

Rhythmyx 7.3 - TinyMCE Control

The content of this document serves to introduce to TinyMCE editor, its implementation, instructions to setup to use TinyMCE control for inline content editing in Rhythmyx.

Note: Current TinyMCE version integrated in Rhythmyx: 4.0.16

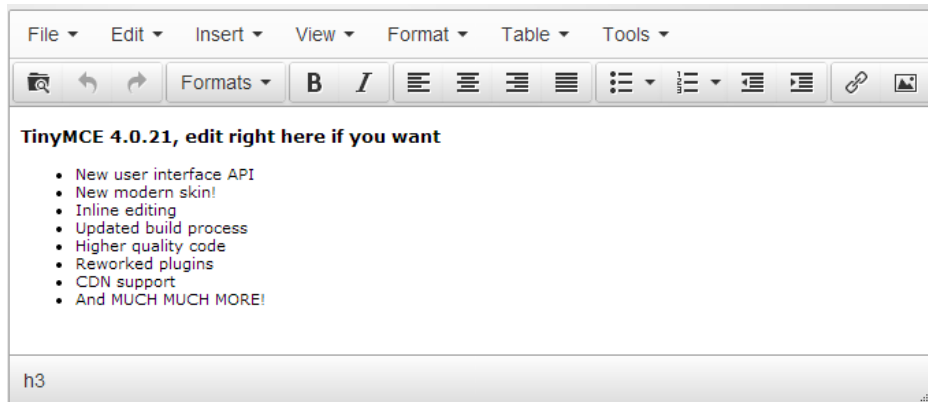
Introduction to TinyMCE

TinyMCE is a platform independent web based JavaScript HTML WYSIWYG editor control.

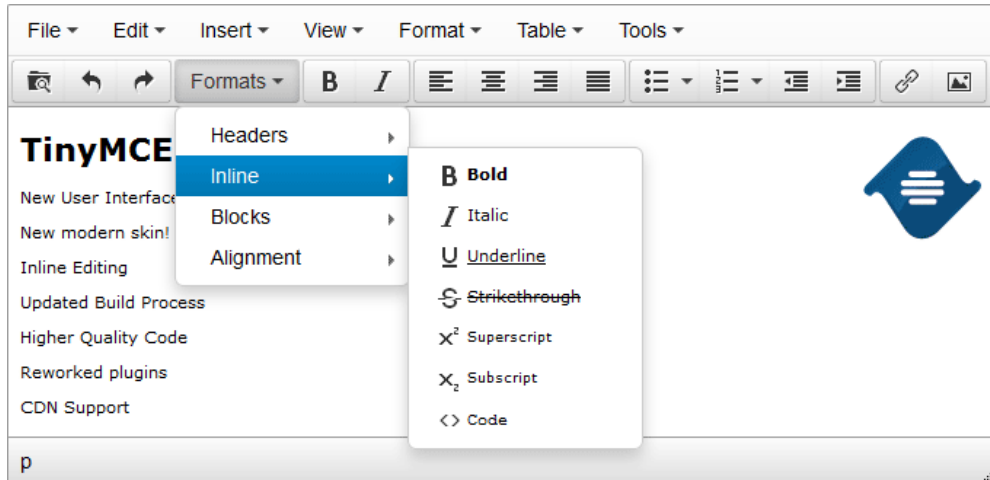
TinyMCE Features:

1. Modern Look and Feel
2. Easy to configure menu options
3. Configurable plugins for customizable features
4. Easy and smart GUI
5. Easier table creation tools

1. Modern Look and Feel

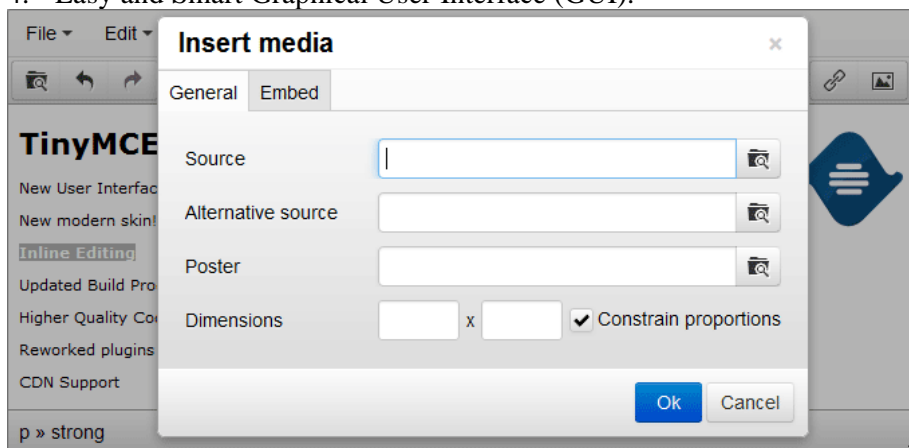


2. Tools/menu options in TinyMCE editor can be customized by configurable options or plugins. For example, adding “format” option in configuration file would allow formatting options like bold, italic, underline, ordered/unordered lists, alignments, etc. Complete list of [Core TinyMCE options](#) is listed in Appendix – A of this document.

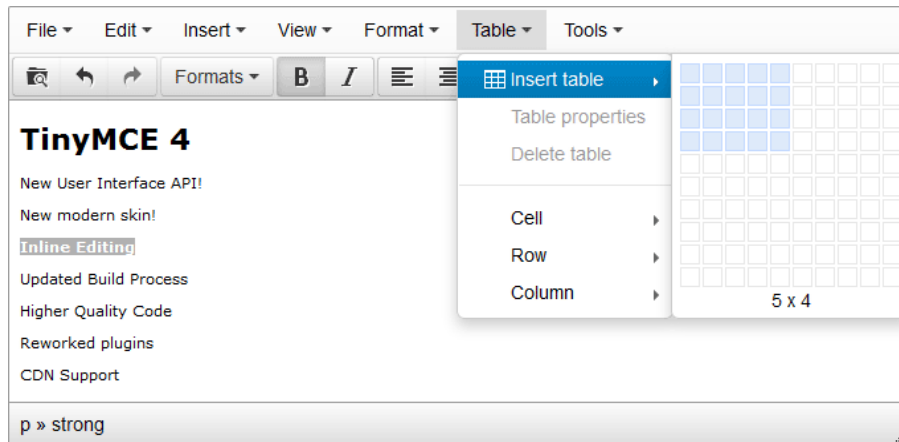


3. TinyMCE provides various configurable plugins to allow added functionality. [Custom plugins](#) can also be created in TinyMCE. Complete list of [TinyMCE Plugins](#) and [TinyMCE plugin options](#) is listed in Appendix – B of this document.

4. Easy and Smart Graphical User Interface (GUI).



5. Table plug-in: Much easier to use table tools are available in TinyMCE, such as the use of a grid interface for creation of tables.



TinyMCE implementation in Rhythmyx

In Rhythmyx 7.3, TinyMCE is an optional control (sys_TinyMCE) which is mainly implemented to replace existing Java based sys_EditLive control for content editor. Rhythmyx integration of TinyMCE as optional control allows having a choice between TinyMCE and EditLive controls for content editing. In addition to TinyMCE's core functionality Rhythmyx provides additional controls for inline content like, inline link, Image and template which is same as that EditLive offers.

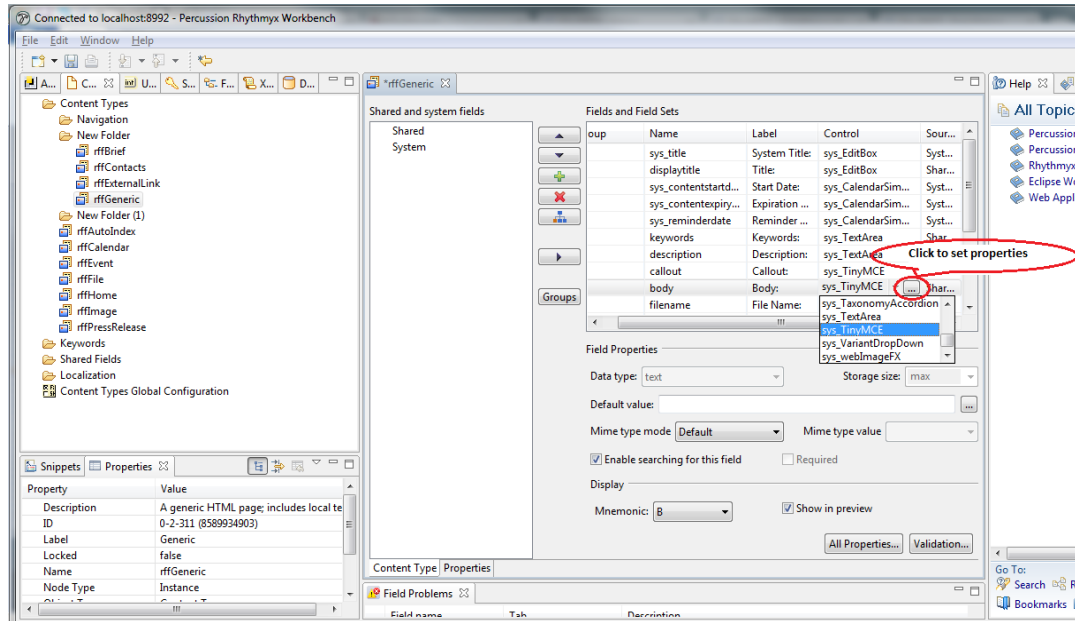
TinyMCE editor setup in Rhythmyx:

1. Requires Rhythmyx7.3 workbench:

Using sys_TinyMCE control requires latest 7.3 workbench tool. If the workbench is not upgraded to 7.3 version, following link can be used to upgrade the Rhythmyx tools:
<http://{host}:{port}/Rhythmyx/Administration/DeveloperTools/InstData/Windows/VM/devToolsSetup.exe>

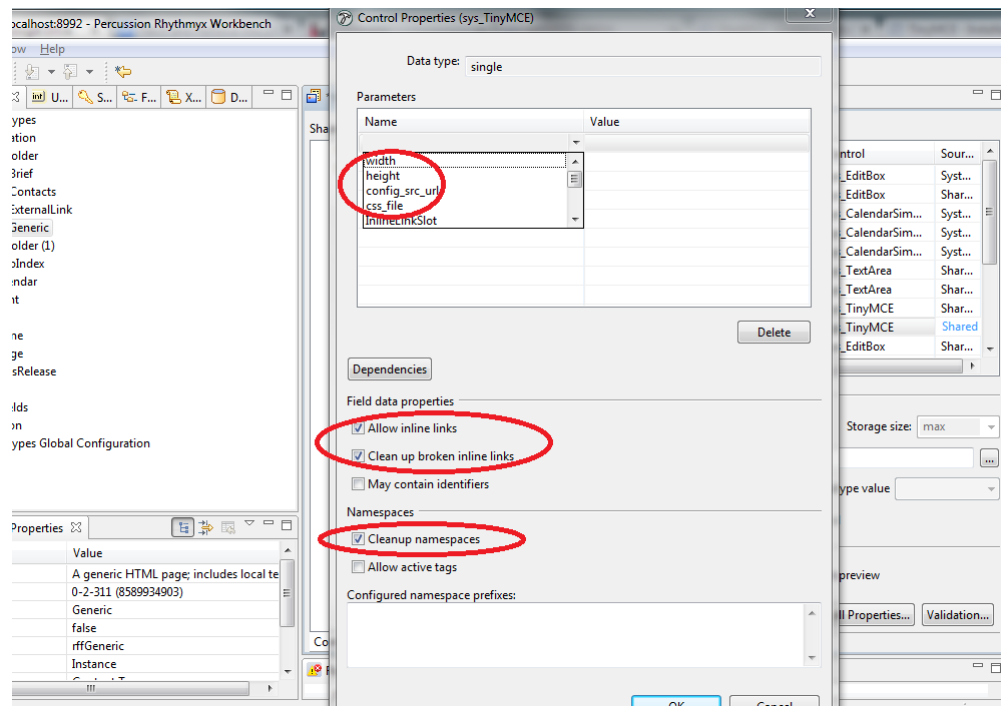
2. Add TinyMCE control in workbench:

In Workbench/Content Design tab, in a content type, choose sys_TinyMCE as control type



3. Configure parameters if necessary:

Following field parameters can be configured as needed by using ... (ellipses) of the selected TinyMCE control field:



Note : As shown in above screenshot, 'Field Data Properties' and 'Namespaces' are mandatory to be selected for TinyMCE control.

Parameter	Purpose	Default value	Note
height	Defines height of the TinyMCE control in	100%	Absolute height can be configured. Eg: 200px

	content editor		
width	Defines width of the TinyMCE control in content editor	350px	Absolute width can be configured. Eg: 600px
config_src_url	Defines configuration file to customize TinyMCE options	<p>Configuration file name path as “literal value”.</p> <p>Eg: ../rx_resources/tinymce/js/config/tinyMceConfig.json</p> <p>Note:</p> <ul style="list-style-type: none"> • If no value specified, the Rhythmyx/rx_resources/tinymce/js/config/default_config.json will be utilized. • If specified, the configured file will replace the default_config.json file if one such exists. 	<ul style="list-style-type: none"> • The custom file location can only be under ../rx_resources/tinymce/js/config/ folder. Failing to place the file in this location will result in JavaScript error in content editor with TinyMCE control. • Sample TinyMCE configuration is available in Rhythmyx\sys_resources\tinymce\ location.
css_file	Defines css to override the default skin of TinyMCE	<p>CSS file name path.</p> <p>Eg: ../rx_resources/tinymce/customTinyMCE.css</p> <p>Note:</p> <ul style="list-style-type: none"> • If no value specified, the Rhythmyx/sys_resources/css/tinymce.css will be utilized. • If specified, the css provided in Rhythmyx/sys_resources/css/tinymce.css will be merged with css provided in Rhythmyx/rx_resources/tinymce/<filename.css> file. 	<ul style="list-style-type: none"> • The custom css file location ../rx_resources/tinymce/ need to be maintained which otherwise will result in JavaScript error in content editor with TinyMCE control • Sample css files (tinymce.css and tinymce1.css) are provided under Rhythmyx/rx_r

			resources/tinymce/ location.
--	--	--	------------------------------

4. Copy default TinyMCE configuration file:

Similar to EditLive configuration file (elj_config.xml), TinyMCE requires default_config.json file in Rhythmyx/rx_resources/tinymce/js/config folder as default configuration file. Any TinyMCE field without custom configuration file requires the default configuration file. If custom configuration file is required for a specific TinyMCE field, such configuration file can be configured in respective field in content type. In such case the default_config.json will be replaced by the configured file. Note that, if custom configuration is not required, at least empty json node ({})) in default_config.json file is required. In case of empty default_config.json file, the TinyMCE is loaded with all default menu and toolbar options shipped in Rhythmyx.

Sample configurations:

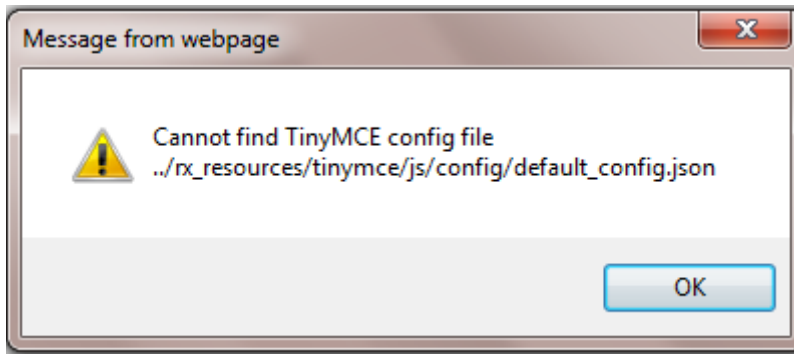
a. Empty configuration:

```
[
  {
  }
]
```

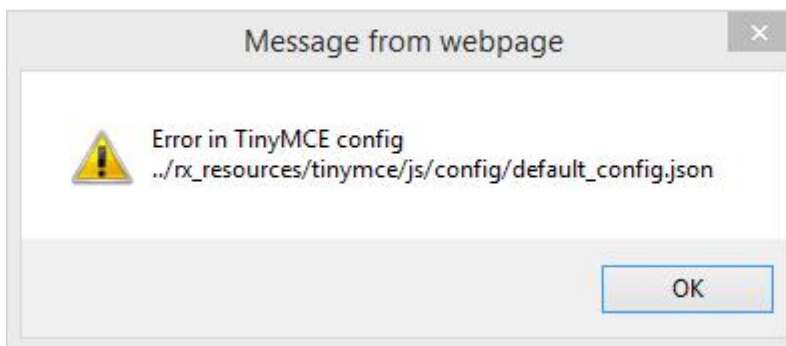
- b. Two set of configurations in a given json file: The first set of configurations applies to all users as “roles” option is specified. The second set is for Admin users only which includes “roles” option for “Admin” role only. In this example Admin role has access to source code of the content by having “code” plugin included in the configs list.

```
[
  {
    "config_css": "../rx_resources/tinymce/tinymce1.css",
    "plugins": [
      "advlist autolink lists link charmap print preview hr anchor pagebreak searchreplace wordcount
      visualblocks visualchars fullscreen insertdatetime media nonbreaking save table contextmenu directionality
      emoticons template paste textcolor noneditable rxinline"
    ]
  },
  {
    "roles": ["Admin"],
    "config_css": "../rx_resources/tinymce/tinymce1.css",
    "plugins": [
      "advlist autolink lists link charmap print preview hr anchor pagebreak searchreplace wordcount
      visualblocks visualchars code fullscreen insertdatetime media nonbreaking save table contextmenu
      directionality emoticons template paste textcolor noneditable rxinline"
    ]
  }
]
```

Please note that failing to place default_config.json file in Rhythmyx/rx_resources/tinymce/js/config folder for custom TinyMCE configurations would result in following error message when content editor with TinyMCE control is loaded.



Please note that if any TinyMCE configuration error, the content editor wouldn't load TinyMCE and would display follow error message:



Creating TinyMCE configuration file:

- Create a json file(.json extension) with TinyMCE configurations and save it under Rhythmyx/rx_resources/tinymce/js/config/ folder.
- Json configuration rules:
 - The complete set of configurations must be enclosed in square brackets ([]).
 - A set of configurations need to be enclosed in curly brackets ({}). There can be more than one set of configurations in a given json configuration file.
 - The TinyMCE configuration option and its associated value need to be enclosed by double quotes and separated by colon ("config_css" : "../rx_resources/tinymce/tinymce1.css")
 - The TinyMCE configuration options and set of configuration must be separated by comma(,).
 - Nested Json object configuration example: In case of nested json configurations consisting of map elements, need to configure with care enclosing every key:value pair in double quotes which may differ from the sample configuration provided in TinyMCE documentation. For example following is sample configuration for customizing TinyMCE editor menu items which has nested map elements in the configuration. The following example is complete set of configuration to include default TinyMCE menu items. Any menu items can be removed/Updated by just removing/updating desired "items" value. We can rename default menu title by updating the "title" value.

```
"menu":{ "file" : { "title" : "File" , "items" : "newdocument" },
        "edit" : { "title" : "Edit" , "items" : "undo redo | cut copy paste pastetext | selectall" },
        "insert" : { "title" : "Insert", "items" : "link media | template hr" },
        "view" : { "title" : "View" , "items" : "visualaid" },
        "format" : { "title" : "Format", "items" : "bold italic underline strikethrough superscript
        subscript | formats | removeformat" },
        "table" : { "title" : "Table", "items" : "inserttable tableprops deletetable | cell row column" },
        "tools" : { "title" : "Tools", "items" : "spellchecker code" }
    }
```

- Menubar can be configured to not to include specific menu as shown in below configuration. The following sample would include all default menu available in TinyMCE. To remove one of the menu we need to remove the desired menu for the list. For exaple not including “table” in the congiguration would remove “table” menu in TinyMCE editor.

```
"menubar": "file tools table format view insert edit"
```

- **Rhythmyx Options:** Rhythmyx ships with three TinyMCE options: *roles*, *langmap* and *rxinline*

- **Rhythmyx Options – roles:** is to assign TinyMCE configuration for specific Rhythmyx user roles – If *roles* option not specified in a given set of configuration, such configuration are available to all Rhythmyx users. The *roles* option value need to be associated with one or more Rhythmyx roles. Multiple roles can be configured as “roles”: [“Admin, EI_Admin_Members, Default”]
- **Rhythmyx Options – langmap :** is to configure desired TinyMCE locale. By default TinyMCE is loaded in en_us locale. To configure TinyMCE in other than en_us locale:
 - Make sure respective locale is enabled in Rhythmyx.
 - From TinyMCE internationalization page, download desired locale language pack.
 - Exact the TinyMCE language package for the locale’s JavaScript file and save the JS in Rhythmyx/rx_resources/tinymce/js/tinymce/langs folder.
 - Identify required TinyMCE i18n code for specific language pack from TinyMCE internationalizations page. For example, for French TinyMCE lang code is fr_FR.
 - Add required langmap: { “<system locale code>”: “<tinymce specific locale code> “} Rhythmyx TinyMCE option in TinyMCE configuration file. For example, for French TinyMCE lang code is fr_FR. This value need to be mapped to system locale code in TinyMCE configuration as “langmap”: {“fr_fr”:”fr_FR”}
 - In case of more than one locale, the mapping need to be separated by comma in the langmap option as : “langmap”: {“fr_fr”:”fr_FR”, “en-ES”:”es”}
- **Rhythmyx Plugin - rxinline:** includes Rhythmyx inline controls such as inline link, image and variant in menu and in toolbar.

- The config_css option can be used to configure css file. More than one css file can be configured which need to be separated by comma.
 - Example: “./rx_resources/tinymce/tinymce.css , ./rx_resources/tinymce/tinymce1.css”
- If custom configurations are in place to override default system plugins configuration, it is required to have “noneditable rxinline” plugin configurations in the custom configuration file. This configuration is required for Rhythmyx inline options.
- Complete set of TinyMCE default options provided by TinyMCE is provided in Appendix – A of this document.

Appendix – A

Core TinyMCE options

General

- [auto_focus](#)
- [directionality](#)
- [browser_spellcheck](#)
- [language](#)
- [language_url](#)
- [nowrap](#)
- [object_resizing](#)
- [plugins](#)
- [external_plugins](#)
- [selector](#)
- [skin](#)
- [skin_url](#)
- [theme](#)
- [theme_url](#)
- [inline](#)
- [hidden_input](#)

Cleanup/Output

- [convert_fonts_to_spans](#)
- [custom_elements](#)
- [doctype](#)
- [element_format](#)
- [entities](#)
- [entity_encoding](#)
- [extended_valid_elements](#)
- [fix_list_elements](#)
- [font_formats](#)
- [fontsize_formats](#)
- [force_p_newlines](#)
- [force_hex_style_colors](#)
- [forced_root_block](#)
- [forced_root_block_attrs](#)
- [formats](#)
- [indentation](#)
- [invalid_elements](#)
- [keep_styles](#)
- [protect](#)
- [schema](#)
- [style_formats](#)
- [block_formats](#)
- [valid_children](#)
- [valid_elements](#)

Content style

- [body_id](#)
- [body_class](#)
- [content_css](#)

Visual aids

- [visual](#)
- [visual_table_class](#)
- [visual_anchor_class](#)

Undo/Redo

- [custom_undo_redo_levels](#)

User interface

- [toolbar](#)
- [toolbar<N>](#)
- [menubar](#)
- [menu](#)
- [statusbar](#)
- [resize](#)
- [width](#)
- [height](#)
- [preview_styles](#)
- [fixed_toolbar_container](#)
- [event_root](#)

URL

- [convert_urls](#)
- [relative_urls](#)
- [remove_script_host](#)
- [document_base_url](#)
- [allow_script_urls](#)

Callbacks

- [file_browser_callback](#)
- [init_instance_callback](#)

- [valid_styles](#)
- [setup](#)

Appendix – B

TinyMCE Plugins

- [advlist](#)
- [anchor](#)
- [autolink](#)
- [autosize](#)
- [autosave](#)
- [bbcode](#)
- [charmap](#)
- [code](#)
- [compat3x](#)
- [contextmenu](#)
- [directionality](#)
- [emoticons](#)
- [example](#)
- [example_dependency](#)
- [fullpage](#)
- [fullscreen](#)
- [hr](#)
- [image](#)
- [insertdatetime](#)
- [layer](#)
- [legacyoutput](#)
- [link](#)
- [lists](#)
- [importcss](#)
- [media](#)
- [nonbreaking](#)
- [noneditable](#)
- [pagebreak](#)
- [paste](#)
- [preview](#)
- [print](#)
- [save](#)
- [searchreplace](#)
- [spellchecker](#)
- [tabfocus](#)
- [table](#)
- [template](#)
- [textcolor](#)
- [visualblocks](#)
- [visualchars](#)
- [wordcount](#)

TinyMCE Plugin options

Advlist plugin

- [advlist_bulldlist_styles](#)
- [advlist_numlist_styles](#)

Link plugin

- [link_class_list](#)
- [link_list](#)
- [target_list](#)
- [rel_list](#)
- [link_title](#)

Image plugin

- [image_class_list](#)
- [image_list](#)
- [image_advtab](#)
- [image_description](#)
- [image_dimensions](#)

Importcss plugin

- [importcss_append](#)
- [importcss_file_filter](#)
- [importcss_selector_filter](#)
- [importcss_groups](#)
- [importcss_merge_classes](#)
- [importcss_selector_converter](#)

Fullpage plugin

- [fullpage_default_doctype](#)
- [fullpage_default_encoding](#)
- [fullpage_default_title](#)
- [fullpage_default_font_size](#)
- [fullpage_default_font_family](#)
- [fullpage_default_text_color](#)
- [fullpage_default_langcode](#)
- [fullpage_default_xml_pi](#)
- [fullpage_hide_in_source_view](#)

Tabfocus plugin

Template plugin

- [templates](#)
- [template_cdate_classes](#)
- [template_mdate_classes](#)
- [template_cdate_format](#)
- [template_mdate_format](#)
- [template_selected_content_classes](#)
- [template_replace_values](#)
- [template_popup_width](#)
- [template_popup_height](#)
- [template_preview_replace_values](#)

AutoSave plugin

- [autosave_interval](#)
- [autosave_restore_when_empty](#)
- [autosave_retention](#)

AutoResize plugin

- [autoresize_min_height](#)
- [autoresize_max_height](#)

Save plugin

- [save_enablewhendirty](#)
- [save_onsavecallback](#)
- [save_onsavecancelcallback](#)

Noneditable plugin

- [noneditable_editable_class](#)
- [noneditable_noneditable_class](#)
- [noneditable_leave_contenteditable](#)

Insertdatetime plugin

- [insertdatetime_dateformat](#)
- [insertdatetime_formats](#)
- [insertdatetime_timeformat](#)
- [insertdatetime_element](#)

Spellchecker plugin

- [tabfocus elements](#)

Table plugin

- [table clone elements](#)
- [table grid](#)

Pagebreak plugin

- [pagebreak separator](#)

Visualblocks plugin

- [visualblocks default state](#)

Paste plugin

- [paste data images](#)
- [paste as text](#)
- [paste preprocess](#)
- [paste postprocess](#)
- [paste word valid elements](#)
- [paste webkit styles](#)
- [paste retain style properties](#)

- [spellchecker wordchar pattern](#)
- [spellchecker rpc url](#)
- [spellchecker language](#)
- [spellchecker languages](#)
- [spellchecker callback](#)

Contextmenu plugin

- [contextmenu](#)

Media plugin

- [media alt source](#)
- [media poster](#)
- [media dimensions](#)

Code plugin

- [code dialog width](#)
- [code dialog height](#)

Appendix – C

EditLive features those are not available in TinyMCE:

Mathematical equations
Accessibility checking as you type
Insert Page Break
Spell check/autocorrect
Thesaurus
Accessibility Report
Comments and Track Changes
Debug Logging
Form Creation
HTML5 - Canvas
HTML5 - Keygen
HTML5 - Progress tag
XML with custom namespaces

TinyMCE references:

Core TinyMCE: <http://www.tinymce.com/index.php>

TinyMCE demo: <http://www.tinymce.com/tryit/full.php>

TinyMCE configurations: <http://www.tinymce.com/wiki.php/Configuration>

TinyMCE custom plugin: http://www.tinymce.com/wiki.php/Tutorials:Creating_a_plugin