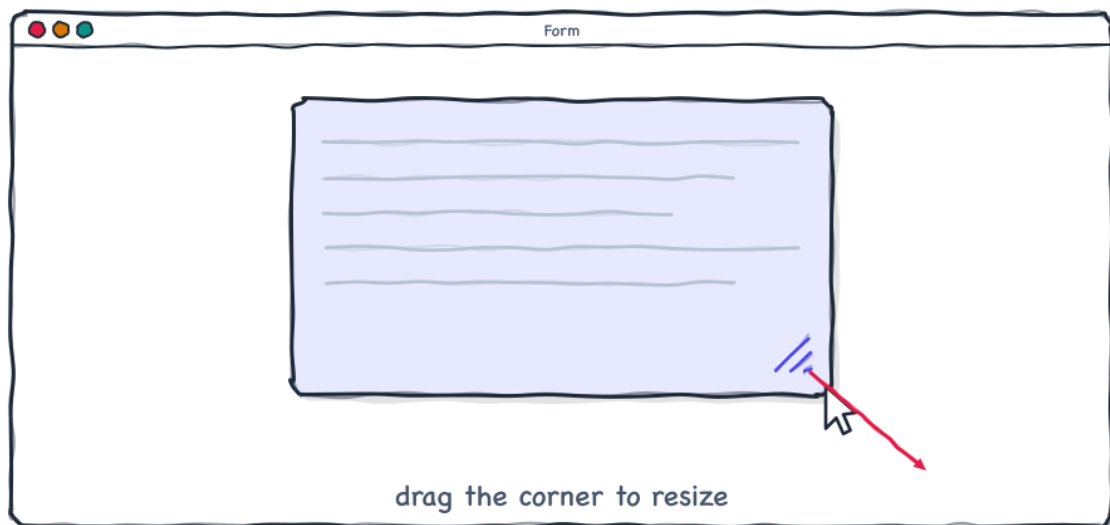


ResizableTextArea

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



A generated cartoon overview of ResizableTextArea.

ResizableTextArea extends TextArea with a resize handle that changes `prefWidth` and/or `prefHeight` according to two direction properties.

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	3
5. Resize interaction	4
6. Sizing boundaries	5
7. Styling	5
7.1 Style classes and pseudo classes	5
8. Accessibility	6
9. Recipes	6
9.1 Vertical only	6
9.2 Cap the drag size	6
9.3 Checklist	6
10. See also	6

1. Introduction

ResizableTextArea is a **TextArea** with a skin-managed resize corner in the lower-right corner. Dragging the handle changes preferred size, so parent layouts can react normally.

1.1 Key features

- Vertical resize enabled by default.
- Horizontal resize disabled by default.
- Handle cursor follows h-resize, v-resize, both-resize or no-resize.
- Drag respects positive max width and max height values.

1.2 Maven dependency

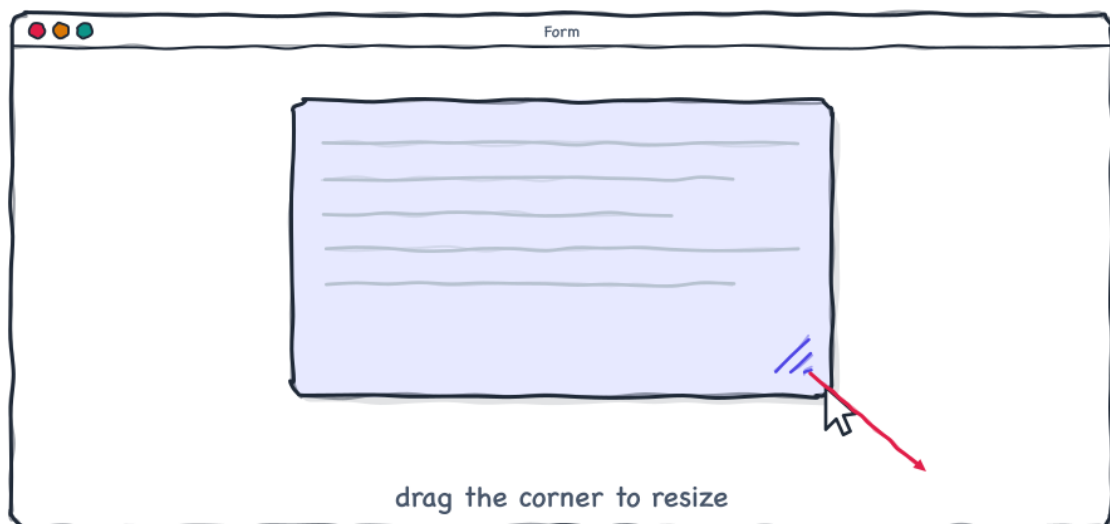
```
<dependency>
  <groupId>com.dlsc.gemsfx</groupId>
  <artifactId>gemsfx</artifactId>
  <version>4.4.1</version>
</dependency>
```

Maven coordinates for the GemsFX control library.

2. Getting started

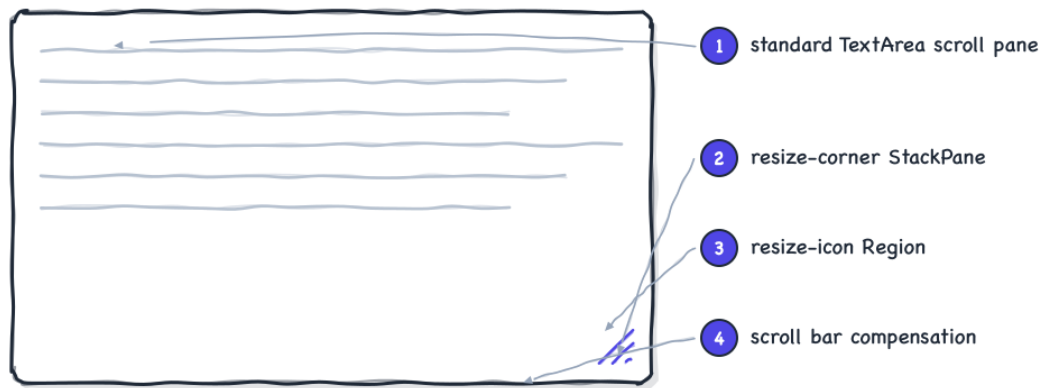
```
ResizableTextArea area = new ResizableTextArea("Resizable text area ...");
area.setWrapText(true);
area.setMinHeight(100);
area.setResizeHorizontal(true);
area.setResizeVertical(true);
```

A complete setup with both resize directions enabled.



The resize handle sits in the lower-right corner.

3. Anatomy



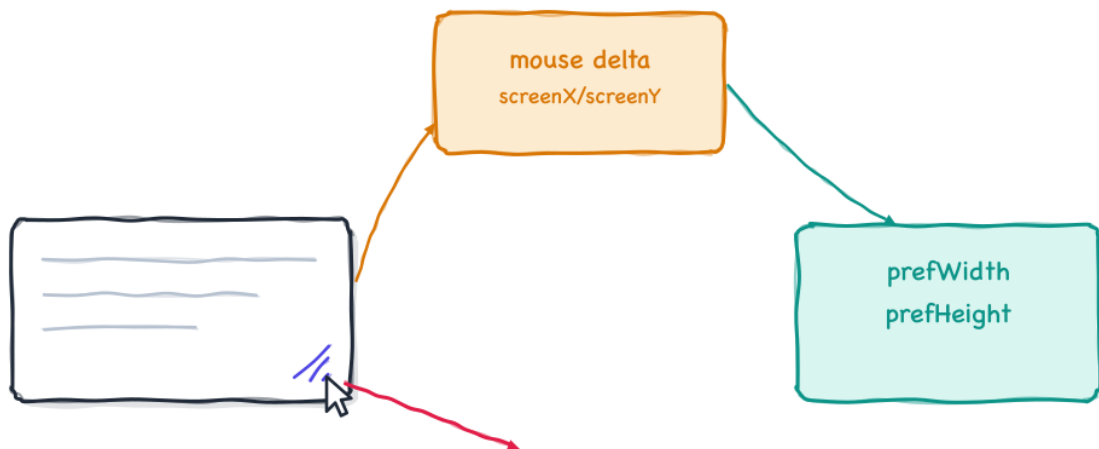
Parts of a ResizableTextArea skin.

Part	Node	Description
Root	<code>.resizable-text-area</code>	TextArea subclass.
Content pane	<code>.content-pane</code>	StackPane containing scroll pane and resize corner.
Resize corner	<code>.resize-corner</code>	Receives mouse press and drag events.
Resize icon	<code>.resize-icon</code>	Region with the bottom-right SVG shape.

4. Control API

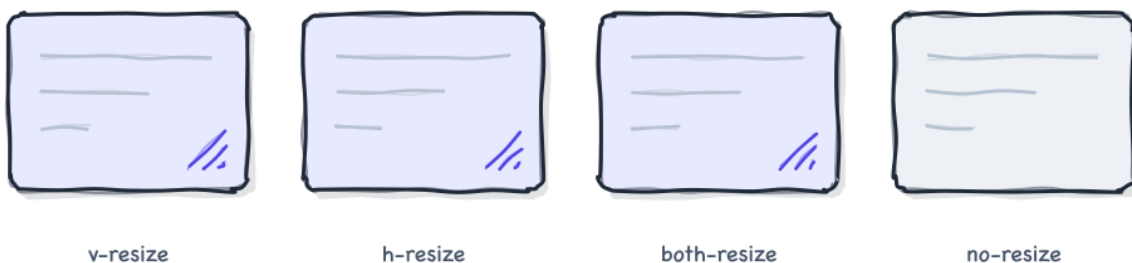
Property	Type	Default	Description
<code>resizeVertical</code>	BooleanProperty	true	If true, dragging changes <code>prefHeight</code> .
<code>resizeHorizontal</code>	BooleanProperty	false	If true, dragging changes <code>prefWidth</code> .
<code>prefWidth</code>	DoubleProperty	inherited	Updated during drag when horizontal resize is enabled.
<code>prefHeight</code>	DoubleProperty	inherited	Updated during drag when vertical resize is enabled.

5. Resize interaction



Mouse deltas are converted into preferred sizes.

On mouse press the skin remembers screen coordinates and current width and height. During drag it computes width and height from the screen delta, caps them to positive `maxWidth` and `maxHeight`, then writes enabled dimensions to preferred size.



pseudo classes live on `.resize-corner`

Resize-direction pseudo classes on the handle.

6. Sizing boundaries

Dragging changes preferred size; it does not directly relocate the node. Parent layout panes still decide final placement from normal JavaFX layout rules.

Constraint	How it is used
maxWidth	If positive, the computed dragged width is capped to this value.
maxHeight	If positive, the computed dragged height is capped to this value.
Minimum size	The skin does not explicitly clamp to min size before writing preferred dimensions. Use parent constraints when a hard minimum matters.

```
area.setResizeHorizontal(true);
area.setMaxWidth(500);
area.setMaxHeight(260);
area.setMinHeight(100);
```

Tip. Because drag writes preferred size, clearing the preferred size later with `Region.USE_COMPUTED_SIZE` hands sizing back to the parent layout.

7. Styling

7.1 Style classes and pseudo classes

Selector	Description
.resizable-text-area	root control
.content-pane	skin content stack
.resize-corner:h-resize	horizontal-only handle
.resize-corner:v-resize	vertical-only handle
.resize-corner:both-resize	both directions
.resize-corner:no-resize	hidden icon / no resize

The control defines no styleable CSS properties.

```
.resizable-text-area .resize-corner .resize-icon {
    -fx-background-color: -fx-accent;
}

.resizable-text-area .resize-corner:hover .resize-icon {
    -fx-background-color: grey;
}
```

8. Accessibility

The constructor sets `AccessibleRole.TEXT_AREA`. The resize corner is mouse-oriented; expose alternative sizing controls in applications that need keyboard resizing.

9. Recipes

9.1 Vertical only

```
area.setResizeVertical(true);
area.setResizeHorizontal(false);
```

9.2 Cap the drag size

```
area.setMaxWidth(600);
area.setMaxHeight(320);
```

9.3 Checklist

- 1 Use preferred-size constraints in the parent layout.
- 2 Set both resize properties to false to hide the icon.
- 3 Use max width / height when the field must stay within a form.

10. See also

- Demo application: `com.dlsc.gemsfx.demo.ResizableTextAreaApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.ResizableTextAreaApp`)
- `LimitedTextArea` - subclass with character limits.
- `ExpandingTextArea` - automatic height instead of dragging.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>