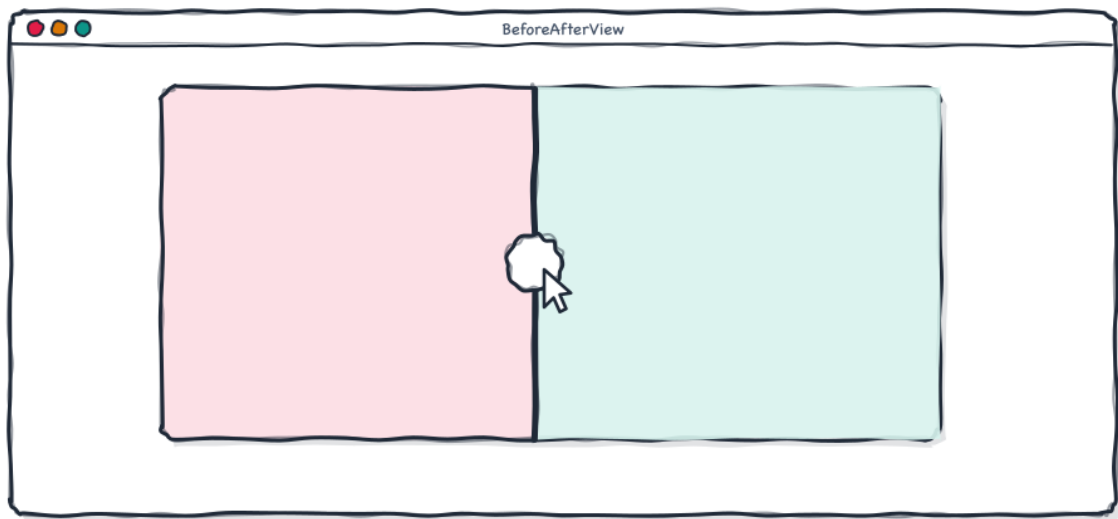


BeforeAfterView

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of BeforeAfterView.

BeforeAfterView layers two nodes and clips them around a draggable divider, so users can reveal more of the before or after state horizontally or vertically.

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
4.1 Properties and callbacks	3
5. Behaviour	4
6. Layout and rendering	5
7. Divider mechanics	5
8. Styling	6
8.1 Style classes and pseudo classes	6
8.2 Styleable CSS properties	6
9. Localization	7
10. Accessibility	7
11. Recipes	7
11.1 Programmatic configuration	7
11.2 Practical checklist	7
12. Integration notes	7
13. See also	8

1. Introduction

BeforeAfterView BeforeAfterView layers two nodes and clips them around a draggable divider, so users can reveal more of the before or after state horizontally or vertically.

1.1 Key features

- before and after can be any Node; image constructor wraps ImageView.
- Default placeholders are localized labels Before and After.
- dividerPosition defaults to 0.5 and is clamped to 0..1 during dragging.
- Orientation.HORIZONTAL is the default; VERTICAL is supported.
- Before and after nodes are made mouseTransparent when set.
- Accessible role is SLIDER with localized role description.

1.2 Maven dependency

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

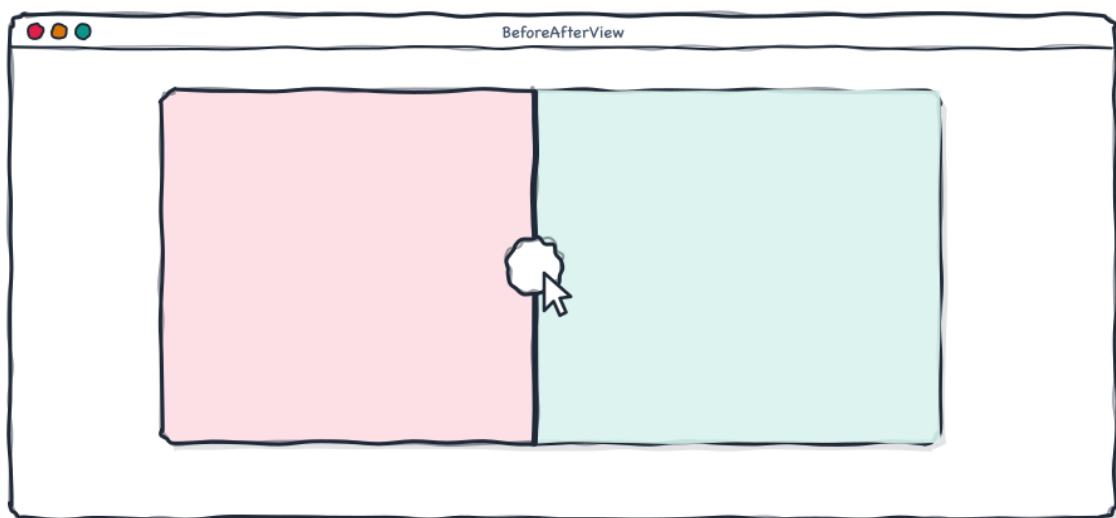
The control lives in module com.dlsc.gemsfx.

2. Getting started

The following snippet uses only public API verified in the control source.

```
BeforeAfterView view = new BeforeAfterView(beforeImage, afterImage);
view.setDividerPosition(.35);
view.setOrientation(Orientation.HORIZONTAL);
view.dividerPositionProperty().addListener((obs, old, pos) -> updateStatus(pos));
```

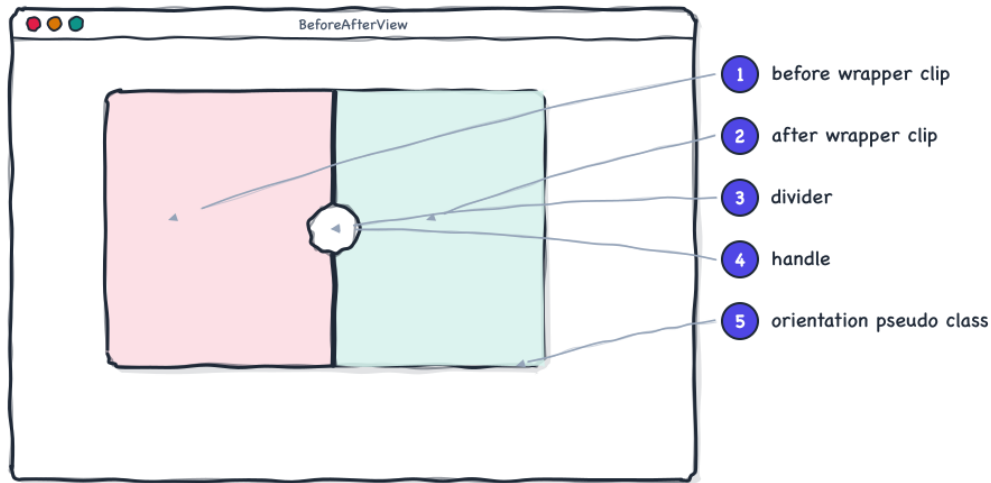
A compact setup for BeforeAfterView.



A generated overview of BeforeAfterView in use.

3. Anatomy

The anatomy diagram identifies the implementation pieces that matter when configuring, styling or debugging the control.



The parts of BeforeAfterView.

Part	Verified detail
Root	Style class <code>.before-after-view</code> is added by the constructor.
Stylesheet	User-agent stylesheet <code>before-after-view.css</code> is returned by the control.

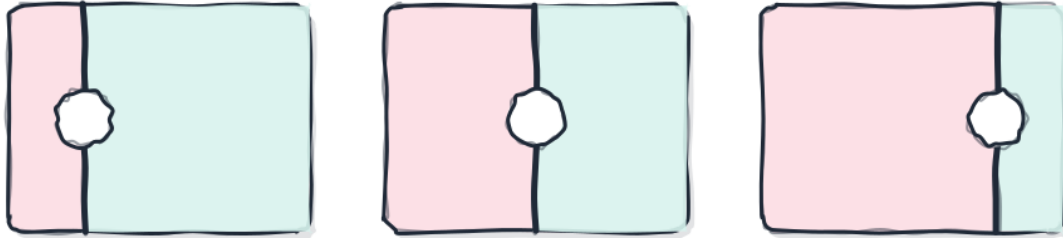
4. Control API

4.1 Properties and callbacks

Property	Type	Default	Description
<code>before</code>	<code>ObjectProperty<Node></code>	localized red Before Label	Node shown on the before side.
<code>after</code>	<code>ObjectProperty<Node></code>	localized green After Label	Node shown on the after side.
<code>dividerPosition</code>	<code>DoubleProperty</code>	0.5	Divider position from 0.0 to 1.0; styleable.
<code>orientation</code>	<code>ObjectProperty<Orientation></code>	HORIZONTAL	HORIZONTAL or VERTICAL; styleable.

Note. Defaults and property names in this table were checked against the Java source for this batch.

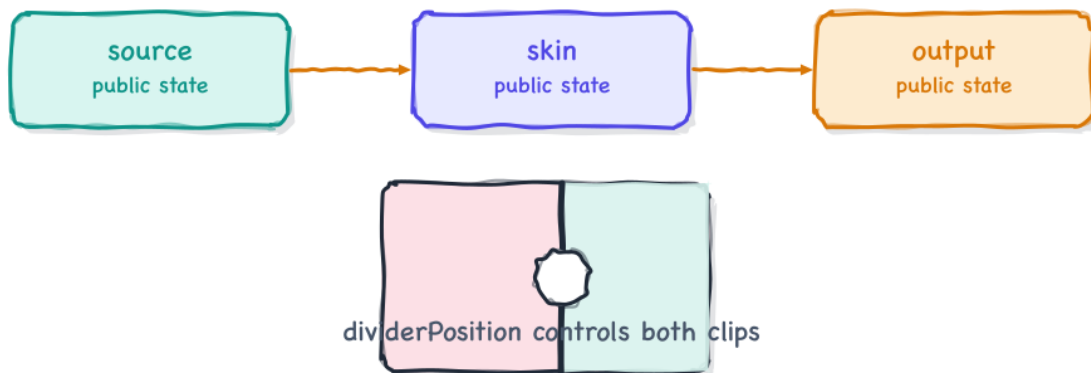
5. Behaviour



drag clamps dividerPosition to 0..1

Important runtime states of BeforeAfterView.

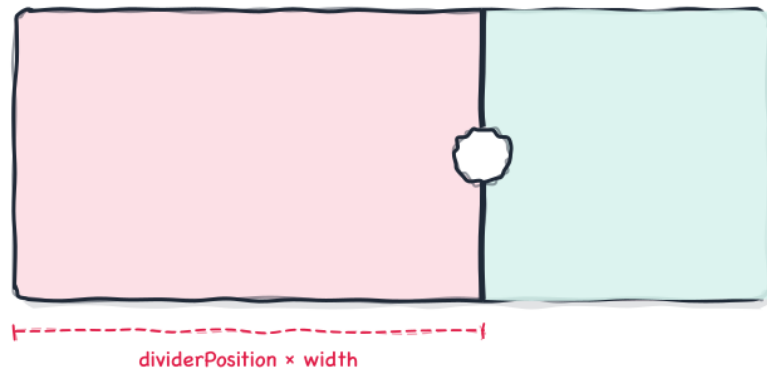
- The skin layers before and after wrappers in a clipped content StackPane.
- For HORIZONTAL, before width is $\text{dividerPosition} * \text{width}$ and after begins at that x.
- For VERTICAL, before height is $\text{dividerPosition} * \text{height}$ and after begins at that y.
- Mouse drag updates dividerPosition by $\text{delta} / \text{width}$ or $\text{delta} / \text{height}$ and clamps to 0..1.
- The divider and handle are visual only and mouseTransparent; the control handles drag events.



How data and geometry flow through BeforeAfterView.

6. Layout and rendering

Rendering is based on two clipped wrappers. The divider position determines how much of the before node and after node is visible.



Rendering and sizing rules for BeforeAfterView.

Concern	Rule
Horizontal	Divider $x = \text{width} \times \text{dividerPosition}$.
Vertical	Divider $y = \text{height} \times \text{dividerPosition}$.
Preferred size	Skin returns max preferred size of before and after nodes.

7. Divider mechanics

The divider is not a child splitter. It is a visual overlay whose position is stored in `dividerPosition`; the actual reveal is produced by two clips on wrapper panes.

Orientation	Clip calculation
Horizontal before	$\text{width} = \text{wrapper.width} \times \text{dividerPosition}$
Horizontal after	$x = \text{wrapper.width} \times \text{dividerPosition}$; $\text{width} = \text{wrapper.width} - x$
Vertical before	$\text{height} = \text{wrapper.height} \times \text{dividerPosition}$
Vertical after	$y = \text{wrapper.height} \times \text{dividerPosition}$; $\text{height} = \text{wrapper.height} - y$

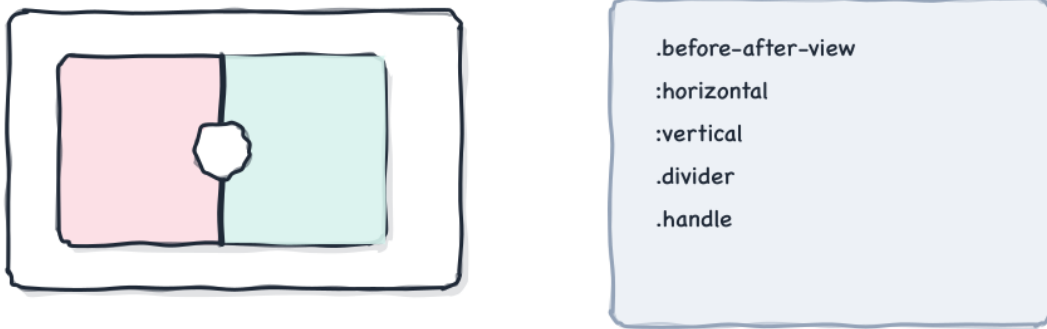
```
view.setOnMouseDragged(evt -> {
    double delta = evt.getX() - startX;
    view.setDividerPosition(Math.min(1, Math.max(0,
        view.getDividerPosition() + delta / view.getWidth())));
});
```

Simplified horizontal drag logic from the skin.

Tip. The source makes before and after nodes mouseTransparent, so pointer input belongs to the comparison control instead of the compared content.

8. Styling

The user-agent stylesheet is before-after-view.css. The table lists selectors and pseudo classes that exist in source, skin or stylesheet.



Style hooks for BeforeAfterView.

8.1 Style classes and pseudo classes

Selector / pseudo class	Purpose
.before-after-view	Verified in source, skin or CSS.
:horizontal	Verified in source, skin or CSS.
:vertical	Verified in source, skin or CSS.
> .content	Verified in source, skin or CSS.
> .divider	Verified in source, skin or CSS.
> .handle	Verified in source, skin or CSS.
.handle .ikonli-font-icon	Verified in source, skin or CSS.

8.2 Styleable CSS properties

CSS property	Type	Default
-fx-orientation	Orientation	HORIZONTAL
-fx-divider-position	size	0.5

```
.before-after-view {
    /* start with the documented root selector */
}
```

CSS example using documented hooks.

9. Localization

Key	English text
placeholder.before	Before
placeholder.after	After
accessible.role-description	before after comparison

10. Accessibility

BeforeAfterView sets AccessibleRole.SLIDER with localized role description "before after comparison". It does not bind an accessible value to dividerPosition.

11. Recipes

11.1 Programmatic configuration

```
BeforeAfterView view = new BeforeAfterView(beforeImage, afterImage);
view.setDividerPosition(.35);
view.setOrientation(Orientation.HORIZONTAL);
view.dividerPositionProperty().addListener((obs, old, pos) -> updateStatus(pos));
```

11.2 Practical checklist

- 1 Clamp programmatic dividerPosition values yourself if they come from user data.
- 2 Set both before and after nodes before showing the control.
- 3 Use the public properties listed in the API chapter.
- 4 Style only through documented selectors and styleable properties.
- 5 Do not depend on private skin node structure except for documented CSS selectors.

12. Integration notes

No keyboard handling for divider movement is implemented.

Topic	Recommendation
Threading	Keep image loading and expensive rendering off the UI path when the control exposes a background option.
Styling	Scope selectors under the documented root style class.
Accessibility	Preserve the source-defined accessible role and add app-specific text when the control does not bind it.
State	Prefer public properties over skin node lookup.

13. See also

- Demo application: `com.dlsc.gemsfx.demo.BeforeAfterViewApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.BeforeAfterViewApp`)
- Related GemsFX media controls: `AvatarView`, `PhotoView`, `SVGImageView`, `BeforeAfterView`, `MaskedView`, `ScreensView`.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>