## **Description**

The **Rich Text Editor** control provides a number of tool items that provide a rich look to the text entered in the editing area. It brings to the Web popular editing features found in the desktop word processors such as the **Microsoft Word** and the **OpenOffice.org Writer.**

## **Solution**

***To enable the standard toolbar items:***

The **Rich Text Editor** control has a list of toolbar items to format the content of the **Rich Text Editor** and also to add the custom tool in the **RTE** toolbar to achieve your own functionality in the **Rich Text Editor** control. You can enable the list of built-in toolbar items in the **RTE** control through the **tools** option present in the **RTE** control.

***List of Standard Toolbar Items*:**

1. Alignment Formatting Tools: Left, right, center, and justify.
2. Color palette:

* Fore-color: To change the color of the text in the editing area.
* Back-color: To change the background color of the editing area.

1. Bulleting: Ordered and unordered list.
2. Style: Bold, Italic, Underline, and Strikethrough.
3. Subscript and Superscript: Text with subscript and superscript placed after or before the baseline respectively.
4. Font options.

* [Font Name](https://docs-old.ckeditor.com/CKEditor_3.x/Users_Guide/Styling/Font) - Typeface that is applied to the document text.
* [Font Size](https://docs-old.ckeditor.com/CKEditor_3.x/Users_Guide/Styling/Size) - Determines how big or small the text should be.

1. Format Style: Predefined sets of formatting features that can be applied to block and to make elements of the document inline.
2. Order list and unorder list
3. Clipboard actions like cut, copy, paste
4. Clear format and clear content
5. Insert hyperlink, image, video
6. Adding and customizing the table
7. Upper and lower case conversion.
8. Indent.
9. Undo and Redo operations.
10. Full screen mode

***To enable the Custom toolbar items*:**

In the **Rich Text Editor**, toolbars are customizable. When you want to include a new tool item with a new function that is not available in the existing **RTE** toolbar items, create a new tool item by using the **customTools** option in the **RTE** control and enable this custom toolbar item through the **tools** option present in the **RTE** control.

|  |
| --- |
| // Custom Toolbar Items.  **customTools: [{**  **text: "saveTool",**  **tooltip: "Save Button",**  **css: "saveButton",**  **action: function () {**  // Writes your action code block.  **alert("Values Saved Successfully");**  **}**  **}]** |

***Custom Toolbar item*:**

* Save operation

In the following section, the way to enable the **RTE** standard toolbar items along with the custom toolbar items in the **RTE** control is explained.

**HTML**

|  |
| --- |
| <textarea id="rteSample" rows="10" cols="30" style="width: 740px; height: 440px">  &lt;p&gt;&lt;b&gt;Description:&lt;/b&gt;&lt;/p&gt;  &lt;p&gt; The Rich Text Editor (RTE) control is an easy to render in  client side. Customer easy to edit the contents and get the HTML content for  the displayed content. A rich text editor control provides users with a toolbar  that helps them to apply rich text formats to the text entered in the text  area. &lt;/p&gt;  &lt;p&gt;&lt;b&gt;Functional  Specifications/Requirements:&lt;/b&gt;&lt;/p&gt;  &lt;ol&gt;&lt;li&gt;&lt;p&gt;Provide  the tool bar support, it’s also customizable.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Options  to get the HTML elements with styles.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Support  to insert image from a defined path.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Footer  elements and styles(tag / Element information , Action button (Upload, Cancel))&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Re-size  the editor support. &lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Provide  efficient public methods and client side events.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Keyboard  navigation support.&lt;/p&gt;&lt;/li&gt;&lt;/ol&gt;  </textarea> |

**JavaScript**

|  |
| --- |
| $("#rteSample").ejRTE({  toolsList: ["formatStyle", "font", "style", "scripts", "alignment", "lists", "indenting", "copyPaste", "doAction", "clear", "links", "images", "tables", "casing","customTools"],  tools: {  font: ["fontName", "fontSize", "fontColor", "backgroundColor"],  style: ["bold", "italic", "underline", "strikethrough"],  alignment: ["justifyLeft", "justifyCenter", "justifyRight", "justifyFull"],  lists: ["unorderedList", "orderedList"],  copyPaste: ["cut", "copy", "paste"],  doAction: ["undo", "redo"],  clear: ["clearFormat", "clearAll"],  links: ["createLink"],  images: ["image", "video"],  tables: ["createTable", "addRowAbove", "addRowBelow", "addColumnLeft", "addColumnRight", "deleteRow", "deleteColumn", "deleteTable"],  scripts: ["superscript", "subscript"],  casing: ["upperCase", "lowerCase"],  paragraph: ["paragraph"],  // Custom Toolbar Items.  **customTools**: [{  name: "saveTool",  tooltip: "Save Button",  css: "saveButton",  text:"saveButton",  action: function () {  // Writes your action code block.  alert("Values Saved Successfully");  }  }]  }  }); |

**CSHTML**

|  |
| --- |
| @{List<String> font = new List<string>() { "fontName", "fontSize", "fontColor", "backgroundColor" };  List<String> style = new List<string>() { "bold", "italic", "underline", "strikethrough" };  List<String> alignment = new List<string>() { "justifyLeft", "justifyCenter", "justifyRight", "justifyFull" };  List<String> lists = new List<string>() { "unorderedList", "orderedList" };  List<String> copyPaste = new List<string>() { "cut", "copy", "paste" };  List<String> doAction = new List<string>() { "undo", "redo" };  List<String> clear = new List<string>() { "clearFormat", "clearAll" };  List<String> tables = new List<string>() { "createTable", "addRowAbove", "addRowBelow", "addColumnLeft", "addColumnRight", "deleteRow", "deleteColumn", "deleteTable" };  List<String> links = new List<string>() { "createLink" };  List<String> images = new List<string>() { "image", "video" };  List<String> scripts = new List<string>() { "superscript", "subscript" };  List<String> casing = new List<string>() { "upperCase", "lowerCase" };  List<String> formatStyle = new List<string>() { "format" };  }  @{Html.EJ().RTE("rteSample").Width("100%").ContentTemplate(@<p>  &lt;p&gt;&lt;b&gt;Description:&lt;/b&gt;&lt;/p&gt;  &lt;p&gt;The Rich Text Editor (RTE) control is an easy to render in  client side. Customer easy to edit the contents and get the HTML content for  the displayed content. A rich text editor control provides users with a toolbar  that helps them to apply rich text formats to the text entered in the text  area. &lt;/p&gt;  &lt;p&gt;&lt;b&gt;Functional  Specifications/Requirements:&lt;/b&gt;&lt;/p&gt;  &lt;ol&gt;&lt;li&gt;&lt;p&gt;Provide  the tool bar support, it’s also customizable.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Options  to get the HTML elements with styles.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Support  to insert image from a defined path.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Footer  elements and styles(tag / Element information , Action button (Upload, Cancel))&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Re-size  the editor support. &lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Provide  efficient public methods and client side events.&lt;/p&gt;&lt;/li&gt;&lt;li&gt;&lt;p&gt;Keyboard  navigation support.&lt;/p&gt;&lt;/li&gt;&lt;/ol&gt;  </p>).ShowFooter(true)  .Tools(tool => tool.Clear(clear)  .FormatStyle(formatStyle)  .Tables(tables)  .Links(links)  .Images(images)  .Effects(scripts)  .Casing(casing)  .Font(font)  .Styles(style)  .Alignment(alignment)  .Lists(lists)  .Clipboard(copyPaste)  .DoAction(doAction)  **.CustomTools(custom => custom.Name("saveTool")**  **.Tooltip("Save Button")**  **.Css("saveButton").Text("Save Button")**  **.Action("onSave").Add()))**  .Render();} |

**Script**

|  |
| --- |
| <script>  function onSave() {  alert("Values Saved Successfully");  }  </script> |

Output of the customTools in RTE control.

|  |
| --- |
| RTE output |

Live demo in JS Playground: <http://jsplayground.syncfusion.com/0dcujapa>