TemplaVoilas page module.</p>	
defaultLanguageFlag	string	<p>Flag icon filename from gfx/flags/ for default language</p> <p>User in Web>List and TemplaVoila page module.</p>	

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

Web>Page (mod.web_layout):

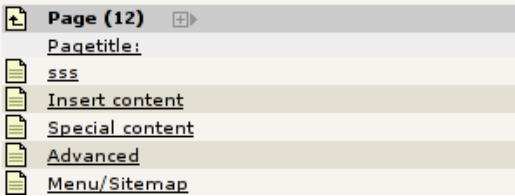
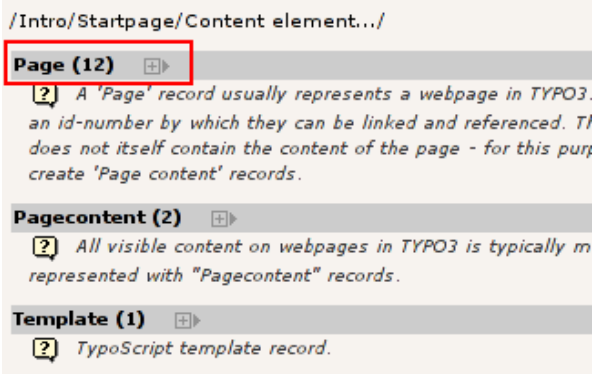
Property:	Data type:	Description:	Default:
tt_content.colPos_list	(list of integers, blank = don't do anything.)	<p>See mod.SHARED.colPos_list for details.</p> <p>If non-blank, this list will override the one set by mod.SHARED.colPos_list.</p> <p>Example: This results in only the Normal and Border column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


Property:	Data type:	Description:	Default:
		<code>mod.web_layout.tt_content.colPos_list = 0,3</code>	
tt_content.fieldOrder	list of field names from tt_content table	<p>This allows you to specify (and thereby overrule) the preferred order of the fieldnames 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.</p> <p>Example: This results in the 'Text' field and thereafter 'Header' field being display as the very first fields instead of the 'Type' field.</p> <pre>mod.web_layout.tt_content { fieldOrder = bodytext, header }</pre>	
editFieldsAtATime	int+	<p>Specifies the number of subsequent content elements to load in the editform when clicking the editicon of a content element in the 'Columns' view of the module.</p> <p>Example:</p> <pre>mod.web_layout { editFieldsAtATime = 2 }</pre>	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	<p>Disables the large buttons in top of the Columns view.</p> <p>These are the buttons that are hidden by this option:</p> 	
disableAdvanced	boolean	Disables the clear cache advanced function in the bottom of the page in the module, including the "Create new record" link.	
disableNewContentElement Wizard	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	<p>Alternative label for "Default" when language labels are shown in the interface.</p> <p>Overrides the same property from mod.SHARED if set.</p>	
defLangBinding	boolean	<p>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.</p> <p>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.</p>	
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	<p>If set, the link in the bottom of the page, "Create new record", is hidden.</p> <p>Example:</p> <pre>mod.web_list { noCreateRecordsLink = 1 }</pre>	

Property:	Data type:	Description:	Default:
		}	
alternateBgColors	boolean	<p>If set, the background colors of elements will alternate.</p> <p>Example:</p> <pre>mod.web_list { alternateBgColors = 1 }</pre> <p>The result is alternating background colors for each element:</p> 	
disableSingleTableView	boolean	<p>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).</p>	
listOnlyInSingleTableView	boolean	<p>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!</p> <p>Example:</p> <pre>mod.web_list { listOnlyInSingleTableView = 1 }</pre> <p>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:</p> 	

Property:	Data type:	Description:	Default:
allowedNewTables	<i>list of tablenames</i>	<p>If this list is set, then only tables listed here will have a link to "create new" in the page and subpages.</p> <p>This also affects "db_new.php" (the display of "Create new record")</p> <p>Note: Technically records can be created (eg. by copying/moving), so this is "pseudo security". The point is to reduce the number of options for new records visually.</p> <p>Example:</p> <pre>mod.web_list { allowedNewTables = pages, tt_news }</pre> <p>Only pages and tt_news table elements will be linked to in the New record screen:</p> 	
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.	
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	<p>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.</p> <p>Also see "options.overridePageModule"</p>	
clickTitleMode	string	<p>Keyword which defines what happens when a user clicks the title in the list.</p> <p>Default is that pages will go one level down while other records have no link at all.</p> <p>Keywords: edit = Edits record info = Shows information show = Shows page/content element in frontend</p>	

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

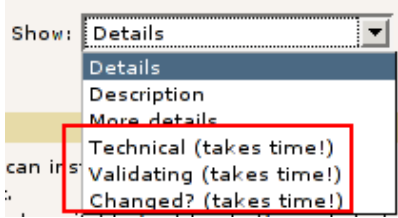
Web>View (mod.web_view):

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

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

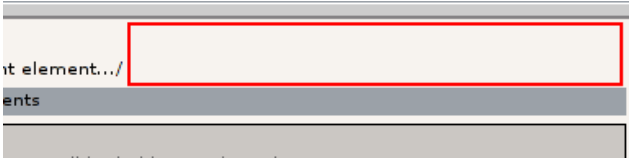
Tools>Extension Manager (mod.tools_em):

Property:	Data type:	Description:	Default:
allowTVlisting	boolean	If set the Technical and Validation listings are available in the EM.	

Property:	Data type:	Description:	Default:
		<p>Those will evaluate ALL available extensions and that can take many seconds (up to 30) depending on number of extensions.</p> <p>Example:</p> <pre>mod.tools_em.allowTVlisting = 1</pre> <p>Enables these options in the Extension Manager:</p> 	

[beuser:mod.tools_em]

Edit document 'module' (mod.xMOD_alt_doc):

Property:	Data type:	Description:	Default:
disableDocSelector	boolean	<p>If set, the document selector is disabled</p> <p>Example:</p> <pre>mod.xMOD_alt_doc { disableDocSelector = 1 disableCacheSelector = 1 }</pre> <p>The result is that both the document and cache action selector is hidden in the form page:</p>  <p>(The interesting part is what is missing...)</p>	
disableCacheSelector	boolean	<p>If set, the cache/save/close selector is disabled.</p> <p>See example above.</p>	

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

->TCMAIN

Property:	Data type:	Description:
table.[tablename] default	->TCMAIN_tables	Options for each table.
permissions.userid permissions.groupid	int+	<p>Hardcodes the default owner Backend User / Group UID of new and copied pages.</p> <p>(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.)</p> <p>Example:</p> <pre>TCMAIN { # Owner be_users UID for new pages: permissions.userid = 2 # Owner be_groups UID for new pages: permissions.groupid = 3 }</pre>

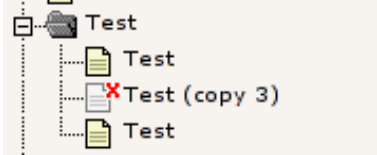
Property:	Data type:	Description:																
		<p>Backend User with UID 2 is "test" and the Backend Group with UID 3 is "test_group". With the configuration above a new page would be created with this user/group setting instead of the defaults:</p> <div><table><thead><tr><th></th><th>Owner</th><th>Group</th><th>Everybody</th></tr></thead><tbody><tr><td>Test</td><td>***** kasper</td><td>***x*</td><td>xxxxx</td></tr><tr><td>Test (copy 2)</td><td>***** kasper</td><td>***x* test</td><td>xxxxx</td></tr><tr><td>Test</td><td>***** test</td><td>***x* test_group</td><td>xxxxx</td></tr></tbody></table></div>		Owner	Group	Everybody	Test	***** kasper	***x*	xxxxx	Test (copy 2)	***** kasper	***x* test	xxxxx	Test	***** test	***x* test_group	xxxxx
	Owner	Group	Everybody															
Test	***** kasper	***x*	xxxxx															
Test (copy 2)	***** kasper	***x* test	xxxxx															
Test	***** test	***x* test_group	xxxxx															
permissions.user permissions.group permissions.everybody	list of string/int[0-31]	<p>Default permissions set for owner-user, owner-group and everybody.</p> <p>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)</p> <p>Defaults from \$TYPO3_CONF_VARS: "user" => "show,edit,delete,new,editcontent", "group" => "show,edit,new,editcontent", "everybody" => ""</p> <p>Example:</p> <pre>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 }</pre> <p>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:</p> <div><table><thead><tr><th></th><th>Owner</th><th>Group</th><th>Everybody</th></tr></thead><tbody><tr><td>Test</td><td>***** kasper</td><td>***x*</td><td>xxxxx</td></tr><tr><td>Test (copy 3)</td><td>***** kasper</td><td>***** test</td><td>*xxxx</td></tr><tr><td>Test</td><td>***** test</td><td>***x* test_group</td><td>xxxxx</td></tr></tbody></table></div>		Owner	Group	Everybody	Test	***** kasper	***x*	xxxxx	Test (copy 3)	***** kasper	***** test	*xxxx	Test	***** test	***x* test_group	xxxxx
	Owner	Group	Everybody															
Test	***** kasper	***x*	xxxxx															
Test (copy 3)	***** kasper	***** test	*xxxx															
Test	***** test	***x* test_group	xxxxx															
clearCacheCmd	List of values (integers, "all", "pages")	<p>This can allow you to have the cache for additional pages cleared when saving to some page or branch of the page tree.</p> <p>Examples:</p> <pre># 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</pre>																
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.																

Property:	Data type:	Description:
		<p>The special string "%s" will be replaced with the language title. Default ist "Translate to [language title]:".</p> <p>Example:</p> <pre>TCEMAIN { # Set a German label: translateToMessage = Bitte in "%s" übersetzen: }</pre>

[page:TCEMAIN]

->TCEMAIN_tables

Processing options for a \$TCA configured table.

Property:	Data type:	Description:
history.maxAgeDays	int+	<p>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 of.</p>
disablePrependAtCopy	boolean	<p>Disables the "prependAtCopy" feature (if configured for table in \$TCA)</p> <p>(The word "prepend" is misleading - the "(copy)" label is <i>appended</i> to (put after) the record title! Sorry for that mistake, it isn't the only time I have made that.)</p> <p>Example:</p> <pre>TCEMAIN.table.pages { # Pages will NOT have "(copy)" appended: disablePrependAtCopy = 1 # Pages will NOT be hidden upon copy: disableHideAtCopy = 1 }</pre> <p>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.</p> 
disableHideAtCopy	boolean	<p>Disables the "hideAtCopy" feature (if configured for table in \$TCA)</p> <p>For example, see "disablePrependAtCopy" above.</p>

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

->TCEFORM

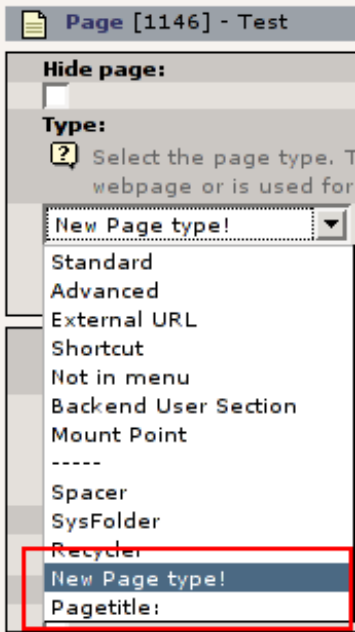
Allows detailed configuration of how TCEforms are rendered for a page tree branch and for individual tables if you like. You can enable and disable options, blind options in selector boxes etc.




Property:	Data type:	Description:
<code>[tablename].[field]</code> <code>[tablename].[field].types.[type]</code>	->TCEFORM_confObj	<p>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.</p> <p>TCEFORM.[tablename].[field] - configures the field in TCEFORM for all types.</p> <p>TCEFORM.[tablename].[field].types.[type] - configures the field in TCEFORM in case the 'type'-value of the field matches type.</p>

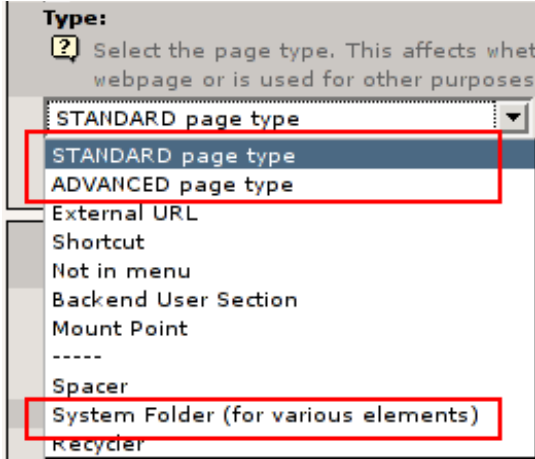
[page:TCEFORM]

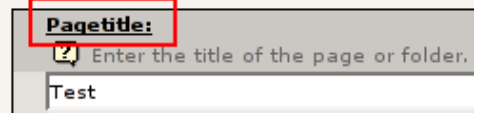
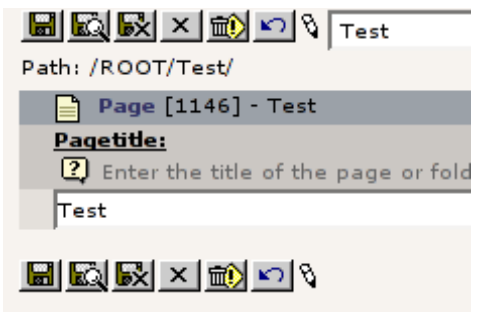
->TCEFORM_confObj

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

Property:	Data type:	Description:
disabled	boolean	<p>If set, the field is not rendered.</p> <p>Example:</p> <pre>TCEFORM.pages.title { # You cannot edit the Page title field now: disabled = 1 }</pre>
removeItems	list of values	<p>(applies to select-types only)</p> <p>This removes the items from the list which has a value found in this comma list of values.</p> <p>Example:</p> <pre>TCEFORM.pages.doktype { # Remove the "Recycler" and "Spacer" page type options: removeItems = 199, 255 }</pre>
addItem.[itemValue]	string (label, LS)	<p>(applies to select-types only)</p> <p>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.</p> <p>Example:</p> <pre>TCEFORM.pages.doktype { # Creates a new page type option: addItem.123 = New Page type! # Creates yet a page type with "locallang" title: addItem.124 = LLL:EXT:lang/locallang_tca.php:title }</pre> <p>This example extends the options for Page types with two new items:</p>  <p>Warning: This example shows the principle of adding adhoc-items to a selector box in TYPO3 but you <i>should not</i> add new <i>page types</i> or <i>Content Element types</i> this way!</p>
disableNoMatchingValueElement	boolean	<p>(applies to select-types only)</p> <p>If set, the element "CURRENT VALUE IS NOT AVAILABLE" will not be added to the list.</p> <p>Example:</p> <p>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:</p>

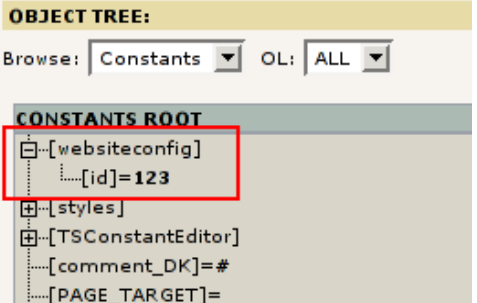
Property:	Data type:	Description:
		<div data-bbox="770 185 1385 320"> <p>Type:</p> <p> Select the page type. This affects whether the p webpage or is used for other purposes.</p> <p>[CURRENT VALUE IS NOT AVAILABLE (123)] ▾</p> </div> <p>Setting this option disables that behavior:</p> <pre>TCEFORM.pages.doktype { # "CURRENT VALUE IS NOT ..." label will never show up: disableNoMatchingValueElement = 1 }</pre> <p>... and the selector box will default to the first element in the selector box:</p> <div data-bbox="761 573 1393 707"> <p>Type:</p> <p> Select the page type. This affects whether the pag webpage or is used for other purposes.</p> <p>Standard ▾</p> </div>
noMatchingValue_label	string (LS)	<p>(applies to select-types only)</p> <p>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.</p> <p>Example:</p> <pre>TCEFORM.pages.doktype { # Alternative "CURRENT VALUE IS NOT ..." label: noMatchingValue_label = VALUE "%s" was not available! }</pre> <p>The result will be:</p> <div data-bbox="823 1120 1335 1254"> <p>Type:</p> <p> Select the page type. This affects wh webpage or is used for other purpos</p> <p>VALUE "123" was not available! ▾</p> </div>

Property:	Data type:	Description:
altLabels.[item_value]	string (LS)	<p><i>(applies to select-types only)</i></p> <p>This allows you to enter alternative labels for the items in the list.</p> <p>Example:</p> <pre>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.1.2 }</pre> <p>Result will be:</p> 
PAGE_TSCONFIG_ID	integer	<p><i>(applies to select-types with foreign table)</i></p> <p>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.</p>
PAGE_TSCONFIG_IDLIST	comma list of integers	<p><i>(applies to select-types with foreign table)</i></p> <p>See above</p>
PAGE_TSCONFIG_STR	string	<p><i>(applies to select-types with foreign table)</i></p> <p>See above</p>
itemsProcFunc.[...]	(custom)	<p><i>(applies to select-types with itemsProcFunc)</i></p> <p>The properties of this key is passed on to the itemsProcFunc in the parameter array by the key "TSconfig".</p>
RTEfullScreenWidth	int+/%	<p><i>(applies for RTE text fields only with the RTE wizard configured)</i></p> <p>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.</p>
linkTitleToSelf	boolean	<p><i>(all fields)</i></p> <p>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.</p> <p>Extra property:</p> <pre>.returnUrl = boolean; if set, then the return URL is also set.</pre> <p>Example:</p> <pre>TCEFORM.pages.title { # The label for the "title" field will link itself linkTitleToSelf = 1 # From the form with "title", return to full form: linkTitleToSelf.returnUrl = 1 }</pre> <p>The result is that the label for the title field will be a link:</p>

Property:	Data type:	Description:
		 <p>Clicking the link brings you to a form where only this field is shown:</p> 

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

->TSFE

Property:	Data type:	Description:	Default:
jumpUrl_transferSession	boolean	<p>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).</p> <p>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:</p> <pre>\$param = '&FE_SESSION_KEY=' . rawurlencode(\$GLOBALS['TSFE']->fe_user->id . '-', md5(\$GLOBALS['TSFE']->fe_user->id . '/', \$GLOBALS['TYPO3_CONF_VARS']['SYS']['encryptionKey']));</pre>	
constants	[TypeScript Frontend Constants defaults]	<p>Defaults for TypeScript Template constants!</p> <p>This features allows you to pass some amount of information (in the form of TypeScript 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.</p> <p>Example:</p> <pre>TSFE.constants.websiteConfig.id = 123</pre> <p>In the TypeScript templates you can now insert this constant as { \$websiteConfig.id }</p> 	

Property:	Data type:	Description:	Default:
		<p>In the backend module (in the Web main module) you can reach the value by a few codelines like these:</p> <pre> \$PageTSconfig = t3lib_BEfunc::getPagesTSconfig(\$this->pObj->id); \$websiteID = \$PageTSconfig['TSFE.']['constants.']['websiteConfig.']['id']; </pre> <p>Notice: In the frontend the setting of default constants will only apply to a branch of the tree <i>if</i> 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.</p>	

[page:TSFE]