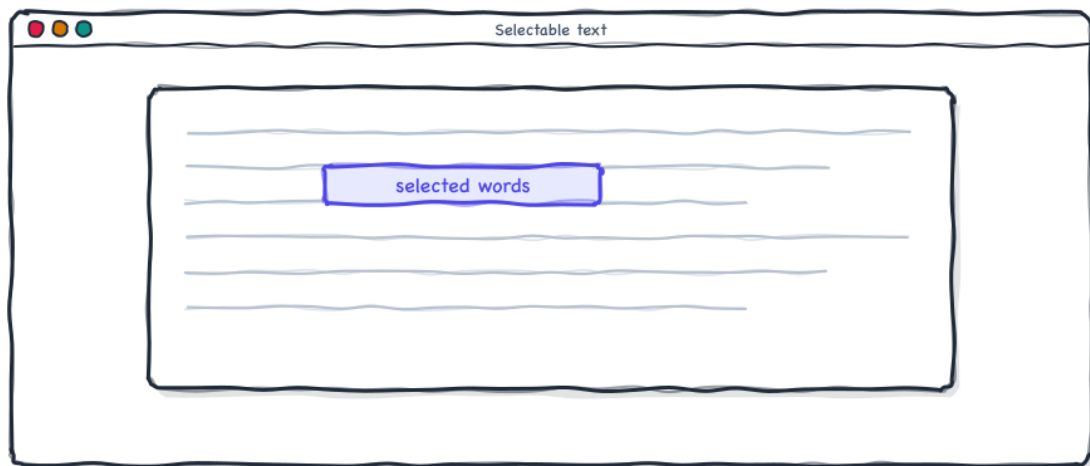


TextView

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



TextView renders selectable text with a highlight path and clipboard actions.

TextView displays multiline text in a selectable TextFlow. Users can drag, double-click words, triple-click paragraphs, copy selection or all text, and customize highlight paints through CSS.

Table of Contents

1. Introduction	2
1.1 Key features	2
1.2 Maven dependency	2
2. Getting started	2
3. Anatomy	3
4. Control API	4
5. Selection behaviour	4
6. Clipboard and context menu	5
7. Styling	5
8. Localization	6
9. Accessibility	6
10. Recipes	6
10.1 Disable click selection shortcuts	6
10.2 Override copy	6
10.3 Use inside a ScrollPane	6
10.4 Checklist	7
11. See also	7

1. Introduction

TextView is a Control for read-only selectable text. It differs from `TextArea` by rendering text through a `TextFlow` and exposing copy operations without editing.

1.1 Key features

- Drag selection across multiline text.
- Double-click selects a word and triple-click selects a paragraph unless disabled.
- `Shortcut+C` copies selected text; `Shortcut+A` selects all; `Backspace` clears selection.
- Context menu offers `Copy All` and `Copy Selection`.
- Read-only `selectedText` property.
- CSS properties for highlight fill, stroke and selected text fill.

1.2 Maven dependency

```
<dependency>
  <groupId>com.dlsc.gemsfx</groupId>
  <artifactId>gemsfx</artifactId>
  <version>4.4.1</version>
</dependency>
```

GemsFX controls live in module `com.dlsc.gemsfx`.

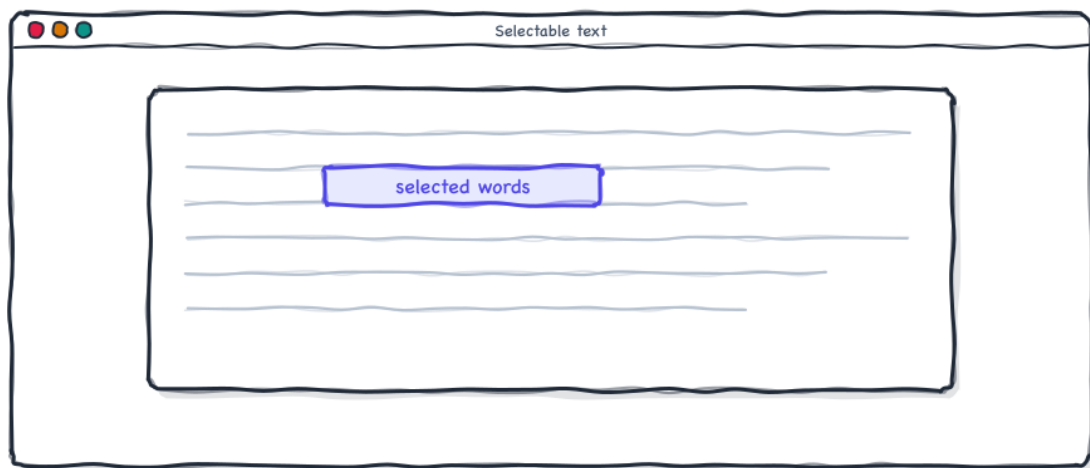
2. Getting started

```
TextView textView = new TextView();
textView.setText(longReadOnlyText);

Label selected = new Label();
selected.textProperty().bind(textView.selectedTextProperty());

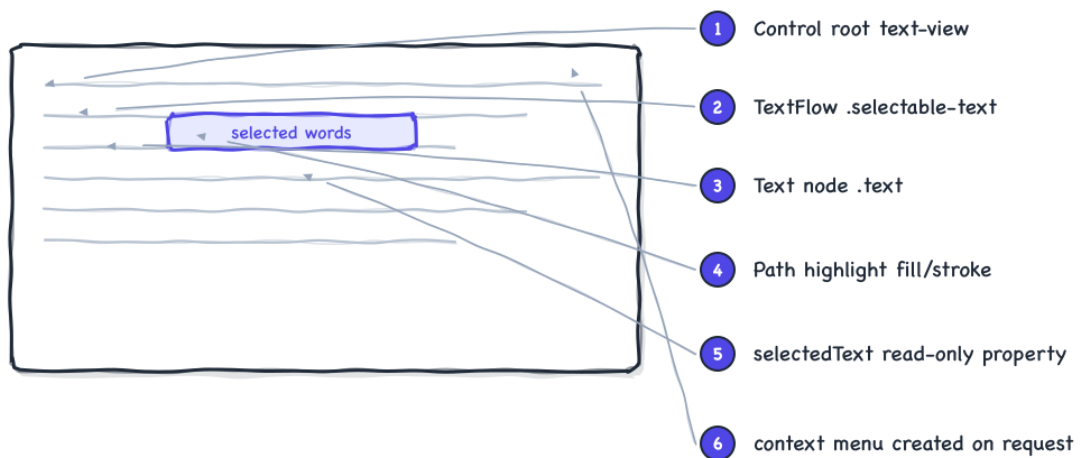
Button copyAll = new Button("Copy all");
copyAll.setOnAction(evt -> textView.copyAll());
```

Display selectable text and observe the selected text.



TextView displays selectable text in a TextFlow.

3. Anatomy



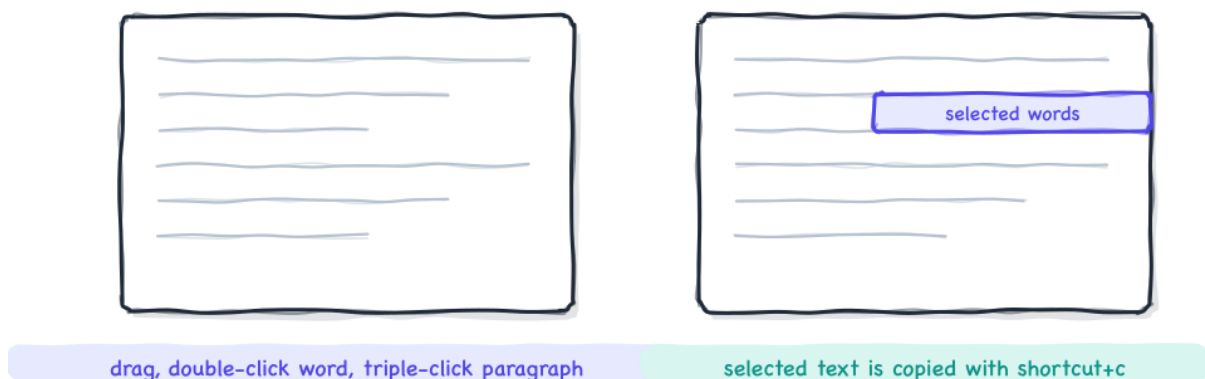
The TextView skin uses a TextFlow, Text and highlight Path.

Part	Style class	Description
.text-view	Root control style class.	
:focused	Changes highlight fill/text fill in text-view.css.	
.selectable-text	TextFlow used by the skin.	
.text	Text node inside the TextFlow.	

4. Control API

Property	Type	Default	Description
text	StringProperty	null	Multiline text displayed by the skin Text node.
selectedText	ReadOnlyStringProperty	null	Updated from the skin when selection changes.
disableTextSelectionByMouseClicks	BooleanProperty	false	Disables double-click word and triple-click paragraph selection, but not dragging.
highlightFill	ObjectProperty<Paint>	Color.BLUE; UA stylesheet derives background	Selection background paint. Styleable as -fx-highlight-fill.
highlightStroke	ObjectProperty<Paint>	Color.TRANSPARENT	Selection outline paint. Styleable as -fx-highlight-stroke.
highlightTextFill	ObjectProperty<Paint>	Color.WHITE; UA stylesheet also sets focused white	Selected text fill. Styleable as -fx-highlight-text-fill.

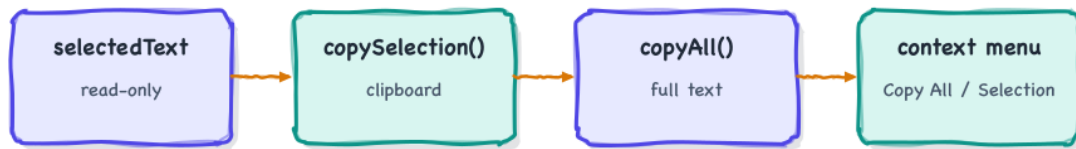
5. Selection behaviour



Mouse selection and keyboard copy behaviour.

- Primary-button drag selects a character range.
- Double-click selects an alphanumeric word.
- Triple-click selects a paragraph up to newline boundaries.
- Selection is removed when size changes or focus is lost while no context menu is open.

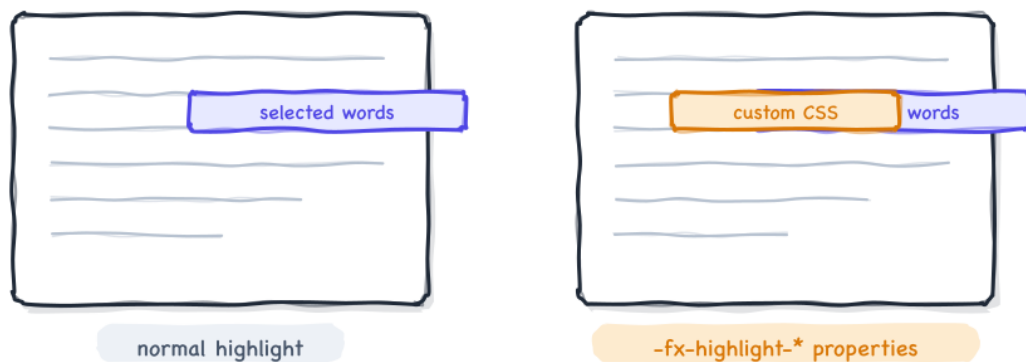
6. Clipboard and context menu



Copy APIs and context-menu items.

`copySelection()` and `copyAll()` are intentionally non-final so applications can override copying behaviour. The context menu is created lazily on first context-menu request and uses localized menu text.

7. Styling



Highlight paints can be customized with CSS.

Style class / pseudo class	Description
<code>.text-view</code>	Root control style class.
<code>:focused</code>	Changes highlight fill/text fill in <code>text-view.css</code> .
<code>.selectable-text</code>	TextFlow used by the skin.
<code>.text</code>	Text node inside the TextFlow.

CSS property	Type	Default
-fx-highlight-fill	Paint	Java default blue; UA default <code>derive(-fx-control-inner-background, -20%)</code>
-fx-highlight-stroke	Paint	transparent
-fx-highlight-text-fill	Paint	Java default white; UA default <code>-fx-text-inner-color</code> / focused white

```
.text-view {
    -fx-highlight-fill: rgba(80, 120, 255, .35);
    -fx-highlight-stroke: #506cff;
    -fx-highlight-text-fill: black;
}
```

Example TextView CSS.

8. Localization

Key	English default
context.copy-selection	Copy Selection
context.copy-all	Copy All

9. Accessibility

TextView sets `AccessibleRole.TEXT` and binds accessible text to the text property through `AccessibilityUtil`. Application-set accessible text takes precedence according to the helper behaviour.

10. Recipes

10.1 Disable click selection shortcuts

```
textView.setDisableTextSelectionByMouseClicks(true);
```

10.2 Override copy

```
TextView view = new TextView() {
    @Override public void copySelection() {
        audit(getSelectedText());
        super.copySelection();
    }
};
```

10.3 Use inside a ScrollPane

```
ScrollPane scroll = new ScrollPane(textView);
scroll.setFitToWidth(true);
```

10.4 Checklist

- 1 Use TextView for read-only selectable text, not editing.
- 2 Bind selectedText when the application needs live selection state.
- 3 Style highlight paints for sufficient contrast.
- 4 Keep focus behaviour in mind when embedding in complex panes.

11. See also

- Demo app: `TextViewApp`. Run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.TextViewApp`.
- Related control: `EnhancedLabel` for whole-label copy selection.
- API docs: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>