

CKEditor integration in Magnolia 5

 The Magnolia 5 series uses CKEditor for rich-text editing. This page highlights the most important milestones of the CKEditor integration since Magnolia 5.0, as well as supported configuration options in the [Rich text field](#) definition.

- [CKEditor integration since Magnolia 5.0](#)
- [Alternate Java-based configuration](#)
- [Switching to CKEditor in Magnolia 4.5](#)

CKEditor integration since Magnolia 5.0

Version	Notable changes	JIRA ticket
Magnolia 5.0	initial CKEditor integration, replaces the FCKEditor module in Magnolia 4.5	-
Magnolia 5.2.2	<ul style="list-style-type: none"> • enabled basic configuration options • enabled usage of custom <code>config.js</code> file 	MGNLUI-2403 - Getting issue details... STATUS
Magnolia 5.3.1	<ul style="list-style-type: none"> • consolidated initialization of CKEditor in the browser (e.g. working with external plugins, running multiple editors) 	MGNLUI-3052 - Getting issue details... STATUS
Magnolia 5.3.4	<ul style="list-style-type: none"> • enabled Magnolia plugins over JS configuration • provided Magnolia's default rich-text field configuration as a <code>config.js</code> file, as a starting point to your customization. 	MGNLUI-3150 - Getting issue details... STATUS

Config overview

For the list of supported config options, along with example values, please head to the [Rich text field](#) documentation page.

	FCKEditor Magnolia 4.5	CKEditor Magnolia 5.0	CKEditor Magnolia 5.2.2+	Outcome
<i>Basic configuration</i>				
alignment	 		 	enables "justify" plugin
images	 	 	 	enables "image" plugin
- images from dam		always both images and dam assets		enables fork of "filebrowser" plugin to invoke assets chooser
fileUpload	 			no more direct upload in CKEditor
lists	 		 	enables "enterkey", "indent" and "list" plugins (dependency chain for list)
source	 		 	enables "sourcearea" plugin
spellChecker				
showSpellChecker	 			

tables	✓	✗ always on	✓	enables "table" and "tabletools" plugin
width	✓ 543	-	-	always 100% of parent Vaadin layout
height	✓ 200	✗	✓	sets height of Vaadin component
colors	✓	✗	✓	sets "colorButton_colors" and "colorButton_enableMore" config options, enables "colorbutton" plugin, disables "colordialog" plugin
fonts	✓	✗	✓	enables "font" and "fontSize" plugins, sets "font_names" config option, enables "Font" button
fontSizes	✓	✗	✓	enables "font" and "fontSize" plugins, sets "fontSize_sizes" config option, enables "FontSize" button
enterMode	✓ undocum ented	-	-	
shiftEnter Mode	✓ undocum ented	-	-	
<i>Advanced configuration</i>				
configsJsFile	✓ called jsConfigFile	✗	✓	sets "customConfig" config option
jsInitFile	✓	-	-	no longer applies to CKEditor
css	✓	✗	✗	use custom config.js file, see "contentsCss"
styles	✓	✗	✗	use custom config.js file, see "stylesSet"
templates	✓	✗	✗	use custom config.js file

Definition vs. JS configuration

The set of config options that are offered by the `RichTextFieldDefinition` is mostly intended for a quick and easy customization of the editor.

As soon as you have more advanced requirements, you may switch to a *configsJsFile*.

i When using a *configsJsFile*, all other definition options are ignored. This is intended because otherwise, they override any configuration done in the JS files.

Alternate Java-based configuration

As of Magnolia 5.0, customizing the `MagnoliaRichTextField` can also be done the Java way, with the following steps:

1. Create a new *fieldType*
 - set *definitionClass*, e.g. `MyRichTextFieldDefinition`
 - set *factoryClass* `MyRichTextFieldFactory` extending magnolia's `RichTextFieldFactory`
2. Override `#initializeCKEditorConfig()`
 - Return a custom extended `MagnoliaRichTextFieldConfig`
 - this is our default config / impl of the Vaadin ckeditor wrapper add-on's `CKEditorConfig`
 - it exposes common CKEditor config / can also set options as plain Json strings
3. Set some field class to `MyRichTextFieldDefinition` in any dialog definition

Although it is technically possible, it has the following limitations:

- One has to implement a Java config class from the Vaadin wrapper + our own definition & factory classes to customize the appearance/feature set of the CKEditor.
 - Thus requiring changes in your module at every config change.
- For each different configuration, one must **register a new field type** and create a field factory

Switching to CKEditor in Magnolia 4.5

The [CKEditor Module](#) for Magnolia 4.5 is available in the forge.

It is very similar to the legacy FCKEditor module; it simply provides the CKEditor 4.x library, and offers a set of config options.

 Note that it is however totally independent from the CKEditor integration in Magnolia 5, and that we do not provide any automatic upgrade to Magnolia 5 for its usages (using only the *config.js* file is then your safest bet).