

# Table of Contents

The `UIEditor` is the standard component for binding editors to the data model. It works great for default data types like text, number, date, or image. But sometimes it's better to use custom editors with advanced features, e.g., if your text contains source code, you could use a text editor with syntax highlighting or auto completion.

You have different options to solve this problem. The first one is to write your own cell editor based on the `ICellEditor` interface. This isn't difficult but sometimes a lot of work and could be tricky. This solution is recommended if you need the cell editor for multiple UI implementations. If you want a quick and easy solution, simply use our wrapper component. Usually, it works without coding and is simple to integrate in your existing application. We used this solution in [VisionX](#) for our [CSS editor](#).

The integration is super simple:

```
panStylesheets.add(new ExternalEditor(new TextEditorComponent(spArea,
    textArea), dataBook, "CSS"));
```

The **ExternalEditor** will be bound to a data book and a specific column, like any other cell editor.

The creation of the `textArea` is not tricky:

```
RSyntaxTextArea textArea = new RSyntaxTextArea(20, 60);
textArea.setSyntaxEditingStyle(SyntaxConstants.SYNTAX_STYLE_CSS);
textArea.setAutoIndentEnabled(true);
textArea.setBracketMatchingEnabled(true);
textArea.setCaretPosition(0);
textArea.setClearWhitespaceLinesEnabled(false);
textArea.setWhitespaceVisible(false);
textArea.setCloseCurlyBraces(true);
textArea.setLineWrap(false);
textArea.setEditable(true);
textArea.setMarkOccurrences(true);
textArea.setTabSize(4);
```

You need a scroll pane as well:

```
RTextScrollPane spArea = new RTextScrollPane(textArea, true);
```

And the theme for syntax highlighting:

```
Theme.load(ResourceUtil.getResourceAsStream("/org/fife/ui/rsyntaxtextarea/themes/eclipse.xml")).apply(textArea);
```

Finally, the auto completion:

```
CompletionProvider provider = new CssCompletionProvider();

AutoCompletion ac = new AutoCompletion(provider);
ac.install(textArea);
```

From:  
<https://doc.sibvisions.com/> - **Documentation**

Permanent link:  
[https://doc.sibvisions.com/applications/external\\_editor](https://doc.sibvisions.com/applications/external_editor)



Last update: **2020/07/03 14:03**