

SelectionBox

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of SelectionBox.

A generic Control that opens a popup containing radio buttons in single-selection mode or check boxes in multiple-selection mode. The popup can host top, bottom, left and right decoration nodes.

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. Styling	5
6.1 Style classes and pseudo classes	5
6.2 Styleable CSS properties	6
7. Localization	6
8. Accessibility	6
9. Recipes	6
9.1 Programmatic configuration	6
9.2 Checklist	6
10. See also	7

1. Introduction

SelectionBox A generic Control that opens a popup containing radio buttons in single-selection mode or check boxes in multiple-selection mode. The popup can host top, bottom, left and right decoration nodes.

1.1 Key features

- Clicking the control toggles the popup unless disabled or read-only.
- Single mode uses RadioButton items; multiple mode uses CheckBox items.
- API and styling details below are verified against the control, skin, CSS and resource bundles.

1.2 Maven dependency

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

2. Getting started

```
SelectionBox<String> box = new SelectionBox<>("One", "Two", "Three");
box.setPromptText("Choose items");
box.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);
box.setSelectedItemsConverter(new SimpleStringConverter<>(items -> items.size() + " selected
"));
```

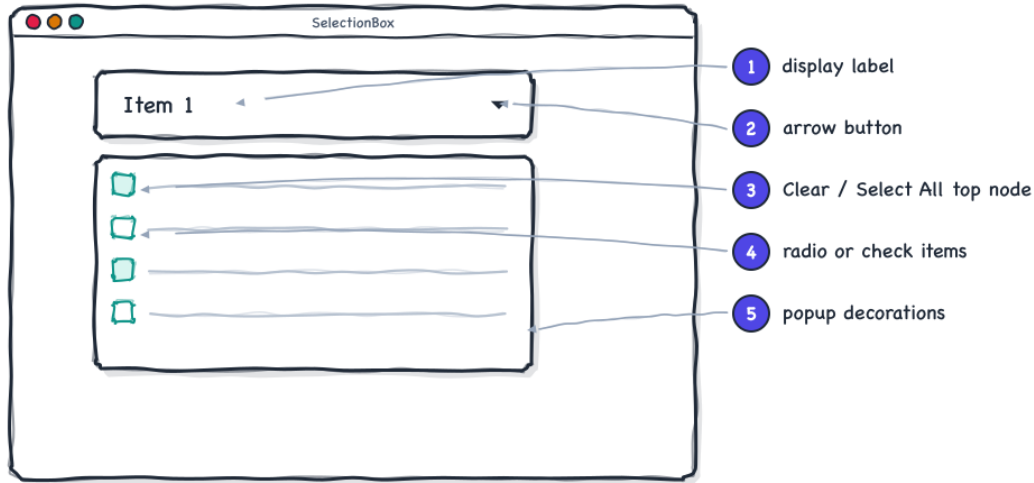
A compact setup for SelectionBox.



SelectionBox in a typical application context.

3. Anatomy

The diagram identifies the nodes and state holders created by the implementation.



The parts of SelectionBox.

Topic	Verified source detail
Stylesheet	com/dlsc/gemsfx/selection-box.css
Root style class	.selection-box
Graphics	Generated SVGs; no screenshots.

4. Control API

4.1 Properties and callbacks

Property	Type	Default	Description
items	ListProperty<T>	empty list	Items displayed in the popup.
selectionModel	ObjectProperty<MultipleSelectionModel<T>>	CustomMultipleSelectionModel	Selection state.
currentSelectionMode	ReadOnlyObjectProperty<SelectionMode>	selection model mode	Read-only mode mirror.
top / bottom / left / right	ObjectProperty<Node>	top has buttons, others null	Popup BorderPane decorations.
placeholder	ObjectProperty<Node>	null	Shown when items is empty.
graphic	ObjectProperty<Node>	null	Graphic in display label.
promptText	StringProperty	null	Shown when nothing is selected.
autoHideOnSelection	BooleanProperty	true	Controls hiding after item / extra-button actions.

Property	Type	Default	Description
selectedItemsConverter	ObjectProperty<String Converter<List<T>>>	null	Formats selected values.
itemConverter	ObjectProperty<String Converter<T>>	null	Formats popup items.
readOnly	BooleanProperty	false	Disables interaction; styleable.
animationEnabled	BooleanProperty	false	Enables 200 ms fade transitions; styleable.
onShowing/onShown/onHiding/onHidden	ObjectProperty<EventHandler<WindowEvent>>	null	Popup lifecycle handlers.

5. Behaviour



Important runtime states of SelectionBox.

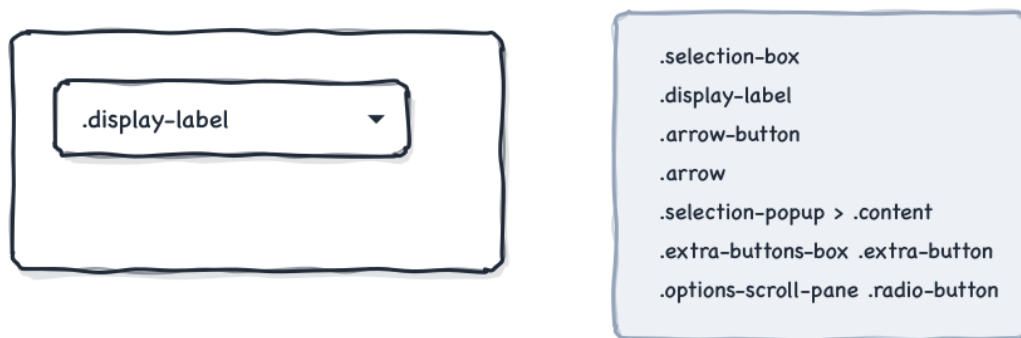
- Clicking the control toggles the popup unless disabled or read-only.
- Single mode uses RadioButton items; multiple mode uses CheckBox items.
- The popup auto-fixes, auto-hides and hides on ESCAPE.
- show() and hide() communicate with the skin via the showPopup marker.
- If animationEnabled is true, popup opacity fades over 200 ms.



Selection model + converters produce display text

How application data flows through SelectionBox.

6. Styling



Style hooks for SelectionBox.

6.1 Style classes and pseudo classes

Selector / pseudo class	Purpose
.selection-box	Verified in source, skin or CSS.
.display-label	Verified in source, skin or CSS.
.arrow-button	Verified in source, skin or CSS.
.arrow	Verified in source, skin or CSS.
.selection-popup > .content	Verified in source, skin or CSS.
.extra-buttons-box .extra-button	Verified in source, skin or CSS.
.options-scroll-pane .radio-button	Verified in source, skin or CSS.
.options-scroll-pane .check-box	Verified in source, skin or CSS.

Selector / pseudo class	Purpose
:showing	Verified in source, skin or CSS.
:empty	Verified in source, skin or CSS.
:single	Verified in source, skin or CSS.
:multiple	Verified in source, skin or CSS.
:readonly	Verified in source, skin or CSS.

6.2 Styleable CSS properties

CSS property	Type	Default
-fx-read-only	boolean	false
-fx-animation-enabled	boolean	false

```
.selection-box {
    /* start with the documented root selector */
}
```

7. Localization

Key	English text
action.clear	Clear
action.select-all	Select All

8. Accessibility

Sets AccessibleRole.COMBO_BOX and binds accessible text to the current selection.

9. Recipes

9.1 Programmatic configuration

```
SelectionBox<String> box = new SelectionBox<>("One", "Two", "Three");
box.setPromptText("Choose items");
box.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);
box.setSelectedItemsConverter(new SimpleStringConverter<>(items -> items.size() + " selected"));
```

9.2 Checklist

- 1 Use the public properties listed in the API chapter.
- 2 Style through documented selectors and CSS properties only.
- 3 Keep application model updates in callbacks such as onClose, onClear or onItemSelected.
- 4 Do not depend on private skin nodes except for documented style selectors.

10. See also

- Demo application: `com.dlsc.gemsfx.demo.SelectionBoxApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.SelectionBoxApp`)
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>