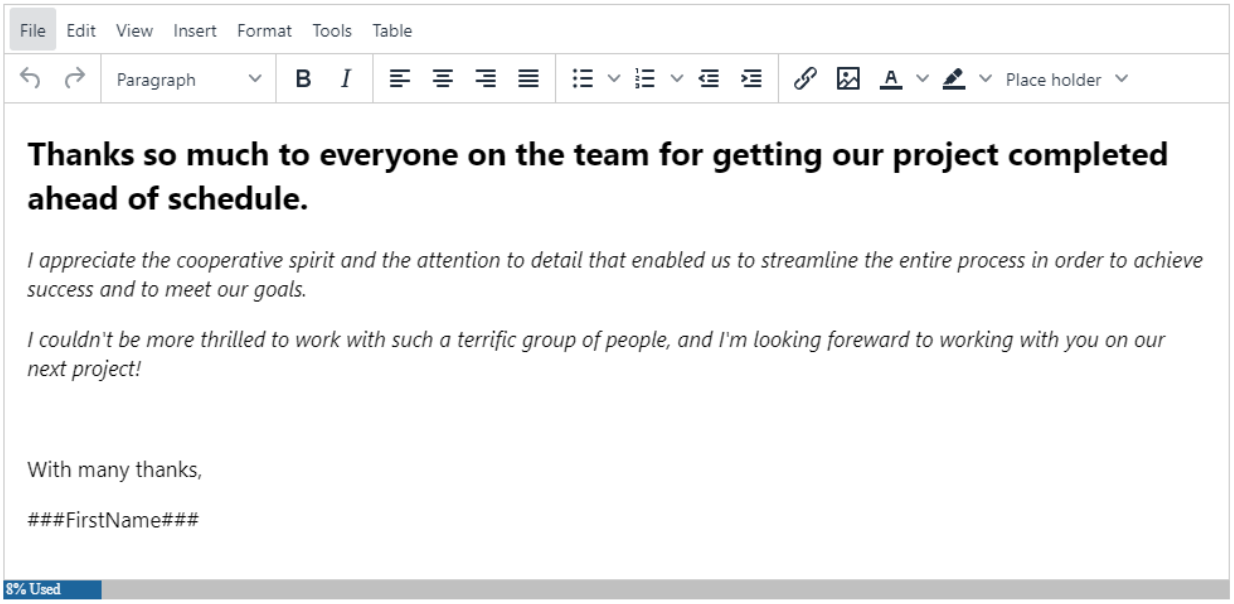
Advanced Rich Text Editor Component:

Function : richTextEditorField()

The Rich text editor component can be used on Appian Screens for Applications where we want to show the Rich Text information in readable format to the end User, Also we can use the component to show it as an email subject typing interface.When compared to the Out of the box rich text editor in this you can add images you can customize tables etc. There is additional functionality called placeholders using which you can use the same information on multiple items in the editor without much typing. Allows uploading of images which get stored in the specified Appian folder (requires separate install of the Vuram Rich Text Editor Connected System Plugin from https://community.appian.com/b/appmarket/posts/vuram-rich-text-editor-image-upload-connected-system).

Note: only the URL to the uploaded image, not the image content itself, is stored in the generated HTML from fn!richTextEditorField(). This means that if you use the generated HTML outside of Appian (e.g., in an email), the image will not display unless the outside viewer is already logged into Appian.



Parameters

Name	Keyword	Type	Description
Label	label	Text	Text to display as the field label.
Label Position	labelPosition	Text	Determines where the label appears. Valid values: “ABOVE” (default), “ADJACENT”, “JUSTIFIED”, “COLLAPSED”.
Height	height	Text	Determines the layout height. Valid values: “SHORT”, “MEDIUM”, “TALL”, “AUTO” (default).
IncludeHTMLength	includeHTMLength	Boolean	Include the HTML format length, if set to “True” and vice versa. default : true.
MaxSize	maxSize	Number (Integer)	Maximum size of the rich text data. Default:10000
Placeholder	placeholder	List of Dictionary	Placeholder text to display in the component wherever you want.
richTextValue	richTextValue	Text	Rich text to display in the field.
richTextSaveInto	richTextSaveInto	List of Save	One or more variables that are updated with the richText value when the user changes it. Use a!save() to save a modified or alternative value to a variable

IsReadOnly	isReadOnly	Boolean	Determines if the field should display as editable or not. Valid Values : True or False
Enable Spell Checker	enableSpellChecker	Boolean	Provide true to enable the spell checker. Default false.
Show Menu Bar	showMenubar	Boolean	If it is true then the menu bar displayed.Default true
Show Usage Bar?	showUsageBar	Boolean	Provide false to hide the usage bar at the bottom of the editor. Default true.
Enable Fullpage Output?	enableFullpageOutput	Boolean	Enabling fullpage exposes <head>, <body> and various meta tags in richText output. Default true.
Allow Image Upload	allowImageUpload	Boolean	Determines if image uploads are allowed. Default: false.
Image Storage Connected System	imageStorageConnectedSystem	Connected System	The instance of the Vuram Rich Text Editor connected system used to store images and export the content as pdf from the rich text editor. This is required if allowImageUpload is set to TRUE or if you want to export the rich text editor content as Pdf document and store it in Appian. You can download the Connected System from the Appian App Market.
Uploaded Images Doc Ids	uploadedImagesDocIds	List of Save	It contains the document ids of all the uploaded images.

Exported PDF Id	exportedPdfId	List of Save	It has the document id of the pdf generated through Export as Pdf feature
Enable export PDF	enableExportPdf	Boolean	Provide false to disable the 'Export as PDF' button. Default: True
Hide Border in Read Only	hideBorderInReadOnly	Boolean	Provide true to hide the border when the editor is in Read Only mode. Default : false
Enable Latex Conversion	enableLatexConversion	Boolean	Provide false to disable 'Convert LaTeX to Mathematical Expression' button. Default: false.

Notes:

- Supported Browsers - Google Chrome, Mozilla Firefox, Internet Explorer11, Edge, Safari.
- Supported Language - English (United States).
- Update the **connected system parameters** in the Demo application.

Example

*Copy and paste the example into the **INTERFACE DEFINITION** in **EXPRESSION MODE** to see how it is displayed.*

```
 richTextEditorField(
     label: "",
     labelPosition: "ABOVE",
     validations: {},
     height: "AUTO",
     richTextValue: local!htmlText,
```

```
richTextSaveInto: local!htmlText,
isReadOnly: false,
showMenubar: true,
showUsageBar: true,
includeHTMLlength:false,
maxSize:2000,
placeholder: {
 {
  name: "Name",
  values: {
   {
    text: "First Name",
    value: "###First Name###"
   },
   {
    text: "Last Name",
    value: "###Last Name###"
   }
  }
 },
 {
  name: "Address",
  values: {
   {
    text: "Permanent Address",
    value: "###PermanentAddress###"
   },
   {
    text: "Temporary Address",
    value: "###TemporaryAddress###"
   }
  }
 }
},
enableSpellChecker: true,
allowImageUpload: true,
imageStorageConnectedSystem: cons!ARTE_CONST_FileUploader,
uploadedImagesDocIds: local!imageDocIds,
exportedPdfId: local!pdfId,
enableExportPdf: true,
```

```
    enableFullpageOutput: true,
    hideBorderInReadOnly: false,
    enableLatexConversion: true
  )
```

```
richHtmlTextField(
  label: "Body",
  labelPosition: "ADJACENT",
  isReadOnly: local!readOnly,
  height: "MEDIUM",
  htmlTextFormatValue: local!outputHtmlText,
  htmlTextFormatSaveInto: local!outputHtmlText,
  maxSize: 5500,
  placeholder: {
    {
      text: "First Name",
      value: "###FirstName###"
    },
    {
      text: "Last Name",
      value: "###LastName###"
    }
  }
)
},
buttons: a!buttonLayout(
 primaryButtons: {
  a!buttonWidget(
    label: "Submit",
    submit: true,
    style: "PRIMARY"
  )
 },
 secondaryButtons: {
  a!buttonWidget(
    label: "Preview",
    style: "SECONDARY",
    value: true(),
    showWhen: not(
      local!readOnly
    ),
    saveInto: local!readOnly
  ),
  a!buttonWidget(
    label: "Go Back",
    style: "SECONDARY",
    value: false(),
```