EXT: htmlArea RTE

Extension Key: **rtehtmlarea**

Copyright 2004-2006, stanislas.rolland(arobas)fructifor.ca, <stanislas.rolland(arobas)fructifor.ca>

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

Table of Contents

Introduction

What does it do?

The extension offers a Rich Text Editor or RTE. The extension offers the following features:

- support for Firefox 0.8+, Mozilla 1.3.1+ and Netscape 7.1+ on all platforms, and for IE5.5+ on Windows platforms;
- integration of TYPO3 image insertion and link insertion browsers, configurable color selector and user element insertion dialog;
- configuration through TYPO3 Extension Manager, Page and User TSConfig RTE properties; three default sets of Page and User TSConfig configuration settings for typical situations, advanced users or demo environments;
- integration with the translation facilities of TYPO3;
- integration of the DynamicCSS and InlineCSS htmlArea plugins providing CSS style selector boxes with style descriptors imported from an external CSS file;
- integration of the SpellChecker htmlArea plugin providing spell checking in many languages, with optional personal dictionaries for each backend user.;
- integration of ContextMenu, TableOperations, InsertSmiley, FindReplace, RemoveFormat, CharacterMap, QuickTag and Acronym htmlArea plugins; Acronym plugin not currently available in IE;
- anchor accessibility feature;
- hook on Lorem Ipsum wizard so that dummy content may be inserted when the editor is in wysiwyg mode;
- optional configurable server-side HTML cleaning when content is pasted into the editor;
- a class that may be used in front end extensions to enable rich text editing of text fields.

Requirements

Version 1.3.0+ of this extension requires TYPO3 4.0.0+.

Version 1.2.0 is the last version of this extension that will run with TYPO3 3.8.0.

Version 1.1.3 is the last version of this extension that will run with TYPO3 3.7.0.

Version 1.0.1 is the last version of this extension that will run with TYPO3 3.6.0.

In TYPO3 4.0, the extension is included as system extension.

If the spell checking is enabled, version 1.1.0 (or more recent) of the Static Info Library extension (sr_static_info) is required.

The spell checker feature requires GNU Aspell 0.60+ to be installed on the server.

The spell checker requires PHP to be compiled with mbstring, mb_regex and pspell. If PHP is not compiled with pspell, the spell checker will function in shell_exec mode, if safe_mode is not enabled. However, the personal dictionaries feature is available only if safe_mode is not enabled.

The hook on the Lorem Ipsum wizard requires version 1.1.0+ of the Lorem Ipsum extension (lorem_ipsum).

Languages

The extension is currently available in Brazilian, Czech, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Russian, Slovak, and Spanish.

Credits

The open source htmlArea editor was originally developed by Mihai Bazon and sponsored by interactivetools.com. A number of htmlArea plugins were contributed by various authors. Both the base editor and the plugins have been substantially modified for the purpose of this extension.

This extension is a continuation of the work initiated by Kasper Skårhøj and Philipp Borgmann who realized the initial integration using the RTE API introduced in TYPO3 3.6.0 as well as the initial integration of the TYPO3 image insertion and link insertion wizards.

Thanks to the people at http:www.cablan.net for their help and support in understanding and debugging the interfaces and configuration issues of the various versions of Aspell.

Thanks to Jan Wulff for analyzing the issues related to Internet Explorer and mod_gzip on Apache servers.

The icons of the anchor accessibility feature are provided by creator and copyright-owner Cornelia Lange (http://www.clkm.de/).

Thanks to Pierre-Yves Gadina for the improved icons of the Insert Image button and of the Table Operations tool bar.

Thanks to Rob De Vries for the RemoveFormat icon.

The initial toggleborders button logic and the automatic url and email recognition logic originate from Xinha, http://xinha.gogo.co.nz/, a fork of htmlArea.

JavaScript scripts are compressed using open source ESC authored by Saltstorm.

The editor uses an adaptation of open source Event Cache script authored by Mark Wubben.

When performing paste operations with Mozilla/Firefox, the extension may trigger the installation of Mozilla/Firefox extension AllowClipboard Helper developed by Petr Simek.

License

The original htmlArea editor was released under so-called “modified BSD” license. The original code was substantially modified. As permitted by the modified-BSD license, and with the permission of the original copyright holders, the modified version included in this extension is released, like all TYPO3 components, under the GNU General Public License.

Sponsors

Development of this extension was sponsored in part by the TYPO3 Association and by: Carla Froitzheim, Ute Luft, Ralf Mertes, Peter Niederlag, Jost Reckmann, Pierre Rouwens, Krystian Szymukowicz, Veldhuizen Interactive, Sacha Vorbeck, Jan Wischnat.

You may sponsor further development of this extension by sending your contribution to stanislas.rolland@fructifor.ca through moneybookers.com or PayPal, or by contacting the author.

Support

You may get support in the use of this extension by subscribing to news://news.netfielders.de/typo3.projects.rte.

Users manual

FAQ: How do I insert a line break instead of a paragraph?

Using the Enter key inserts a paragraph (<p>, <hx> or <pre>). Using shift-Enter will insert a line break (<br />).

You may also disable RTE Page TSConfig property EnterParagraphs. Then, Mozilla/Firefox browsers will behave in their native mode and pressing the Enter key will insert a br tag. However, Internet Explorer will continue to insert paragraphs on pressing the Enter key.

FAQ: How do I remove a link?

To remove a link, select the link and click on the “Insert/Modify link” button. At the top of the popup window, you have the option to remove the link.

FAQ: How do I delete a table?

You may proceed as follows:

- click in any cell in the table;
- in the editor status bar, displayed at the bottom of the editor frame, click on “table”;
- press the “Delete” key or the “Backspace” key.

You may also proceed as follows:

- click in any cell in the table;
- click on the right button of the mouse or pointing device;
- the context menu is displayed;
- at the bottom of the context menu, you have the option to delete the TABLE element.

In Internet Explorer, you may also proceed as follows:

- put the cursor just after the table and press the “Backspace” key;
- or click on the border of the table and press the “Delete” key.

FAQ: How do I configure the editor to use my CSS styles?

In htmlArea RTE, the following style sheets apply to the contents of the editing area and are linked in the following order:

1. the htmlarea-edited-content.css file from the skin in use; it contains selectors for use in the editor, but not intended to be applied in the frontend;
2. a css file generated from the mainStyleOverride and inlineStyle assignments in Page TSConfig;
3. the css file specified by property contentCSS in Page TSConfig: you may define the styles you want to use in an external CSS file and assign the file name to this property.

An additional Page TSConfig property is recognized by htmlArea RTE: ignoreMainStyleOverride. If set, htmlArea will ignore the mainStyleOverride and inlineStyle sets of assignments and will use only files 1 and 3 mentioned above. This allows the Classic RTE to apply the css definitions defined in Page TSConfig and the htmlArea RTE to use the external css file (presumably containing the same or similar css definitions).

FAQ: Why does the text style selection list remain empty?

This selection list should display the classes specified by RTE.default.classesCharacter. The specified classes must be defined in the CSS style sheet contained in the file specified by RTE.default.contentCSS.

If RTE.default.classesCharacter is empty, classes associated with the span tag are used. Classes not associated with any tag in the contentCSS style sheet will be displayed only if

```
RTE.default.showTagFreeClasses = 1
```

FAQ: How can I define custom bullet images?

You can define custom bullet images in your style sheet.

For example, you may define the following styles:

```
ul { list-style-type: disc; list-style-image: url(/media/bullets/bullet1_n.gif); list-style-position:
outside; }
li { padding-left:2px; }
```

FAQ: Is it possible to style the appearance of links in the RTE itself?

«In the front end, all links are rendered correctly, but in the RTE itself the links are shown in standard color (blue) with underline, except links I've already visited. These are in black with underline and a hover effect. If I add new links, it's the same: Visited links become black, unvisited blue. The rest of the style sheet is parsed without problems. Any hints?»

This is a Mozilla/Firefox feature. In the editing area, the link attributes defined in your profile take precedence over the corresponding attributes specified in your style sheet. Apparently, these preferences cannot be neutral.

In a editing area displayed by Internet Explorer, the style sheet specification is applied to the link.

FAQ: How can I use a Bengali Open Type font in the editor?

Since Bengali is not well supported by all browsers, the recommended approach would be to add the Bengali font in the list of font families specified on the body selector of the RTE.default.contentCSS stylesheet. For example:

```
body { font-family: Verdana, sans-serif, Likhan; }
```

For some reason, with some fonts, the lines may overlap when using larger font sizes. It is the case with the Bengali Likhan font in Firefox 1.0.2. This may also be corrected through the stylesheet. For example:

```
body { font-family: Verdana, sans-serif, Likhan; line-height: 1.4; }
```

Note that, when using the Bengali Likhan font, a line-height with em or % unit may not produce any effect in Firefox 1.0.2.

FAQ: How do I configure the RTE to keep my custom tag?

Add the following lines to your Page TSConfig:

```
RTE.default.proc.allowTags = list-of-allowed-tags
RTE.default.proc.HTMLparser_rte.tags.mycustomtag.protect =1
```

where

list-of-allowed-tags is the list of all allowed tags, including your custom tag, and

mycustomtag is the name of your tag.

FAQ: Why is the class attribute on table tags always rendered as contenttable in the front end?

«Typo3 always replaces the class I selected in the RTE for a table with the class "contenttable". Do you have an idea how to switch that off?»

Assuming that you have installed extension CSS Styled Content (css_styled_content), add the following line in your TS template Setup field:

```
lib.parseFunc_RTE.externalBlocks.table.stdWrap.HTMLparser.tags.table.fixAttrib.class.list >
```

The contenttable class will then be added only if no class is specified for the table.

FAQ: Why are abbr and acronym tags not correctly rendered in the front end?

Make sure that abbr and acronym are included in the list:

```
styles.content.links.allowTags
```

in your TS template constants.

FAQ: What are htmlArea plugins?

Just as TYPO3 may be extended, the htmlArea editor provides a mechanism by which new features may be added without changing the core of the editor.

Each of the htmlArea plugins integrated in the present TYPO3 extension may be enabled/disabled by proper configuration of the extension.

The available plugins are the following:

- DynamicCSS: adds to the tool bar a CSS style selection list; the selected style is applied to the currently selected paragraph; the styles are defined in an external CSS file identified by configuration properties of the extension;
- InlineCSS: adds to the tool bar a CSS style selection list; the selected style is applied to the currently selected text; the styles are defined in an external CSS file identified by configuration properties of the extension;
- SpellChecker: adds to the tool bar a button that allows to check the spelling of the contents of the editor; some configuration properties of the extension apply specifically to this feature;
- ContextMenu: enables a contextual menu that is activated by clicking on the right button of the mouse;
- TableOperations: adds to the tool bar a set of buttons allowing to manipulate tables; the “table” button must also be enabled;
- InsertSmiley: adds to the tool bar a button that allows to insert a so-called “emoticon”;
- FindReplace: adds to the tool bar a button that allows to find and replace a specified string within the edited content;
- RemoveFormat: adds to the tool bar a button that offers format cleaning options; particularly useful to clean text pasted in from Microsoft Word;
- CharacterMap: adds to the tool bar a button that allows to insert a special character;
- QuickTag: adds to the tool bar a button to insert a tag to be applied to the currently selected text;
- TYPO3Browsers: replaces the original htmlArea popup windows for image and link insertion with the TYPO3 image and link internal browsers; note that this plugin is not available when the editor is used in the TYPO3 front end;
- SelectColor: replaces the original htmlArea color selector with one in which colors may be configured with RTE.default Page TSConfig properties;
- UserElements: adds to the tool bar a button to insert a custom tag or a shorthand; note that this plugin is not available when the editor is used in the TYPO3 front end; the custom elements may be maintained with extension Custom Tags (extension key: de_custom_tags);
- Acronym: adds to the tool bar a button to insert/edit acronyms and abbreviations; acronyms and abbreviations may be pre-defined with records of type Acronym inserted on the root of the TYPO3 backend;
- TYPO3HtmlParser: enables configurable server-side html cleaning when content is pasted into the editor; the cleaning operation is then performed on the server.

FAQ: Why are popup windows too small when they open?

We have tried to make sure that all popup windows open with usable dimensions. Since the contents of the windows may be variable and may be localized, it is difficult to ensure that all windows will open with the correct size in all languages. Therefore, the editor will try to adjust the width and height of the windows to their specific contents and to center the windows on the screen.

In some OS/browser configurations, the editor may be prevented from resizing and moving the windows due to lack of privileges:

- If you use a Firefox browser, select through the browser menu Tools -> Options -> Web Characteristics -> Javascript -> Advanced, and check the options that allows scripts to Hide the status bar and that allow scripts to resize and move windows. Popup windows should then open with the correct size.
- If you use IE6 on Windows XP SP2, select through the browser menu Tools -> Internet Options -> Security -> Trusted Sites, and add the url of TYPO3 site to the list of trusted sites. Popup windows should then open with the correct size.

FAQ: Why is the editor not displayed with full width when I use the full window wizard?

If you are editing the most usual content element, that is the bodytext column of tt_content table, try to add the following line to your Page TSConfig:

```
TCEFORM.tt_content.bodytext.RTEfullScreenWidth= 100%
```

Note that this setting is now included in the default configuration of the extension.

If editing some other column, use the same model:

```
TCEFORM.my_table_name.my_column_name.RTEfullScreenWidth= 100%
```

FAQ: Is it possible to disable the “New magic image”?

You may disable the magic image option in the TYPO3 «Insert/modify image» popup window with the following statement in Page TSConfig:

```
RTE.default.blindImageOptions = magic
```

Other options that may be added to this list are: plain and dragdrop.

FAQ: Why are images not displayed in the RTE?

If you are using Mozilla or Firefox, you may have set the preference not to display images or to display images only from the originating site.

FAQ: Why do style selector boxes remain disabled in IE?

When using IE, make sure that the browser cache setting is set to Automatic. In particular, with recent versions of Windows xp sp2, do not use the setting “On every visit to the page”.

FAQ: Why can't I get all buttons to be displayed?

Try the following:

1. User TSConfig restricts the user to a specified set of buttons; therefore, for access to all buttons without restriction, in User TSConfig, set: options.RTEkeyList = *
2. Page TSConfig adds the buttons required to edit the table and field you wish to edit; therefore, to add all buttons by default, in Page TSConfig, set RTE.default.showButtons = *
3. If you are trying to edit the bodytext field of a content element from table tt_content, then the TCA field types and palettes may specify a list of buttons to add; this specification takes precedence over RTE.default.showButtons ; to override any such setting in TCA for the bodytext field of table tt_content, in Page TSConfig, set RTE.config.tt_content.bodytext.showButtons = *
4. If you are trying to edit a text field from another table, then, in Page TSConfig, set RTE.config.tableName.columnName.showButtons = *

If you have not modified $GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['rtehtmlarea']['HTMLAreaPluginList'], all buttons should now be displayed (except perhaps the spellchecker button if some requirements are not met).

FAQ: Why does it take so long to load the editor in Internet Explorer?

Try to adjust your IE browser cache settings:

1. From the IE main menu, navigate to: Tools -> Internet Options -> General -> button: Configure...   or in some other IE versions: Extras -> Internet Options : Temporary Files -> button: Advanced
2. Select the radio button Automatic.

Some server configuration settings may also help working around Internet Explorer caching problems. See the Server Configuration section and the Tutorial section of this document.

FAQ: Can I use both the htmlArea RTE and the classic RTE?

You can install both RTE extensions, in fact, as many RTE's has you wish. Their availability will be checked in the order in which they are loaded, that is in the order they are installed.

If you install first the classic RTE and, secondly, htmlArea RTE, a user with IE 5+ on Windows will be presented the classic RTE; a user with a supported version of Mozilla 1.3.1+, Firefox 0.8+ or Netscape 7.1+ on any platform will be presented the htmlArea RTE.

If you install htmlArea RTE before you install the classic RTE, htmlArea RTE will be presented to users with IE 5.5+ on Windows, Mozilla 1.3.1+, Firefox 0.8+ or Netscape 7.1+; the classic RTE will only be presented to users with IE 5 on Windows.

FAQ: Can htmlArea RTE be used in the front end?

If you are a TYPO3 front end extension developer, you can use class tx_rtehtmlarea_pi2 to enable rich text editing of text fields. See extension Modern Guestbook Extended (sr_guestbook) for an example of how to do it. Note that the TYPO3Browsers , UserElements and the Acronym plugins are not available when the editor is used in the TYPO3 front end.

FAQ: What hotkeys are available in htmlArea RTE?

The following hotkeys are available:

ctrl+A	:	Select all
ctrl+B	:	Bold (if the corresponding button is configured in the toolbar of the RTE)
ctrl+I	:	Italic (if the corresponding button is configured in the toolbar of the RTE)
ctrl+U	:	Underline (if the corresponding button is configured in the toolbar of the RTE)
ctrl+S	:	Strike Through (if the corresponding button is configured in the toolbar of the RTE)
ctrl+L	:	Justify Left (if the corresponding button is configured in the toolbar of the RTE)
ctrl+E	:	Justify Center (if the corresponding button is configured in the toolbar of the RTE)
ctrl+R	:	Justify Right (if the corresponding button is configured in the toolbar of the RTE)
ctrl+J	:	Justify Full (if the corresponding button is configured in the toolbar of the RTE)
ctrl+Z	:	Undo (if the corresponding button is configured in the toolbar of the RTE)
ctrl+Y	:	Redo (if the corresponding button is configured in the toolbar of the RTE)
ctrl+X	:	Cut
ctrl+C	:	Copy
ctrl+V	:	Paste
ctrl+N	:	Format Block as Paragraph (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+0	:	Clean Word
ctrl+1	:	Format Block as Heading 1 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+2	:	Format Block as Heading 2 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+3	:	Format Block as Heading 3 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+4	:	Format Block as Heading 4 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+5	:	Format Block as Heading 5 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+6	:	Format Block as Heading 6 (if the corresponding select box is configured in the toolbar of the RTE)
ctrl+space:		Insert a non-breaking space

FAQ: How do I configure the anchor accessibility feature?

When the TYPOBrowsers plugin is enabled, you may attach special accessibility features to CSS classes when they are applied to anchors with the TYPO3 link browser.

The classes should first be defined in the CSS file specified by RTE.default.contentCSS.

The classes should be part of the list specified by property RTE.default.classesAnchor.

The accessibility features attached to the classes are specified by property RTE.classesAnchor.

Configuration

Step 1: Extension installation

Use the Extension Manager to install the extension.

The extension may be installed as a global or local extension.

You may be requested to uninstall the following extension: rte_conf.

If you plan to use the SpellChecker plugin, you should install extension sr_static_info. The spell checker feature requires GNU Aspell 0.60+ to be installed on the server.

Custom elements presented by the UserElements plugin may be maintained with extension Custom Tags (extension key: de_custom_tags).

Note that the installation dialog will request to create table tx_rtehtmlarea_acronym; this table is used by plugin Acronym.

Upon installation directory uploads/tx_rtehtmlarea will be created. Personal dictionaries are stored in subdirectories of this directory.

Upon installation, if RTE has not yet been enabled with the TYPO3 Install tool, it will be automatically enabled:

```
$TYPO3_CONF_VARS['BE']['RTEenabled'] = 1;
```

Step 2: Extension configuration variables

The Extension Manager installation dialog allows to set the following extension configuration variables:

- **Default configuration settings:** let you choose a set of default Page TSConfig and User TSConfig settings; select **Typical (Most commonly used features are enabled. Select this option if you are unsure which one to use.)** for the typical site requirements; select **Minimal (Most features disabled. Administrator needs to enable them using TypoScript. For advanced administrators only.)** for minimal settings; select **Demo (Show-off configuration. Includes pre-configured styles. Not for production environments.)** to explore some of the available features; default value is Typical;
- **Enable images in the RTE**: if this boolean variable is set, the use of images in the "Typical" default configuration of the RTE will be enabled.
- **Maximum width of plain images:** Maximum width in pixels of plain images inserted with the TYPO3 image browser; default value is 640;
- **Maximum height of plain images:** Maximum height in pixels of plain images inserted with the TYPO3 image browser; default value is 680;
- **Enable Mozilla/Firefox extension:** if this boolean variable is set, the triggering of installation of a browser extension to allow the RTE to access the clipboard will be enabled; the feature will trigger the installation of Mozilla/Firefox extension AllowClipboard Helper from mozilla.org; default value is 0;
- **Url of AllowClipboard Helper**: the full absolute Url of the AllowClipboard Helper extension for Mozilla/Firefox;
- **Enable troubleshooting mode**: if this boolean variable is set, a troubleshooting log will be displayed below the editing area; default value is 0;
- **Enable compressed scripts:** if this boolean variable is set, the compressed version of the editor scripts will be used; default value is 1; disabling compressed scripts is useful only in troubleshooting mode;
- **Force Aspell command mode:** if this boolean variable is set, the Aspell command interface will be used; this may be useful when PHP is compiled with pspell, but with an old version of Aspell, and a newer version is available in another directory; if set, PHP safe mode should NOT be enabled; default value is 0;
- **List of htmlArea plugins:** this is the list of htmlArea plugins that may be enabled; default value is "TableOperations, SpellChecker, ContextMenu, SelectColor, TYPO3Browsers, InsertSmiley, FindReplace, RemoveFormat, CharacterMap, QuickTag, InlineCSS, DynamicCSS, UserElements, Acronym, TYPO3HtmlParser"; note that plugins TYPO3Browsers, UserElements, Acronym and TYPO3HtmlParser are not available when the editor is used in the TYPO3 front end.

If the SpellChecker is not enabled, then the remaining configuration variables are irrelevant; note that if extension sr_static_info is not installed, SpellCheker will not be enabled.

- **List of Aspell dictionaries:** the list of dictionaries available to the htmlArea SpellChecker plugin; this list is used only if safe_mode is enabled; if safe_mode is not enabled, the list is automatically obtained from Aspell: default value is "en";
- **Default Aspell dictionary:** the default dictionary to be used by the htmlArea SpellChecker plugin; this should be set to the default language of the site; default value is "en";
- **No spell checking languages**: the list of languages for which Aspell does not provide spell checking (see Unsupported Languages) and for which the htmlArea SpellChecker plugin will therefore be disabled (not shown in the RTE tool bar); default value is "ja, km, ko, lo, th, zh, b5, gb";
- **Aspell directory:** the server directory in which Aspell is installed; default value is "/usr/bin/aspell".

Step 3: User TSConfig

Upon installation, the extension will set default properties in User TSConfig as specified by the extension configuration variable: **Default configuration settings**. Three default configurations are available: Minimal, Typical, and Demo. These default configurations are documented in the next section of the present document.

These properties may be modified for any particular BE user or BE user group, with the Typo3 User Admin Tool. Properties of User TSConfig are documented in this document: http://typo3.org/documentation/document-library/doc_core_tsconfig/User_TSconfig/

Property:	Data type:	Description:
setup.default.edit_RTE	boolean	Specifies that RTE editing should be enabled or disabled by default for all users.
options.RTEkeyList	list of id-strings	Specifies the list of RTE buttons to which the BE user or BE user group is restricted. Default: * (means all) Note: For the list of possible buttons, see property showButtons of Page TsConfig.
options.HTMLAreaPspellMode (htmlArea RTE only)	string	Specifies the mode of spelling suggestions. Possible values are: ultra, fast, normal or bad-spellers. Default: normal Note: For more information on spelling suggestions modes, see Notes on the Different Suggestion Modes).
options.enablePersonalDicts (htmlArea RTE only)	boolean	Enables the personal dictionaries feature for the user or user group, when the feature is enabled in Page TSConfig. Default: 0 Note: The option is ignored if PHP safe_mode is enabled. Note: The feature must also be enabled in Page TSConfig.
options.uploadFieldsInTopOfEB	boolean	Inserts a file uploader on the File tab of the Insert/Modify link dialog, when TYPO3Browsers is enabled. Default: 0
options.RTESmallWidth options.RTESmallHeight (htmlArea RTE only)	pixels	Width and height of the editor area when edit_wideDocument is NOT set. Default: RTESmallWidth = 530, RTESmallHeight = 380
options.RTELargeWidthIncrement options.RTELargeHeightIncrement (htmlArea RTE only)	pixels	Increments applied to the width and height of the editor area when edit_wideDocument IS set. Default: RTELargeWidthIncrement= 150, RTELargeHeilghtIncrement = 0

Step 4: Page TSConfig

Upon installation, the extension will set default properties in Page TSConfig as specified by the extension configuration variable: **Default configuration settings**. Three default configurations are available: Minimal, Typical, and Demo. These default configurations are documented in the next section of the present document.

These properties may be modified in the PageTS Config of your page; this is most often done on the root page of your site.

classes:

Properties of each class available in the RTE.

Property:	Description:

classes.[*classname*]	Defines the classes available in the RTE. *classname* is the actual name of the style-class you're configuring. Notice you must specifically assign the classes to the various facilities also. See later.
	
	Properties:
	.name = Label of the class
	.value = The style for the class
	.noShow = Boolean; if set, the style of the class is not used to render it in the pop-up selector.
	
	Example:
	`# General configuration of the available classes:`
	`RTE.classes {`
	`  mainBodyText {`
	`    name = Body Text (Primary)`
	`    value = font:bold; color:blue;`
	`  }`
	`}`
	`# Specific configuration for the Character/Text Style menu:`
	`RTE.default.classesCharacter = mainBodyText`

[page:RTE]

colors:

Properties of each color available in the RTE.

Property:	Description:

colors.[*id-string*]	Defines the colors available in the RTE.
	
	Properties:
	.name = Label of the color in menu
	.value = The HTML-color value
	
	Example:
	`# General configuration of the available colors:`
	`RTE.colors {`
	`  color1 {`
	`    name = Background color`
	`    value = blue`
	`  }`
	`  color2 {`
	`    name = Another color I like!`
	`    value = #775533`
	`  }`
	`  noColor {`
	`    name = No color`
	`    value =`
	`  }`
	`}`
	`# Specific setting for the font color selector:`
	`RTE.default.colors = color1, color2, noColor`

[page:RTE]

fonts:

Properties of each font available in the RTE.

Property:	Description:

fonts.[*id-string*]	Defines the fonts available in the RTE.
	
	Properties:
	.name = Label of the font in menu
	.value = The font face value
	
	Example:
	`# General configuration of the available fonts:`
	`RTE.fonts {`
	`  face1 {`
	`    name = Verdana`
	`    value = verdana, arial`
	`  }`
	`  face2 {`
	`    name = Comic Sans`
	`    value = Comic Sans MS`
	`  }`
	`  noFace {`
	`    name = No font`
	`    value =`
	`  }`
	`}`
	`# Specific setting for the fontstyle selector:`
	`RTE.default.fontFace = face2 , face1, noFace`

[page:RTE]

RTE interface configuration objects:

These objects contain the actual configuration of the RTE interface.

Property:	Description:
default.[...] config.[*tablename*].[*field*].[...] config.[*tablename*].[*field*].types.[*type*].[...]	These objects contain the actual configuration of the RTE 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 fields/types. 'RTE.default' configures the RTE for all tables/fields/types 'RTE.config.[*tablename*].[*field*]' configures a specific field. The values inherit the values from 'RTE.default' in fact this is overriding values. 'RTE.config.[*tablename*].[*field*].types.[*type*]' configures a specific field in case the 'type'-value of the field matches *type*. Again this overrides the former settings.

[page:RTE]

RTE interface configuration properties:

These properties may be set for each RTE interface configuration object.

Property:	Data type:	Description:
disabled	boolean	If set, the editor is disabled.

classesParagraph	list of id-strings	Classes available in the Paragraph Style selector.
		The Paragraph Style selector lets you format a block of content also known typographically as a *paragraph* (the section between two linebreaks). The content of the paragraph is wrapped in block-tags and by default you can select between <Hx>, <P> and <PRE> formatting (you can disable any or all of these default values by the 'hidePStyleItems', see later).
		
		Example:
		This configuration would result in this Paragraph style selector (provided that the classes 'mainBodyText' and 'header2' are found in the RTE.classes definition, see above):
		
		`RTE.default {`
		`  classesParagraph = mainBodyText, header2`
		`}`
		
		In the case of htmlArea RTE, the classes must be defined either in one of the occurrences of the inlineStyle property or in the CSS file specified contentCSS property.
		
		In the case of the Classic RTE, the classes defined in the 'classesParagraph' value are prepended to the types of paragraphs options. They are inserted as <DIV>-tags with a class-parameter.

Property:	Data type:	Description:
classesCharacter	list of id-strings	Classes available for the Text Style selector. The Character Style selector is used to format any string of text inside a paragraph, tablecell. So this formatting relates to the selected piece of text in opposition to the Paragraph Styles which always formats the current paragraph no matter the selection. In the case of htmlArea RTE, the classes must be defined either in one of the occurrences of the inlineStyle property or in the CSS file specified contentCSS property. In the case of htmlArea RTE, if the property is empty, classes associated with the span tag in either the inlineSyle properties or in the contentCSS file are used. In the case of the Classic RTE, although the Character Style selector is aimed at text-formatting, it will also apply classes to other elements, for instance tables or images if selected as well as links and table-cells. The allowed classes for these operations may be configured separately by listing the classes with the options, you find below.
showTagFreeClasses (htmlArea RTE only)	boolean	Specifies that CSS classes not associated with any tag in the contentCSS style sheet should be or should not be shown in the paragraph style and the text style selection lists. Default: 0
disablePCexamples	boolean	If set, the examples of classes in Paragraph and Character selectors are disabled. Note: In the case of htmlArea RTE, the styling is removed on the options of the Paragraph and Text style selector lists.
disableTYPO3Browsers (htmlArea RTE only)	boolean	If set, the TYPO3 element and image browsers are disabled. Default: 0 Note: If set, the htmlArea plugin TYPO3Browsers is disabled. Note: The TYPO3Browsers plugin is not available when the editor is used in the TYPO3 front end.
classesImage	list of id-strings	Classes available for images. In the case of htmlArea RTE, each of the listed class must be defined either in one of the occurrences of the inlineStyle property or in the CSS file specified contentCSS property. In the case of htmlArea RTE, the same list of classes is used in the class selection list of the Insert/Modify image dialog. In the case of the Classic RTE, see description of 'classesCharacter'.
blindImageOptions	list of strings	List of tab items in the image selector to remove. Keylist is magic,plain,dragdrop

classesAnchor
(htmlArea RTE only)
	list of id-strings	Classes available in the Insert/Modify link dialog.
		These classes must be defined by the RTE.classesAnchor property.
		The following property specifies the default class of the class selector of the corresponding tab:
		`.default {`
		`      page = className`
		`      url = className`
		`      file = className`
		`      mail = className`
		`}`

classesLinks	list of id-strings	Classes available for links. See description of 'classesCharacter'. In the case of htmlArea RTE, classesAnchor offers more options.
defaultLinkTarget	string	This sets the default target for new links in the RTE.
blindLinkOptions	list of strings	List of tab items in the link selector to remove. Keylist is page,file,url,mail,spec

Property:	Data type:	Description:
classesTable	list of id-strings	Classes available for tables. See description of 'classesCharacter'. In the case of htmlArea RTE, each of the listed class must be defined either in one of the occurrences of the inlineStyle property or in the CSS file specified contentCSS property. In the case of htmlArea RTE, the same list of classes is used in the class selection list of the Table Properties dialog.
classesTD	list of id-strings	Classes available for tablecells. See description of 'classesCharacter'. In the case of htmlArea RTE, each of the listed class must be defined either in one of the occurrences of the inlineStyle property or in the CSS file specified contentCSS property. In the case of htmlArea RTE, the same list of classes is used in the class selection list of the Cell Properties dialog.

colors	list of id-strings	Defines the specific colors generally available in the color selectors. The id-strings must be configured in the RTE.colors array (see description earlier).
		
		Example:
		`RTE.default {`
		`  colors = color1, color2,noColor`
		`}`

disableColorPicker	boolean	Disables the color picker matrix in all color dialogs. The color picker lets you select web-colors. Note: In the case of htmlArea RTE, this applies only when the TYPO3 configurable color selection dialog is enabled (plugin SelectColor).
disableSelectColor (htmlArea RTE only)	boolean	If set, the TYPO3 configurable color selection dialog is disabled. Default: 0 Note: If set, the htmlArea plugin SelectColor is disabled.
hideFontFaces	list of id-numbers, * removes all	Lets you disable any of the default font faces in the Font Style selector. These are the possible values you can set: 1: Arial 2: Arial Black 3: Verdana 4: Times New Roman 5: Garamond 6: Lucida Handwriting 7: Courier 8: Webdings 9: Wingdings
fontFace	list of id-strings	Defines the specific fonts generally available in the font selector. The id-strings must be configured in the RTE.fonts array (see description earlier).
hideFontSizes	list of size-numbers, * removes all	Lets you disable any of the default font sizes available in the Font Size selector. Values are ranging from 1-7.
showButtons hideButtons	list of id-strings	Note: In the case of htmlArea RTE, available buttons are: blockstylelabel, blockstyle, textstylelabel, textstyle, fontstyle, fontsize, formatblock, bold, italic, underline, strikethrough, subscript, superscript, lefttoright, righttoleft, left, center, right, justifyfull, orderedlist, unorderedlist, outdent, indent, textcolor, bgcolor, textindicator, emoticon, insertcharacter, line, link, image, table, user, acronym, findreplace, spellcheck, chMode, inserttag, removeformat, copy, cut, paste, undo, redo, showhelp, about, toggleborders, tableproperties, rowproperties, rowinsertabove, rowinsertunder, rowdelete, rowsplit, columninsertbefore, columninsertafter, columndelete, columnsplit, cellproperties, cellinsertbefore, cellinsertafter, celldelete, cellsplit, cellmerge Note: In the case of htmlArea RTE, if extension sr_static_info is not installed, the spellcheck button is not enabled. Note:In the case of htmlArea RTE, the following elements will always be shown: undo redo, about. Note: In the case of htmlArea RTE, buttons user and acronym are never available in the front end. Note: In the case of htmlArea RTE, button Acronym is not currently available in IE.

Property:	Data type:	Description:
toolbarOrder (htmlArea RTE only)	list of id-strings	Specifies the order and grouping of buttons in the RTE tool bar. The keywords space, bar and linebreak may be used to insert a space, a separator or a line break at the corresponding position in the tool bar. Default: blockstylelabel, blockstyle, space, textstylelabel, textstyle, bar, linebreak, fontstyle, space, fontsize, space, formatblock, bar, bold, italic, underline, bar, strikethrough, subscript, superscript, bar, lefttoright, righttoleft, bar, left, center, right, justifyfull, bar, orderedlist, unorderedlist, outdent, indent, bar, textcolor, bgcolor, textindicator, bar, emoticon, insertcharacter, line, link, image, table, user, acronym, bar, findreplace, spellcheck, bar, chMode, inserttag, removeformat, bar, copy, cut, paste, bar, undo, redo, bar, showhelp, about, linebreak, toggleborders, bar, tableproperties, bar, rowproperties, rowinsertabove, rowinsertunder, rowdelete, rowsplit, bar, columninsertbefore, columninsertafter, columndelete, columnsplit, bar, cellproperties, cellinsertbefore, cellinsertafter, celldelete, cellsplit, cellmerge
keepButtonGroupTogether (htmlArea RTE only)	boolean	Specifies that all buttons of a button group are displayed on the same line of the tool bar. A button group is delimited by a linebreak or by a bar. Default: 0 Note: If enabled, the setting is honored only by Mozilla/Firefox. It is ignored when the browser is Internet Explorer, Safari, Opera or Mozilla 1.3.
hideTableOperationsInToolbar (htmlArea RTE only)	boolean	Specifies that table operations buttons should be hidden in the tool bar or not. Default: 0 Note: If enabled, table operations will appear only in the context menu, provided that they may be enabled in the given context.
keepToggleBordersInToolbar (htmlArea RTE only)	boolean	Specifies that the toggleborders button should be kept in the tool bar or not. Default: 0 Note: If enabled, the toggleborders button will be kept in the tool bar even if the hideTableOperationsInToolbar is enabled.
disableContextMenu (htmlArea RTE only)	boolean	Specifies that the context menu should be disabled or not. Default: 0 Note: This property is a synonymous of disableRightClick property.
disableRightClick	boolean	If set, the right click (context) menu of the RTE is disabled.
showStatusBar (htmlArea RTE only)	boolean	Specifies that the editor status bar should be displayed or not. Default: 0

Property:	Data type:	Description:

mainStyleOverride	string	By default the editor style section is set with the CSS-code below. However you may override this by this option.
		
		Default for Classic RTE:
		
		`BODY {`
		`  border: 1px black solid;`
		`  border-top: none;`
		`  margin : 2 2 2 2;`
		`  font-family:Verdana;`
		`  font-size:10px;`
		`  color:black;`
		`  background-color:white;`
		`}`
		`TD {font-family:Verdana; font-size:10px;}`
		`P {margin-top:0px; margin-bottom:5px;}`
		`DIV {margin-top:0px; margin-bottom:5px;}`
		`OL {margin: 5px 10px 5px;}`
		`UL {margin: 5px 10px 5px;}`
		`BLOCKQUOTE {margin-top:0px; margin-bottom:0px;}`
		
		Default for htmlArea RTE:
		
		`body.htmlarea-content-body {`
		`      font-family: Verdana,sans-serif;`
		`      font-size: 12px;`
		`      color: black;`
		`      background-color: white;`
		`}`
		`td { }`
		`div { }`
		`pre { }`
		`ol { }`
		`ul { }`
		`blockquote { }`
		
		Note: In the case of htmlArea RTE, see also property ignoreMainStyleOverride.

mainStyleOverride_add.[*key*]	string (css-style)	Allows to add style configuration to the values above (for .mainStyleOverride above) Keys are: .P / .DIV / .TD /.BODY / .BLOCKQUOTE /.OL / .UL / .PRE / .Hx will all take values which are prepended to the above settings. Note: In the case of htmlArea RTE, see also property ignoreMainStyleOverride.
mainStyle_font mainStyle_size mainStyle_color mainStyle_bgcolor	string	Setting the default font-family (verdana) , font-size (10px), font-color (black) and background color (white) The default is shown in ".mainStyleOverride" above. Note: In the case of htmlArea RTE, see also property ignoreMainStyleOverride.
inlineStyle.[any *keysting*]	string	CSS code to be included in the editor style section. This will be included *after* the default code. Note: In the case of htmlArea RTE, see also property ignoreMainStyleOverride.
skin (htmlArea RTE only)	resource	The skin contains the CSS files and the images used to style the editor. The skin is specified by specifying the location of the main CSS file to be used to style the editor. The folder containing the CSS file MUST also contain a structure of folders and files identical to the structure found in the folder of the default skin. All folder names and all file names must be identical. Default: EXT:rtehtmlarea/htmlarea/skins/default/htmlarea.css Note: Extensions sr_rtehtmlarea_bluelook and sr_rtehtmlarea_xpblue provide two additional example skins. Note: these example skins do not work in Mozilla 1.3; if the property is set to one of them, the default skin will be used when the browser is Mozilla 1.3. Note: See also property ignoreMainStyleOverride.

Property:	Data type:	Description:
contentCSS (htmlArea RTE only)	resource	The CSS file that contains the style definitions that should be applied to the edited contents. The selectors defined in this file will also be used by the DynamicCSS and InlineCSS plugins in the block style and text style selection lists. Default: EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/dynamiccss.css For example, this default could be overridden with: fileadmin/styles/my_contentCSS.css Note: See also property ignoreMainStyleOverride.
ignoreMainStyleOverride (htmlArea RTE only)	boolean	Instruct htmlArea RTE to ignore properties mainStyleOverride and inlineStyle. Default: 0 Note: In htmlArea RTE, the following style sheets apply to the contents of the editing area and are linked in the following order: 1. the htmlarea-edited-content.css file from the skin being used (contains selectors for use in the editor but not intended to be applied in the frontend); 2. a css file generated from the mainStyleOverride and inlineStyle assignments; 3. the css file specified by contentCSS in Page TSConfig. When ignoreMainStyleOverride is enabled, the middle style sheet is ignored and only the first and third style sheets are used.
proc	->PROC	Customization of the server processing of the content - also called 'transformations'. See http://typo3.org/documentation/document-library/core-documentation/doc_core_api/current/view/5/2/
enableWordClean (htmlArea RTE only)	boolean/ ->HTMLparser	Specifies that text pasted from external sources, presumably from Microsoft Word, should be “cleaned” or not. Default: 0 Note:If no HTMLparser configuration is specified, a limited default cleaning operation will be performed. If a HTMLparser specification is specified, parsing will be performed on the server at the time of the paste operation. Note: The HTMLparser configuration is ignored If TYPO3HtmlParser plugin is disabled. Note: Additional cleanup may be performed by the user when the removeformat button is enabled.
removeComments (htmlArea RTE only)	boolean	Specifies that html comments should be removed or not by the editor on save and on toggle to HTML source mode. Default: 0
removeTags (htmlArea RTE only)	list of tags	List of tags that should be removed by the editor on save and on toggle to HTML source mode.
removeTagsAndContents (htmlArea RTE only)	list of tags	List of tags that should be removed by the editor, contents included, on save and on toggle to HTML source mode. The tags and the contents inside the tags will be removed.
useCSS (htmlArea RTE only)	boolean	Specifies that Mozilla/Firefox should use style attributes or not. When enabled, Mozilla/Firefox use span tags with style attributes rather than tags such as b, i, font, etc. Default: 0
disableEnterParagraphs (htmlArea RTE only)	boolean	Specifies that the insertion of paragraphs when hitting the Enter key in Mozilla/Firefox, Safari or Opera should be disabled. Default: 0 Note: If NOT enabled, the behavior of Mozilla/Firefox, Safari and Opera is modified as follows: when the Enter key is pressed, instead of inserting a br tag, the behavior of Internet Explorer is simulated and a new paragraph is created. Note: If enabled, the behavior of Mozilla/Firefox, Safari and Opera is not modified: a br tag is inserted when the Enter key is pressed.

Property:	Data type:	Description:
removeTrailingBR (htmlArea RTE only)	boolean	Specifies that trailing br tags should be removed from block elements. Default: 0 Note: If set, any trailing br tag in a block element will be removed on save and/or change mode. However, multiple trailing br tags will be preserved. Note: In Mozilla/Firefox/Netscape, whenever some text is entered in an empty block, a trailing br tag is added by the browser.
hideTags (htmlArea RTE only)	list of tags	List of tags that should NOT be shown by the QuickTag plugin. Note: A list of tags or values among the following: a, a (full), abbr, acronym, address, b, big, blockquote, code, div, em, fieldset, font, font (full), h1, h2, h3, h4, h5, h6, i, legend, li, ol, ul, p, pre, small, span, strong, sub, sup, table, tt, q, cite, abbr, acronym
disableAlignmentFieldsetInTableOperations disableSpacingFieldsetInTableOperations disableColorFieldsetInTableOperations disableLayoutFieldsetInTableOperations disableBordersFieldsetInTableOperations (htmlArea RTE only)	boolean	Disables the corresponding fieldsets of the table operations dialogs. Default: 0
enablePersonalDicts (htmlArea RTE only)	boolean	Specify that personal dictionaries should be enabled or not. Default: 0 Note: The SpellChecker plgin must be enabled. Note: The option is ignored if PHP safe_mode is enabled on the server. Note: The feature must also be enabled in User TSConfig. Note: Personal dictionaries are stored in subdirectories of uploads/tx_rtehtmlarea
userElements.[#]	string/->userCategory	Configuration of the categories of user elements The string value sets the name of the category. Value is language-splitted (by |) to allow for multiple languages.
userLinks.[#]	string/->userLinks	Configuration of user defined links. The string value sets the name of the category. Value is language-splitted (by |) to allow for multiple languages.

[page:RTE.default/RTE.default.FE/RTE.config.(table).(field)/RTE.config.(table).(field).types.(type)]

classesAnchor:

The following property allows to configure the anchor accessibility feature:

Property:	Description:
classesAnchor.[*id-string*] (htmlArea RTE only)	Attaches special properties to the classes available in the Insert/Modify link dialog. Properties: .class = CSS-class-name: the name of the CSS class to which the properties are attached .type = page, url, file, mail or spec: specifies that the class applies to anchors for internal pages, external URL's, files, email addresses or special user-defined links respectively; the class will be presented only in the corresponding tab of the TYPO3 link browser .image = URL of an icon file that will prefix the content of the anchor when the class is applied to an anchor; the TYPO3 syntax EXT:extension-key/sub-directory/image-file-name may be used .altText = the text that will be used as altText for the image when the class is applied to an anchor; may be language-splitted; the TYPO3 syntax LLL:EXT:extension-key/sub-directory/locallang.php:label-index may also be used in order for the text to be localized to the language of the content using the specified language file and label index .titleText = the text that will be used as title for the anchor when the class is applied to an anchor; may be language-splitted; the TYPO3 syntax LLL:EXT:extension-key/sub-directory/locallang.php:label-index may also be used in order for the text to be localized to the language of the content using the specified language file and label index See the Demo default configuration for a complete example.

[page:RTE]

userCategory:

Properties of each user element category.

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th></tr>
<tr><td>load</td><td>string</td><td>If set, the a predefined set of user element is loaded into this category. They are always loaded in the key starting with 100 and then forward in steps of 10.
Current options are:
“images_from_folder”: Loads gif,jpg,jpeg,png images from the specified folder (defined by the .path property)</td></tr>
<tr><td>merge</td><td>Boolean</td><td>If set, then any manually configured user elements are merged onto the ones loaded by the .load operation.</td></tr>
<tr><td>path</td><td>String</td><td><i>(Applies for load=images_from_folder only)</i>

Sets the path of the folder from which to fetch the images (gif,jpg,jpeg,png)

<b>Example:</b>
.path = fileadmin/istate/</td></tr>
<tr><td>[#]</td><td>string/-
>userElements</td><td>Configuration of the user elements.
The string value is the name of the user element. Language-splitted.

<b>Example:</b>

```
RTE.default.userElements {
    # Category with various elements
  10 = Various elements | Diverse elements
  10 {
      # An image is inserted
    1 = Logo 1 | Bomærke 1
    1.description = This is the logo number 1. | Dette er logo nummer 1
    1.content = <img src="###_URL###fileadmin/istate/curro.png">

      # The text-selection is wrapped with <sup> tags.
    2 = Subscript
    2.description = Selected text is wrapped in <sup>-tags.
    2.mode = wrap
    2.content = <sup>|</sup>

      # This submits the selected text content to the script, rte_cleaner.php
    5 = Strip all tags
    5.description = All HTML-codes are removed from selection.
    5.mode = processor
    5.submitToScript = typo3/rte_cleaner.php
  }

    # Category with images from the fileadmin/istate/ folder
  2.load = images_from_folder
  2.merge = 1
  2.path = fileadmin/istate/
    # here the logo from "Various elements" is included as well
  2.1 < .10.1
}
  # Show the user-button, if not existing
RTE.default.showButtons = user
```

</td></tr>
</table>

[page:->userCategory]

userElements:

Properties of each user elements setup.

Property:	Data type:	Description:
mode	string	Which kind of object it is. Options: “wrap”: If a wrap, then the content is exploded by “|” and wrapped around the current text selection. “processor”: The content is submitted to the php-script defined by .submitToScript. GPvar(“processContent”) carries the selection content of the RTE and GPvar(“returnUrl”) contains the return url. (The “content” property is not used here!) default: The content is just inserted (pasted into) at the cursor or substituting any current selection.

Property:	Data type:	Description:
description	string	A short description shown beneath the user element title (which is in bold)
content	string	The content inserted/wrapped into the RTE

submitToScript	string	*(Applies only to mode=processor)*
		PHP script to which the current text selection of the RTE is submitted. The script must be relative to the site-url or a full url starting with http://...
		Example:
		`.submitToScript = typo3/rte_cleaner.php`
		or
		`.submitToScript = http://www.domain.org/some_extenal_script.php`

dontInsertSiteUrl	boolean	If set, the marker ###_URL### in the content property's content IS NOT substituted by the current site url. Normally you wish to do this for all image-references which must be prepended with the absolute url in order to display correctly in the RTE!

[page:->userElements]

userLinks:

Properties of user links

Property:	Data type:	Description:
url	string	The url. If set, the marker ###_URL### in the content property's content is substituted by the current site url.
description	string	A short description shown beneath the link title (which is in bold)
target	string	Default target (if isset())

[page:->userLinks]

Step 5: Default Website Language

If you enable the SpellChecker plugin, make sure you have created a Website Language record for the default language of your site, even if your TypoScript template does not refer to it.

Step 6: Server Configuration

Internet Explorer has caching problems that may affect the performance of the htmlArea RTE. These problems may be worked around with the following server configuration recommendations.

Apache configuration:

Add the following lines to your Apache httpd.conf file or in the .htaccess file of the root directory of your site:

```
BrowserMatch "MSIE" brokenvary=1
BrowserMatch "Mozilla/4.[0-9]{2}" brokenvary=1
BrowserMatch "Opera" !brokenvary
SetEnvIf brokenvary 1 force-no-vary

ExpiresActive On
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
```

The last two statements require the **mod_expires** Apache module to be installed. For information on this module, see:

http://httpd.apache.org/docs/1.3/mod/mod_expires.html

Microsoft IIS configuration:

See: http://www.aspnetresources.com/blog/cache_control_extensions.aspx

More information:

For more information on this subject, see the following articles:

http://dean.edwards.name/my/flicker.html

http://httpd.apache.org/docs/1.3/mod/mod_expires.html

http://fivesevensix.com/studies/ie6flicker/

See also the Tutorial section of the present document for information on IE caching problems in relation with the Apache mod_gzip module.

Default Configurations

htmlArea RTE offers four default configurations:

- Typical: sets Page TSConfig and User TSConfig with most commonly used features representing a good start for typical sites and for most situations;
- Minimal: sets Page TSConfig and User TSConfig with minimal features; this is for advanced administrators who are familiar with the features of the RTE and understand well how to configure the RTE using Page TSConfig;
- Demo: sets Page TSConfig and User TSConfig with as many features as possible for users who want to explore the features of the RTE; this is **not** recommended for a production environment;
- Image: adds support for images to the Typical default configuration.

The default configuration may be chosen in the installation dialog of the htmlArea RTE extension presented by the Extension Manager. If not otherwise specified, the Typical default configuration will be used.

All default configurations share the same default configuration of RTE content transformation.

Default configuration of RTE content transformation

This default configuration establishes default settings in Page TSConfig for RTE content transformation.

For documentation of RTE tranformations, see: http://typo3.org/documentation/document-library/core-documentation/doc_core_api/current/view/5/2/

For documentation of Page TSConfig configuration of RTE processing, see: http://typo3.org/documentation/document-library/references/doc_core_tsconfig/current/view/1/3/

For documentation of the HTMLparser, see: http://typo3.org/documentation/document-library/references/doc_core_tsref/current/view/5/17/

For documentation of RTE settings in TCA, see: http://typo3.org/documentation/document-library/core-documentation/doc_core_api/current/view/4/3/

The following is inserted in Page TSConfig:

```
          ## Default RTE processing rules
RTE.default.proc {
              ## TRANSFORMATION METHOD

              ## We assume that CSS Styled Content is used.
      overruleMode = ts_css
              ## DO NOT CONVERT BR TAGS INTO LINEBREAKS

              ## br tags in the content are assumed to be intentional.
      dontConvBRtoParagraph = 1
              ## REMAP DIV TAGS TO P TAGS

              ## For compatibility between Classic RTE and htmlArea RTE
      remapParagraphTag = p
              ## TAGS ALLOWED OUTSIDE P & DIV
      allowTagsOutside = hr, address
              ## TAGS ALLOWED

              ## Added to the default internal list:
b,i,u,a,img,br,div,center,pre,font,hr,sub,sup,p,strong,em,li,ul,ol,blockquote,strike,span

              ## But, for the sake of clarity, we use a complete list in alphabetic order.

              ## center, font, strike, sdfield and  u will be removed on entry (see below).

              ## b and i will be remapped on exit (see below).

              ## Note that the link accessibility feature of htmlArea RTE does insert img tags.
      allowTags (
            a, abbr, acronym, address, blockquote, b, br, caption, cite, code, div, em,
            h1, h2, h3, h4, h5, h6, hr, i, img, li, link, ol, p, pre, q, sdfield,
            span, strong, sub, sup, table, tbody, td, th, tr, tt, ul
            )
              ## TAGS DENIED

              ## Make sure we can set rules on any tag listed in allowTags.
      denyTags >
```

```
        ## ALLOWED P & DIV ATTRIBUTES
        ## Attributes class and align are always preserved
        ## Align attribute will be unset on entry (see below)
        ## This is a list of additional attributes to keep
    keepPDIVattribs = xml:lang
        ## CONTENT TO DATABASE
    entryHTMLparser_db = 1
    entryHTMLparser_db {
                ## TAGS ALLOWED
                ## Always use the same list of allowed tags.
        allowTags < RTE.default.proc.allowTags
                ## TAGS DENIED
                ## Make sure we can set rules on any tag listed in allowTags.
        denyTags >
                ## AVOID CONTENT BEING HSC'ed TWICE
        htmlSpecialChars = 0

        tags {
                        ## REMOVE IMG TAGS
            img.allowedAttribs = 0
            img.rmTagIfNoAttrib = 1
                        ## CLEAN ATTRIBUTES ON THE FOLLOWING TAGS
            span.fixAttrib.style.unset = 1
            p.fixAttrib.align.unset = 1
            div.fixAttrib.align.unset = 1
            hr.allowedAttribs = class
            b.allowedAttribs = xml:lang
            blockquote.allowedAttribs = xml:lang
            cite.allowedAttribs = xml:lang
            em.allowedAttribs = xml:lang
            i.allowedAttribs = xml:lang
            q.allowedAttribs = xml:lang
            strong.allowedAttribs = xml:lang
            sub.allowedAttribs = xml:lang
            sup.allowedAttribs = xml:lang
            tt.allowedAttribs = xml:lang
        }
                ## REMOVE OPEN OFFICE META DATA TAGS AND DEPRECATED HTML TAGS
                ## We use this rule instead of the denyTags rule so that we can protect custom tags without protecting these unwanted tags.
        removeTags = center, font, o:p, sdfield, strike, u
                ## PROTECT CUSTOM TAGS
        keepNonMatchedTags = protect
    }

    HTMLparser_db {
                ## STRIP ALL ATTRIBUTES FROM THESE TAGS
                ## If this list of tags is not set, it will default to: b,i,u,br,center,hr,sub,sup,strong,em,li,ul,ol,blockquote,strike.
                ## However, we want to keep xml:lang attribute on most tags and tags from the default list where cleaned on entry.
        noAttrib = br
                ## XHTML COMPLIANCE
                ## Note that applying xhtml_cleaning on exit would break non-standard attributes of typolink tags
        xhtml_cleaning = 1
    }

    exitHTMLparser_db = 1
    exitHTMLparser_db {
```

## REMAP B AND I TAGS

## b and i tags are used by Mozilla/Firefox in editing mode.

## This must be done on exit because the default HTMLparser_db parsing executes the reverse mapping.

```
            tags.b.remap = strong
            tags.i.remap = em
```

## KEEP ALL TAGS

## Unwanted tags were removed on entry.

## Without this rule, the parser will remove all tags! Presumably, this rule will be more efficient than repeating the allowTags rule

```
            keepNonMatchedTags = 1
```

## AVOID CONTENT BEING HSC'ed TWICE

```
            htmlSpecialChars = 0
      }
}
```

## Use same RTE processing rules in FE

```
RTE.default.FE.proc < RTE.default.proc
```

## RTE processing rules for bodytext column of tt_content table

## Erase settings from other extensions

```
RTE.config.tt_content.bodytext >
```

## Make sure we use ts_css transformation

```
RTE.config.tt_content.bodytext.proc.overruleMode = ts_css
RTE.config.tt_content.bodytext.types.text.proc.overruleMode = ts_css
RTE.config.tt_content.bodytext.types.textpic.proc.overruleMode = ts_css
```

Typical default configuration

This default configuration establishes default settings in Page TSConfig and User TSConfig with most commonly used features representing a good start for typical sites.

The following is inserted in Page TSConfig:

## Define labels and styles to be applied to class selectors in the interface of the RTE

## The examples included here make partial re-use of color scheme and frame scheme from CSS Styled Content extension

```
RTE.classes {
      align-left {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifyleft
            value = text-align: left;
      }
      align-center {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifycenter
            value = text-align: center;
      }
      align-right {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifyright
            value = text-align: right;
      }
      csc-frame-frame1 {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:frame-frame1
            value = background-color: #EDEBF1; border: 1px solid #333333;
      }
      csc-frame-frame2 {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:frame-frame2
            value = background-color: #F5FFAA; border: 1px solid #333333;
      }
      important {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:important
            value = color: #8A0020;
      }
      name-of-person {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:name-of-person
            value = color: #10007B;
      }
      detail {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:detail
```

```
              value = color: #186900;
        }
        component-items {
              name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:component-items
              value = color: #186900;
        }
        action-items {
              name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:action-items
              value = color: #8A0020;
        }
        component-items-ordered {
              name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:component-items
              value = color: #186900;
        }
        action-items-ordered {
              name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:action-items
              value = color: #8A0020;
        }
}
```

## Anchor classes configuration for use by the anchor accesibility feature (htmlArea RTE only)

```
RTE.classesAnchor {
      externalLink {
            class = external-link
            type = url
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_titleText
      }
      externalLinkInNewWindow {
            class = external-link-new-window
            type = url
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_new_window_titleText
      }
      internalLink {
            class = internal-link
            type = page
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_titleText
      }
      internalLinkInNewWindow {
            class = internal-link-new-window
            type = page
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_new_window_titleText
      }
      download {
            class = download
            type = file
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:download_titleText
      }
      mail {
            class = mail
            type = mail
            titleText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:mail_titleText
      }
}
```

## Default RTE configuration

```
RTE.default {
```

## Markup options (htmlArea RTE only)

```
      enableWordClean = 1
      removeTrailingBR = 1
      removeComments = 1
      removeTags = center, font, o:p, sdfield, strike,u
      removeTagsAndContents = style,script
```

## Toolbar options

## The TCA configuration may add buttons to the toolbar

## The following buttons are specific to Classic RTE: class

## The following buttons are specific to htmlArea RTE: blockstylelabel, blockstyle, textstylelabel, textstyle,

```
        ##                      insertcharacter, findreplace, removeformat, toggleborders, tableproperties,
        ##                      rowproperties, rowinsertabove, rowinsertunder, rowdelete, rowsplit,
        ##                      columninsertbefore, columninsertafter, columndelete, columnsplit,
        ##                      cellproperties, cellinsertbefore, cellinsertafter, celldelete, cellsplit, cellmerge
    showButtons (
        class, blockstylelabel, blockstyle, textstylelabel, textstyle,
        formatblock, bold, italic, subscript, superscript,
        orderedlist, unorderedlist, outdent, indent, textindicator,
        insertcharacter, link, table, findreplace, chMode, removeformat, undo, redo, about,
        toggleborders, tableproperties,
        rowproperties, rowinsertabove, rowinsertunder, rowdelete, rowsplit,
        columninsertbefore, columninsertafter, columndelete, columnsplit,
        cellproperties, cellinsertbefore, cellinsertafter, celldelete, cellsplit, cellmerge
    )
        ## More toolbar options (htmlArea RTE only)
    keepButtonGroupTogether = 1

        ## Enable status bar (htmlArea RTE only)
    showStatusBar =  1

        ## Hide infrequently used paragraph types in the paragraph type selector (formatblock button)
    hidePStyleItems = pre,address

        ## Add default example styles
        ## The example styles included here make partial re-use of color scheme and frame scheme from CSS Styled Content extension
        ## Left, center and right alignment of text in paragraphs and cells.
    inlineStyle.text-alignment (
        p.align-left, h1.align-left, h2.align-left, h3.align-left, h4.align-left, h5.align-left, h6.align-left, td.align-left { text-align: left; }
        p.align-center, h1.align-center, h2.align-center, h3.align-center, h4.align-center, h5.align-center, h6.align-center, td.align-center { text-align: center; }
        p.align-right, h1.align-right, h2.align-right, h3.align-right, h4.align-right, h5.align-right, h6.align-right, td.align-right { text-align: right; }
    )
        ## Two frame examples taken from the example CSS file of CSS Styled Content extension and applied to p and table block elements.
    inlineStyle.frames (
        p.csc-frame-frame1, table.csc-frame-frame1 { background-color: #EDEBF1; padding: 2px 4px 2px 4px; border: 1px solid #333333; }
        p.csc-frame-frame2, table.csc-frame-frame2 { background-color: #F5FFAA; padding: 2px 4px 2px 4px; border: 1px solid #333333; }
    )
        ## Bullet styles for unordered lists.
    inlineStyle.ul (
        ul.component-items { color: #186900; list-style-type: circle; }
        ul.action-items { color: #8A0020; list-style-image: url(img/red_arrow_bullet.gif); }
    )
        ## Numbering styles for ordered lists.
    inlineStyle.ol (
        ol.component-items-ordered { color: #10007B; list-style-type: lower-roman; }
        ol.action-items-ordered { color: #8A0020; list-style-type: lower-greek; }
    )
        ## Three inline text colors taken from the color scheme of CSS Styled Content extension.
    inlineStyle.inline-text (
        span.important { color: #8A0020; }
        span.name-of-person { color: #10007B; }
        span.detail { color: #186900; }
    )
        ## Use stylesheet file rather than the above mainStyleOverride and inlineStyle properties to style the contents (htmlArea RTE only)
    ignoreMainStyleOverride = 1

        ## List all class selectors that are allowed on the way to the database
    proc.allowedClasses (
        external-link, external-link-new-window, internal-link, internal-link-new-window, download,
mail,
            align-left, align-center, align-right,
            csc-frame-frame1, csc-frame-frame2,
            component-items, action-items,
            component-items-ordered, action-items-ordered,
            important, name-of-person, detail
      )
             ## classesParagraph, classesTable, classesTD, classesLinks, classesCharacter

             ## Classic RTE: Specify the list of class selectors that should be presented in the RTE interface:

             ## htmlArea RTE: Restrict the list of class selectors presented by the RTE to the following:

      classesParagraph (
            align-left, align-center, align-right,
            csc-frame-frame1, csc-frame-frame2
      )
      classesTable = csc-frame-frame1, csc-frame-frame2
      classesTD = align-left, align-center, align-right
      classesLinks = external-link, external-link-new-window, internal-link, internal-link-new-window,
download, mail
      classesCharacter = important, name-of-person, detail

             ## Configuration of the anchor accessibility feature (htmlArea RTE only)

             ## These classes should also be in the list of allowedClasses.

      classesAnchor = external-link, external-link-new-window, internal-link, internal-link-new-window,
download, mail
      classesAnchor.default {
            page = internal-link
            url = external-link-new-window
            file = download
            mail = mail
      }
             ## Configuration specific to the TableOperations feature (htmlArea RTE only)

             ## Remove the following fieldsets from the properties popups

      disableAlignmentFieldsetInTableOperations = 1
      disableSpacingFieldsetInTableOperations = 1
      disableColorFieldsetInTableOperations = 1
      disableLayoutFieldsetInTableOperations = 1
}
         ## front end RTE configuration for the general public (htmlArea RTE only)

RTE.default.FE < RTE.default
RTE.default.FE.showStatusBar = 0
RTE.default.FE.hideButtons = chMode, blockstyle, textstyle, underline, strikethrough, subscript,
superscript, lefttoright, righttoleft, left, center, right, justifyfull, table, inserttag, findreplace,
removeformat, copy, cut, paste
RTE.default.FE.userElements >
RTE.default.FE.userLinks >

         ## tt_content TCEFORM configuration

         ## Let use all the space available for more comfort.

TCEFORM.tt_content.bodytext.RTEfullScreenWidth= 100%
```

The following is inserted in User TSConfig:

```
         ## Enable the RTE by default for all users

setup.default.edit_RTE = 1
```

Minimal default configuration

This default configuration establishes default settings in Page TSConfig and User TSConfig with minimal features. This is for advanced administrators with good knowledge of RTE configuration in Page TSConfig.

The following is inserted in Page TSConfig:

```
         ## Default RTE configuration

RTE.default {
             ## Toolbar options applicable to all RTE's

             ## The TCA configuration may add buttons to the toolbar

      showButtons =  bold,italic,undo,redo,about
}
```

## front end RTE configuration for the general public (htmlArea RTE only)

```
RTE.default.FE < RTE.default
```

The following is inserted in User TSConfig:

## Enable the RTE by default for all users

```
setup.default.edit_RTE = 1
```

Demo default configuration

This default configuration sets Page TSConfig and User TSConfig with as many features as possible for users who want to explore the features of the RTE. This is **not** recommended for production environments.

The following is inserted in Page TSConfig:

## Define labels and styles to be applied to class selectors in the interface of the RTE

## Partial re-use of color scheme and frame scheme from CSS Styled Content extension

```
RTE.classes {
      align-left {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifyleft
            value = text-align: left;
      }
      align-center {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifycenter
            value = text-align: center;
      }
      align-right {
            name = LLL:EXT:rtehtmlarea/htmlarea/locallang_tooltips.xml:justifyright
            value = text-align: right;
      }
      csc-frame-frame1 {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:frame-frame1
            value = background-color: #EDEBF1; border: 1px solid #333333;
      }
      csc-frame-frame2 {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:frame-frame2
            value = background-color: #F5FFAA; border: 1px solid #333333;
      }
      important {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:important
            value = color: #8A0020;
      }
      name-of-person {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:name-of-person
            value = color: #10007B;
      }
      detail {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:detail
            value = color: #186900;
      }
      component-items {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:component-items
            value = color: #186900;
      }
      action-items {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:action-items
            value = color: #8A0020;
      }
      component-items-ordered {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:component-items
            value = color: #186900;
      }
      action-items-ordered {
            name = LLL:EXT:rtehtmlarea/htmlarea/plugins/DynamicCSS/locallang.xml:action-items
            value = color: #8A0020;
      }
}
```

## Anchor classes configuration for use by the anchor accessibility feature (htmlArea RTE only)

```
RTE.classesAnchor {
      externalLink {
            class = external-link
            type = url
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/external_link.gif
            altText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_altText
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_titleText
      }
      externalLinkInNewWindow {
            class = external-link-new-window
            type = url
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/external_link_new_window.gif
            altText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_new_window_altText
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_new_window_titleText
      }
      internalLink {
            class = internal-link
            type = page
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/internal_link.gif
            altText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_altText
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_titleText
      }
      internalLinkInNewWindow {
            class = internal-link-new-window
            type = page
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/internal_link_new_window.gif
            altText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_new_window_altText
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_new_window_titleText
      }
      download {
            class = download
            type = file
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/download.gif
            altText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:download_altText
            titleText =
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:download_titleText
      }
      mail {
            class = mail
            type = mail
            image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/mail.gif
            altText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:mail_altText
            titleText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:mail_titleText
      }
}
```

## Default RTE configuration

```
RTE.default {
```

## Markup options (htmlArea RTE only)

```
      enableWordClean = 1
      removeTrailingBR = 1
      removeComments = 1
      removeTags = center, o:p, sdfield
      removeTagsAndContents = style,script
```

## Allow img tags

```
      proc.entryHTMLparser_db.tags.img >
```

## Allow align and style attributes on p and span tags

```
      proc.entryHTMLparser_db.tags {
            p.fixAttrib.align.unset >
            div.fixAttrib.align.unset >
            p.allowedAttribs = class,style,align
            span.fixAttrib.style.unset >
      }
```

## Toolbar options

```
      showButtons = *
```

```
        ## More toolbar options (htmlArea RTE only)
keepButtonGroupTogether = 1

        ## Enable status bar (htmlArea RTE only)
showStatusBar =  1

        ## Add default example styles
        ## Left, center and right alignment of text in paragraphs and cells.
inlineStyle.text-alignment (
      p.align-left, h1.align-left, h2.align-left, h3.align-left, h4.align-left, h5.align-left, h6.align-left, td.align-left { text-align: left; }
      p.align-center, h1.align-center, h2.align-center, h3.align-center, h4.align-center, h5.align-center, h6.align-center, td.align-center { text-align: center; }
      p.align-right, h1.align-right, h2.align-right, h3.align-right, h4.align-right, h5.align-right, h6.align-right, td.align-right { text-align: right; }
)
        ## Two frame examples taken from the example CSS file of CSS Styled Content extension and applied to p and table block elements.
inlineStyle.frames (
      p.csc-frame-frame1, table.csc-frame-frame1 { background-color: #EDEBF1; padding: 2px 4px 2px 4px; border: 1px solid #333333; }
      p.csc-frame-frame2, table.csc-frame-frame2 { background-color: #F5FFAA; padding: 2px 4px 2px 4px; border: 1px solid #333333; }
)
        ## Bullet styles for unordered lists.
inlineStyle.ul (
      ul.component-items { color: #186900; list-style-type: circle; }
      ul.action-items { color: #8A0020; list-style-image: url(img/red_arrow_bullet.gif); }
)
        ## Numbering styles for ordered lists.
inlineStyle.ol (
      ol.component-items-ordered { color: #10007B; list-style-type: lower-roman; }
      ol.action-items-ordered { color: #8A0020; list-style-type: lower-greek; }
)
        ## Three inline text colors taken from the color scheme of CSS Styled Content extension.
inlineStyle.inline-text (
      span.important { color: #8A0020; }
      span.name-of-person { color: #10007B; }
      span.detail { color: #186900; }
)

        ## Use stylesheet file rather than the above mainStyleOverride and inlineStyle properties to style the contents (htmlArea RTE only)
ignoreMainStyleOverride = 1

        ## For this demo, do not remove font, strike and u tags
proc.entryHTMLparser_db.removeTags := removeFromList(font,strike,u)

        ## List all class selectors that are allowed on the way to the database
proc.allowedClasses (
      external-link, external-link-new-window, internal-link, internal-link-new-window, download, mail,
      align-left, align-center, align-right,
      csc-frame-frame1, csc-frame-frame2,
      component-items, action-items,
      component-items-ordered, action-items-ordered,
      important, name-of-person, detail
)

        ## classesParagraph, classesTable, classesTD, classesLinks, classesCharacter
        ## Classic RTE: Specify the list of class selectors that should be presented in the RTE interface:
        ## htmlArea RTE: Restrict the list of class selectors presented by the RTE to the following:
classesParagraph (
      align-left, align-center, align-right,
      csc-frame-frame1, csc-frame-frame2
)
classesTable = csc-frame-frame1, csc-frame-frame2
classesTD = align-left, align-center, align-right
```

```
	classesLinks = external-link, external-link-new-window, internal-link, internal-link-new-window, 
download, mail
	classesCharacter = important, name-of-person, detail

			## Configuration of the anchor accessibility feature (htmlArea RTE only)
			## These classes should also be in the list of allowedClasses.
	classesAnchor = external-link, external-link-new-window, internal-link, internal-link-new-window, 
download, mail
	classesAnchor.default {
		page = internal-link
		url = external-link-new-window
		file = download
		mail = mail
	}

			## Show all applicable class selectors available in the style sheet file (htmlArea RTE only)
	showTagFreeClasses = 1

			## Configuration specific to the table button or TableOperations feature (htmlArea RTE only)
			## Use the context menu instead of the toolbar for table operations, but keep toggleborders button in toolbar
	hideTableOperationsInToolbar = 1
	keepToggleBordersInToolbar = 1

			## Configuration specific to the inserttag button or QuickTag feature (htmlArea RTE only)
			## Do not allow insertion of the following tags
	hideTags = font, font (full), underline, strike, table

			## Configuration specific to the spellcheck button or SpellCheck feature (htmlArea RTE only)
			## Enable the use of personal dictionaries
	enablePersonalDicts = 1
}

		## Use same processing as on entry to database to clean content pasted into the editor
RTE.default.enableWordClean.HTMLparser < RTE.default.proc.entryHTMLparser_db

		## front end RTE configuration (htmlArea RTE only)
RTE.default.FE < RTE.default
RTE.default.FE.userElements >
RTE.default.FE.userLinks >

		## tt_content TCEFORM configuration
		## Let use all the space available for more comfort.
TCEFORM.tt_content.bodytext.RTEfullScreenWidth= 100%
```

The following is inserted in User TSConfig:

```
		## Enable the RTE by default for all users
setup.default.edit_RTE = 1

		## Enable the file upload feature of the element browser by default for all users
options.uploadFieldsInTopOfEB = 1

		## Set the default spelling ability of the check speller for all users
options.HTMLAreaPspellMode = bad-spellers

		## Enable the personal dictionary feature of the check speller by default for all users
options.enablePersonalDicts = 1
```

Image default configuration

This default configuration establishes support for images in Page TSConfig. This is for use in addition to the Typical default configuration.

The following is inserted in Page TSConfig:

## Anchor classes configuration for use by the anchor accesibility feature (htmlArea RTE only)

## Add inline icons to the links

```
RTE.classesAnchor {
	externalLink {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/external_link.gif
		altText = 
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_altText
	}
	externalLinkInNewWindow {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/external_link_new_window.gif
		altText = 
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:external_link_new_window_altText
	}
	internalLink {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/internal_link.gif
		altText = 
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_altText
	}
	internalLinkInNewWindow {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/internal_link_new_window.gif
		altText = 
LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:internal_link_new_window_altText
	}
	download {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/download.gif
		altText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:download_altText
	}
	mail {
		image = EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/img/mail.gif
		altText = LLL:EXT:rtehtmlarea/htmlarea/plugins/TYPO3Browsers/locallang.xml:mail_altText
	}
}
```

## Default RTE configuration

```
RTE.default {

		## Enable the image button
	showButtons := addToList(image)

		## Tags allowed outside p and div
		## Adding img tag to the default list
	proc.allowTagsOutside := addToList(img)

		## Do not remove img tags
	proc.entryHTMLparser_db.tags.img >
}

RTE.default.FE.showButtons < RTE.default.showButtons
RTE.default.FE.proc.allowTagsOutside < RTE.default.proc.allowTagsOutside
RTE.default.FE.proc.entryHTMLparser_db.tags.img >
```

Tutorial

Working around Internet Explorer caching problem

This article by Jan Wulff describes the problems encountered with Internet Explorer 4/5/6 and the Apache server with mod_gzip activated. It describes how to work around these problems.

Problem:

If the Apache module mod_gzip is activated on your server, you may encounter the problem that Internet Explorer denies any caching for the whole site, thus stopping the DynamicCSS and InlineCSS plugins of htmlArea RTE to work correctly. Besides, it may even slow down some other features of TYPO3, like graphical JavaScript menus. This effect does not depend on the gzip compression itself. Internet Explorer is indeed able to handle compressed files. The problem is IE's handling of one of the HTTP response headers sent with every served document.

Background:

The HTTP Vary response header indicates whether a cache is permitted to use the response to reply to a subsequent request without re-validating the document. This is necessary if a document is not suitable for all clients and is served in multiple different versions according to the HTTP headers the client sends with his request.

For example, with activated mod_gzip, every document is at least available in two versions, compressed and uncompressed. If a browser with gzip support requests such a document, it will receive the compressed version. A proxy between the client and the server may cache this file. Now, another browser without gzip support requests the same document via the same proxy. Without the Vary header the proxy would not know if the compressed document may be delivered to the new client, because it can't compare the HTTP headers of the second browser with the Vary header. If it would nevertheless serve it, the client would receive a bunch of data, without any idea, how to process it. Therefore mod_gzip sends a Vary header with each response with at least 'Accept-Encoding' as content.

The problems arise when the Internet Explorer enters the stage. IE 4, 5 and 6 recognizes only one kind of Vary header: 'User-agent', used to distinguish between versions for different browsers. Every other Vary header will be interpreted as it would have a single '*' as content. Because this does not compare with the headers send by any client, it forbids any caching of documents received with this header.

Solution:

There is more than one approach to handle this problem. The following configuration directives all have to be set in the Apache configuration file or in a .htaccess file which has to be located in your TYPO3 root.

Easy going:

So, you have no need for any gzip support? Fine, just deactivate the module and your problems are gone. Use this directive:

```
mod_gzip_on No
```

Complex approach:

You do have a lot of big code or text files, or you have to save as much transfer bandwidth as possible? Anyway, deactivating mod_gzip is no option for you? Then, you should first check what release of mod_gzip your server is using. If you don't know how, ask your provider, or just use the solution for releases from 1.3.19.2a till 1.3.26.1a.

- mod_gzip release < 1.3.19.2a

Releases before this version didn't send Vary headers, so there shouldn't be any problem. But because you're reading this, you most probably don't use these versions.

- mod_gzip release 1.3.19.2a <> 1.3.26.1a

These releases all use Vary headers. However, they send these headers without verifying if the document is really checked for compression. The only recommended way to get around this, is to deactivate mod_gzip. But thanks to Apache, you can deactivate mod_gzip separately for chosen files, and let it do it's work for the rest. You could use this to deactivate mod_gzip for all css files:

```
<FilesMatch "\.css$">
mod_gzip_on No
</FilesMatch>
```

Or going even further, you could also include image files:

```
<FilesMatch "\.(css|gif|jpe?g|png)$">
mod_gzip_on No
</FilesMatch>
```

By the way, there is another possibility. You could deactivate Vary headers in mod_gzip with this:

```
mod_gzip_send_vary Off
```

But there is a reason why mod_gzip, since 1.3.19.2a, uses Vary headers. As described above, you could badly mess up proxy servers, by serving compressed files without Vary headers. Therefore, I strongly discourage this approach.

- mod_gzip release > 1.3.26.1a

Since release 1.3.26.1a, mod_gzip is a bit more discriminate. It only sends Vary headers with documents which were checked for compression. So you can tell mod_gzip to exclude some files. This approach is not so much different from the former solution, but it is cleaner because it addresses mod_gzip firsthand. To exclude CSS files from compression, use this directive:

```
mod_gzip_item_exclude file \.css$
```

If you would like to add images and Javascript files, you could use this:

```
mod_gzip_item_exclude file \.css$
mod_gzip_item_exclude file \.png$
mod_gzip_item_exclude file \.gif$
mod_gzip_item_exclude file \.jpg$
mod_gzip_item_exclude file \.jpeg$
mod_gzip_item_exclude file \.js$
```

This is just a short survey of the caching problems with Internet Explorer and mod_gzip. I wrote it with best intent and hope it may be helpful. If you find any mistakes, please let me know at <messages@janwulff.de>. I'm in no way responsible for any consequences that may come forth by the use of this information.

Using htmlArea RTE in a frontend plugin

Extension developers may use the htmlArea RTE to edit textarea fields in frontend plugins. In order to do so, you must invoke the htmlArea plugin in the plugin class and insert some markers in the HTML template.

Invoking the htmlArea RTE frontend plugin

Step 1. Include the htmlArea RTE frontend plugin:

```
      require_once(t3lib_extMgm::extPath('rtehtmlarea').'pi2/class.tx_rtehtmlarea_pi2.php');
```

Step 2. Declare the following class variables for the RTE API:

```
      var $RTEObj;
      var $docLarge = 0;
      var $RTEcounter = 0;
      var $formName;
      var $additionalJS_initial = '';               // Initial JavaScript to be printed before the form
(should be in head, but cannot due to IE6 timing bug)
      var $additionalJS_pre = array(); // Additional JavaScript to be printed before the form (works in
Mozilla/Firefox when included in head, but not in IE6)
      var $additionalJS_post = array();             // Additional JavaScript to be printed after the
form
      var $additionalJS_submit = array();           // Additional JavaScript to be executed on submit
      var $PA = array(
            'itemFormElName' =>  '',
            'itemFormElValue' => '',
            );
      var $specConf = array();
      var $thisConfig = array();
      var $RTEtypeVal = 'text';
      var $thePidValue;
```

Step 3. Set the HTML markers for rendering the form:

```
      $this->postvars = t3lib_div::_POST('tx_myextension_pi1');
      if(!$this->RTEObj)  $this->RTEObj = t3lib_div::makeInstance('tx_rtehtmlarea_pi2');
      if($this->RTEObj->isAvailable()) {
            $this->RTEcounter++;
            $this->table = 'my-table-name';
            $this->field = 'my-field-name';
            $this->formName = 'my-form-name';
            $this->PA['itemFormElName'] = 'tx_myextension_pi1[my-field-name]';
            $this->PA['itemFormElValue'] = $this->postvars['my-field-name'];
            $this->thePidValue = $GLOBALS['TSFE']->id;
            $RTEItem = $this->RTEObj->drawRTE($this, 'my-table-name', 'my-field-name', $row=array(),
$this->PA, $this->specConf, $this->thisConfig, $this->RTEtypeVal, '', $this->thePidValue);
            $markerArray['###ADDITIONALJS_PRE###'] = $this->additionalJS_initial.'
            <script type="text/javascript">'. implode(chr(10), $this->additionalJS_pre).'
            </script>';
            $markerArray['###ADDITIONALJS_POST###'] = '
            <script type="text/javascript">'. implode(chr(10), $this->additionalJS_post).'
            </script>';
            $markerArray['###ADDITIONALJS_SUBMIT###'] = implode(';', $this->additionalJS_submit);
            $markerArray['###FORM_RTE_ENTRY###'] = $RTEItem;
      }
```

Step 4. Process and save data sent in the textarea:

```
      if($this->RTEObj->isAvailable()) {
            $pageTSConfig = $GLOBALS['TSFE']->getPagesTSconfig();
            $RTEsetup = $pageTSConfig['RTE.'];
            $this->thisConfig = $RTEsetup['default.'];
            $this->thisConfig = $this->thisConfig['FE.'];
            $dataArray['my-field-name'] = $this->RTEObj->transformContent('db',$dataArray['my-field-
name'], 'my-table-name', 'my-field-name', $dataArray, $this->specConf, $this->thisConfig, '', $this-
>thePidValue);
      }
```

```
        $insert = $GLOBALS['TYPO3_DB']->exec_INSERTquery('my-table-name', $dataArray);
```

where $dataArray is the array of data used to insert the data into the table.

Inserting markers in the HTML template

Step 1. Insert JavaScript marker before the form:

```
###ADDITIONALJS_PRE###
```

Step 2. Insert onSubmit marker on the form tag:

```
<form name="my-form-name" method="post" action="###ACTION_URL###" onsubmit="###ADDITIONALJS_SUBMIT###" >
```

Step3. Replace the <textarea> tag and its contents with:

```
###FORM_RTE_ENTRY###
```

Step4. Insert JavaScript marker after the form:

```
###ADDITIONALJS_POST###
```

Known problems

Please see/report problems in the TYPO3 Bugtracker under project tx_rtehtmlarea.

You may get support in the use of this extension by subscribing to news://news.netfielders.de/typo3.projects.rte.

Internet Explorer and HTML tag abbr

Internet Explorer does not support HTML tag abbr. If a content element containing this tag is edited with htmlArea RTE in IE, the tag may be broken. Therefore, abbr tags are transformed to acronym tags if IE is used.

Internet Explorer and mod_gzip

If the Apache module mod_gzip is activated, it adds a 'vary'-header to every document. Internet Explorer cannot process correctly such headers, causing incorrect behavior of htmlArea RTE and, in particular, of the DynamicCSS and InlineCSS plugins. Therefore, mod_gzip compression, at least of style sheet files, should be disabled. See the article written by Jan Wulff on this subject in the Tutorial section of this document.

Mozilla 1.3.1 and TCA feature dividers2Tabs

When the TCA feature **dividers2Tabs** is enabled on some table of the TYPO3 back end, the text field to be edited with the RTE may be hidden in a tab layer. This generally causes problem to Mozilla/Firefox browsers. The htmlArea RTE has been modified to support this feature in Mozilla 1.4+ and Firefox 0.8+. The modification does not work in Mozilla 1.3.1.

“All-in-One Gestures” Firefox extension and the context menu

The “All-in-One Gestures” Firefox extension causes erratic behavior of the htmlArea RTE context menu.

To-Do list

Please see/submit feature requests in the TYPO3 Bugtracker under project tx_rtehtmlarea.

You may sponsor further development of this extension by sending your contribution to stanislas.rolland@fructifor.ca through moneybookers.com or PayPal, or by contacting the author.

Change log

Version 0.5.0:

- Use doubleval in place of floatval in SpellChecker plugin.
- Use XMLHttpRequest to asynchronously download required scripts.
- Add troubleshooting log and extension configuration variable to enable logging.
- Multiple changes to support Safari 1.3. Although the editor loads in Safari 1.3, it does not currently execute correctly due to some bugs in Safari edit mode. If you want to do some testing and/or debugging, you need to set $GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['rtehtmlarea']['safari_test']=1;
- Restructure the tool bar as an unordered list of label, select and button elements so that it can be walked through with the tab key. The list, including the button icons, is css-styled through the editor skin. Beware that custom skins will need to be adjusted.

- Restructure the context menu as an unordered list of button elements. The context menu, including the button icons, is css-styled through the editor skin. Beware that custom skins will need to be adjusted.
- Show in the context menu all table operations enabled in the given context.
- Each of the table operations becomes a configurable tool bar element like all other tool bar elements.
- New RTE Page TSConfig property: hideTableOperationsInToolbar. If set, the table operations buttons will not be displayed in the tool bar, but will appear in the context menu, when enabled in the given context.
- Method TS_transform_db of class t3lib_parsehtml_proc is modified in order to replace true line breaks with spaces, instead of eliminating them. Thanks to Johannes Bornhold. See section Known problems.
- New htmlArea plugin UserElements to enable the “user” button.
- Add the title attribute on the TYPO3 link insertion/modification dialog.
- New toggleborders button to show table borders.

Version 0.5.1:

- Correction to the behavior of the style selection lists.

Version 0.5.2:

- Correction for IE: when the collapsed borders attribute was set on a table, the table properties popup dialog did not reflect the set attribute: the checkbox was not checked.
- Correction: when TYPO3 is not installed in the web server's root path, the dynamiccss.css file was not properly included. Thanks to Johannes Bornhold for this correction.
- Correction for IE: on some Windows platforms, the display of the window status bar cannot be avoided unless the script is granted some privileges; in such cases, the height of the resized popup windows was insufficient; the resizing algorithm has been revised in order to correct this situation.
- Correction: the mailto link function did not work.
- Allow insertion of a file uploader on the File tab of the Insert/Modify link popup window (when TYPO3Browsers is used); the feature is enabled when property options.uploadFieldsInTopOfEB is set to 1 in User TSConfig. Thanks to Peter Niederlag for adding this feature.
- For TYPO3 3.7.0+: use the tab menu in TYPO3 Insert/Modify link popup window.
- Modify default Page TSConfig RTE configuration so as to remove Open Office <sdfield>-tags.
- New Page TSConfig properties: removeComments, removeTags, removeTagsAndContents.
- Correction for Mozilla/Firefox: when pasting in text by pressing the ctrl-v keys, the Mozilla extension installation was incorrectly triggered and/or failed.
- Correction: inserting a paragraph from the context menu when positioned in a table or a list produced incorrect result.
- New plugin/button allows to insert/edit acronyms and abbreviations. Not currently available in IE. Acronyms and abbreviations may be pre-defined with Acronym record type inserted on the TYPO3 site root.
- Transform abbr tags into acronym tags when IE is used.

Version 0.5.3:

- Correction to misbehavior of mailto link popup when no anchor class is specified in Page TSConfig.

Version 0.5.4:

- Correction: one label of plugin RemoveFormat was not localized.
- Correction: custom skins prevented correct loading of the RTE.
- Correction for IE: incorrect display of the current class in the paragraph style selector in some configurations.
- Correction for IE: the border handles of images and tables could not be selected for resizing.
- Correction for IE: table, img, ol and ul elements may be correctly selected through the editor status bar.
- Correction for IE: incorrect positioning of some popup windows.
- Correction for IE: incorrect behavior of the InlineCSS text style selector when Arabic text was selected.

Version 0.5.5:

- The size of the back end editor area may be configured in User TSConfig. See the configuration section. Thanks to Peter Enzenberger.

- In Mozilla/Firefox, url's and email's are automatically transformed into links as soon as they are followed by a space; hitting escape or ctrl-z before any other key unlinks the text. Thanks to Xinha.
- Scripts compressed by about 25%. Use of compressed scripts is controlled by extension configuration.
- Improved memory management and reduced memory leakage.
- Correction: the cellinsertafter button was incorrectly configured in the default configuration.
- Correction: when multiple instances of the editor were loaded on the same page, the always-show buttons were added only on the tool bar of the first instance.
- Correction: the path to default skin component htmlarea-edited-content.css was incorrectly set.
- When the configuration variable **Enable Mozilla/Firefox extension** is enabled, if the client browser is Firefox 1+, the installation of the Firefox extension AllowClipboard Helper will triggered instead of the extension generated by this TYPO3 extension.
- In Mozilla/Firefox, an exception will be raised only when performing a paste with the tool bar button. Pasting with ctrl-v key will not raise any exception. Thanks to Guillaume Crico.

Version 0.5.6:

- Correction: the MS Word Formatting option of the RemoveFormat button was broken in the previous version.
- Url of AllowClipboard Helper is added as an extension configuration variable.
- The address and dl tags are added to the list $headListTags of TYPO3 system class t3lib_parsehtml_proc. See Known problems.

Version 0.5.7:

- Correction: since version 0.5.5, html attribute values could be html-encoded more than once.

Version 0.5.8:

- Correction: the editor will load correctly even if it is located in a hidden tab at te time of loading. See Known problems.
- Improved memory management for IE.

Version 0.5.9:

- Correction for IE: with some locales, the type of paragraph displayed in the toolbar may not correspond to the type of paragraph of the current selection.
- Address is added to the default list of types of paragraphs.
- Correction: since 0.5.5, html-encoding of single quotes was causing incompatibilities.
- Correction: the Acronym plugin was broken since version 0.5.5.
- Correction: if a title is added on a link, the link is not eligible as typolink, and TYPO3 will add a rtekeep attribute on the a tag; if the link is edited and the title removed, the editor should remove the rtekeep attribute so that the link becomes eligible as typolink again.
- Correction: since 0.5.2, the Insert/Modify Link dialog could be broken in TYPO3 3.6.
- New Page TSConfig property: removeTrailingBR. In Mozilla/Firefox/Netscape browsers, whenever some text is entered in an empty block, a trailing br tag is added by the browser. If this property is set, any trailing br tag in a block element will be removed on save and/or change mode. However, multiple trailing br tags will be preserved.
- For IE: an error message will be displayed if execution of ActiveX controls is not enabled.

Version 0.6.1:

- Added documentation section on server configuration.
- Correction: the compressed version of plugin scripts were not used.
- Various changes for performance improvement.
- Added FAQ on hotkeys.
- Update to Finnish and Norwegian translations.

Version 0.6.2:

- Correction: the image update popup dialog was broken in the previous version.
- Update to the recommended server configuration for png images.

- Addition of two example skins: blue-look and xp-blue, inspired by Xinha.
- New Page TSConfig property: keepButtonGroupTogether will keep buttons of a same group together on the same line of the editor tool bar. Not supported in IE, nor in Mozilla 1.3.
- Adjustments to the default skin. Custom skins may need to be adjusted.

Version 0.6.3:

- Correction: the editor failed to initialize correctly when Page TSConfig property hideTableOperationsInToolbar was set to 1.
- Addition of Toggle borders in the context menu.
- Correction: updating a link did not work when the link was set on an image and on multiple elements.

Version 0.7.0:

- Problem with Firefox extension “All-in-One Gestures” documented as Known problem.
- Blue-look and xp-blue skins moved to separate extensions: sr_rtehtmlarea_bluelook and sr_rtehtmlarea_xpblue respectively.
- Update to Finnish and Norwegian translations.

Version 0.7.2:

- Correction: incorrect display of the toolbar in some configurations.
- Correction: incorrect update of some table properties.
- Adjustments to the names of the css selectors of some toolbar buttons. Skins sr_rtehtmlarea_bluelook and sr_rtehtmlarea_xpblue need to be upgraded. Custom skins need to be updated.
- Correction: incorrect behavior of the hot keys (ctrl+key) and of the context menu's access keys (alt+key).
- Remove generation of Mozilla/Firefox extension. Remove requirement for PHP zlib. Remove Archive_Zip package. Trigger install of AllowClipboard Helper for all supported versions of Mozilla/Firefox. AllowClipboardHelper Firefox extension has been localized to French and Spanish.
- Correction: some labels of the non-TYPO3 link and image insertion popups were not localized.
- Revised About window. Skins need to be upgraded. Custom skins need to be updated.
- New anchor accessibility feature.
- Correction: erratic behaviors in Mozilla/Firefox when automatically detecting links and when backspacing.
- Correction: PHP Fatal error when SpellChecker is enabled under PHP5. See http://bugs.typo3.org/view.php?id=1470
- Update to the Dutch and German translations.
- Correction for IE (and maybe other cases, see http://bugs.typo3.org/view.php?id=1425): When an unforgiving browser refuses to parse the content coming from the database, a popup window will inform the user that the HTML document is not well-formed, and the editor will be started in textmode so that the user can edit the error in the HTML source.
- Correction: nested lists were not compliant with XHTML.
- Correction: the path to skin file htmlarea-edited-content.css could be incorrect when typo3 is installed in a sub-directory.
- Correction for IE: the paragraph style selector box was disabled.

Version 0.7.4:

- Correction for Mozilla/Firefox: when the editor was loaded in a hidden tab, the size of the editing area would be incorrect when the tab becomes visible.
- Update to Finnish and Italian translations.
- Addition of Czech translation.
- Correction to the documentation of the anchor accessibility feature.
- Correction: insufficiently qualified CSS selectors in the editor skins resulted in styling problems mainly when the editor is used in the front end. Thanks to Claudio Strizzolo.
- Correction: toggleborders button caused inconsistencies in the class selector of the table properties popup window.
- Correction to the character set used by the xml parser and Aspell when invoked by the spell checker with

iso-8859-1 content.

Version 0.7.6:

- Two new extension configuration variables allow to specify the maximum resolution of plain images inserted with the TYPO3 image browser.
- Correction for TYPO3 3.8 on Windows server: the path to the locallang file of the spell checker was incorrectly set.
- Addition of Danish translation.
- Update to Russian translation.
- Correction: if multiple instances of the editor were loaded on the same page, with different plugins configuration for each, only the scripts of plugins enabled in the last instance were loaded on the page, resulting in JavaScript errors.
- New Page TSConfig and User TSConfig property enablePersonalDicts may be used to enable the use of personal dictionaries when using the spell checker.
- Correction: some links could not be edited. See http://bugs.typo3.org/view.php?id=1748
- Correction: possible incorrect character set conversion of default titleText (set in language file through anchor accessibility feature configuration) when backend is utf-8.

Version 1.0.0:

- Extension of system class t3lib_parsehtml_proc : move any style attribute of an a-tag to a wrapping span-tag, so that the link may be transformed into a typolink, if no other attribute prevents its transformation. Thanks to Dimitrij Denissenko for this idea and its implementation.
- Correction: updating the properties of an image was broken in version 0.7.6.
- Correction: in the Remove Format window, when some text is selected, the default cleaning area will be “Selection” (rather than “All”). Thanks to Bernhard Kraft for noting this usability issue.
- Method HTMLcleaner_db of class t3lib_parsehtml_proc is modified to honor Page TSConfig setting RTE.default.proc.HTMLparser_db.xhtml_cleaning = 1. See section Known problems.

Version 1.1.0:

- New FAQ about style selector boxes in IE.
- Numerous adjustments for eventual Opera support.
- Add non-breaking space to the palette of the insertcharacter palette. Add hotkety ctrl+space to insert a non-breaking space.
- Restructure the order of characters in the Insert special character popup window.
- Page TSConfig removeTags property would incorrectly remove too many tags.
- New Page TSConfig property keepToggleBordersInToolbar.
- Update of the link to the Mozilla/Firefox AllowClipboardHelper extension. The new version has new features, new translations and will work with Firefox 1.5.
- New TYPO3HtmlParser plugin and enableWordClean configurability.
- Add support for header cells (th tag) and for table sections (thead and tfoot tags) in table operations.
- Update all skins for th elements and scope attribute.
- Add support for all Page TSConfig properties supported by Classic RTE: RTE.classes, classesTable, classesTD, classesLinks, mainStyleOverride, mainStyleOverride_add, mainStyle_font, mainStyle_size, mainStyle_color, mainStyle_bgcolor, inlineStyle, disableRightClick and disablePCexamples.
- Make Page TSCongfig settings compatible when migrating from Classic RTE to htmlArea RTE.
- Add No font and No size elements if font and size selectors.
- New default configurations: Typical, Advanced, Demo. Extension configuration variable **Enable all editor options** is replaced by **Default configuration settings**.
- Correction for Mozilla/Firefox: when an email address is automatically recognized, the created link would be wrong if the local part of the email address contained parts with only one letter.
- Correction: some icons were missing in the element browser when adding or editing links.
- Correction for IE: the link button did not get activated when text was selected using keyboard arrows.
- Correction: The title and alt attributes produced by the link creation/editing button could undergo incorrect

character sets conversions in some configurations of backend and/or frontend character sets.

- Add Page TSConfig properties to configure table operations dialogs: disableAlignmentFieldsetInTableOperations, disableSpacingFieldsetInTableOperations, disableColorFieldsetInTableOperations, disableLayoutFieldsetInTableOperations and disableBordersFieldsetInTableOperations
- New Page TSConfig property ignoreMainStyleOverride for ease of migration from Classic RTE to htmlArea RTE.
- Correction to some cleaning regular expressions in RemoveFormat plugin. Thanks to Ernesto Baschny.
- Keep character map window open until explicitly closed.
- Feature for Mozilla/Firefox: apply style to multiple selected cells of a table.
- Correction to the extraction of class selectors from some css rules.
- Extend t3lib_parsehtml_proc only if TYPO3 version is smaller than 4.0
- Updates to Brazilian, Danish and Finnish localizations.
- Add Slovak localization.
- Modify configuration and module scripts so that the extension may be run as system, global or local extension. Restructure extension modules and resources folders.
- Convert some files to UNIX linebreaks.
- Make extension conflict with extension rte_conf.
- Reconfigure to hook on new version of extension Lorem Ipsum.
- Convert all localization files to xml format. From this version, TYPO3 3.7.0+ is required.

Version 1.1.1:

- Correction to path settings affecting image drag & drop and file/image upload.

Version 1.1.2:

- Correction to path settings affecting CSS files when TYPO3 is installed in a subdirectory.
- Cache the main JS files in typo3temp directory with hashed name including the extension version number.
- Adjust opening size of some popup windows.
- Correction to the automatic detection of url's in Mozilla/Firefox: the rightmost domain should be alphabetic.
- Enable link editing options in the context menu when the cursor is positioned inside a link.
- Enable the link button in the tool bar if some text is selected or if the cursor is positioned inside a link.

Version 1.1.3:

- Remove TYPO3HtmlParser plugin when the editor is used in the frontend.
- Correct height, margins and padding of toolbar select fields in all editor skins.
- Increase default width of the editor area from 460px to 530px.
- Correction for Mozilla/Firefox: remove ending > from auto-detected email link.
- Cleaning code and adding hook in TYPO3HtmlParser plugin (mod6/parse_html.php).

Version 1.1.4:

- Improved link auto-detect in Mozilla/Firefox.
- Fix issue 2358: http://bugs.typo3.org/view.php?id=2358
- Correction to the behaviour of popups.
- Correction to the QuickTag plugin (attributes of hx tags).
- Change opening default of Acronym plugin to abreviation.
- Reduce font size of context menu in all skins.
- Work around Mozilla bug to force display update when table borders change.
- Adjust opening height of SpellChecker window.

Version 1.1.6:

- Enable the use of the RTE in IE 7.
- Change name of the “Advanced” default settings to “Minimal” default settings.
- Remove the title field from the link creation window when the version of TYPO3 is smaller than 3.8.2.
- Correct default processing configuration for the case of center, font, strike and u tags when encountered outside paragraphs.
- Replace true linebreaks with space inside block tags and after hr tags (see bugtracker issue 2365).
- Correction: textstyle selection list (plugin InlineCSS) was not honoring property showTagFreeClasses.

Version 1.1.7:

- Correction: incorrect default styles were assigned to links other than internal.
- Allow style attribute on span tag in the Demo default settings.

Version 1.1.8:

- Correction: RemoveFormat plugin was removing too many tags and attributes. Thanks to Vincent Giroux.
- Remove conflicts with obsolete extensions deleted from TER.

Version 1.2.0:

- Remove Word 2003 o:p tags on paste with enableWordClean and/or save.
- Path setting correction in SpellChecker plugin.
- Fix bugtracker issue 2656.

Version 1.3.0:

- Add $RTEWrapStyle variable to frontend plugin class.tx_rtehtmlarea_pi2.php (bugtracker issue 2760).
- Use new TS parsing := for cumulative sets of configuration defaults.
- Requires TYPO3 4.0.

Version 1.3.1:

- Correction to default value of $RTEWrapStyle variable in frontend plugin.
- Fix bugtracker issue 2682.
- Fix bugtracker issue 2241: Error displaying htmlarea in the frontend. **API change: extensions using the RTE in the frontend must be adjusted! See tutorial section.**
- Remove class.ux_t3lib_parsehtml_proc.php
- Remove support for versions older than 4.0 and clean code.
- Adjustment to default settings for FE plugin.
- Ensure IE7 always uses native XMLHttpRequest rather than ActiveX.
- Fix bugtracker issue 2820: link creation did not work when RTE.default.anchorClasses was empty.

Version 1.3.4:

- Correction to configuration of enableWordClean. TYPO3HtmlParser was loaded even when no parser configuration was provided. See bugtracker issue 2530.
- Correction to server-based word cleaning when backend is not utf-8 and PHP is compiled with mbstring.
- Merge remaining documentation elements from manual of extension Classic RTE (rte).
- Cosmetic adjustments for compatibility with t3skin system extension.

Version 1.3.5:

- Further correction to server-based word cleaning . See bugtracker issue 2530.

Version 1.3.6:

- Correction to RemoveFormat plugin.
- Link entered in the RTE would not be rendered correctly in the frontend when using locale tr_TR. See bugtracker issue 1303.

Version 1.3.7:

- Fix bugtracker issue 3120: transforming text in link in Firefox could take excessive time.
- Fix bugtracker issue 3060: RTE includes a page from the Frontent when pasting text with

enableWordClean configured with server-based parsing, and TYPO3 installed in sub-directory.